

TOWARD THE FRONTIERS OF STACKED GENERALIZATION
ARCHITECTURE FOR LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CÜNEYT MERTAYAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2007

Approval of the thesis

**“TOWARD THE FRONTIERS OF STACKED GENERALIZATION
ARCHITECTURE FOR LEARNING”**

submitted by **Cüneyt Mertayak** in partial fulfillment of the requirements for
the degree of **Master of Science in Computer Engineering, Middle East
Technical University** by,

Prof. Dr. Canan Özgen

Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. Volkan Atalay

Head of Department, **Computer Engineering**

Prof. Dr. Fatoş Tünay Yarman-Vural

Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Volkan Atalay

Computer Engineering, METU

Prof. Dr. Fatoş Tünay Yarman-Vural

Computer Engineering, METU

Prof. Dr. Neşe Yalabık

Computer Engineering, METU

Assoc. Prof. Göktürk Üçoluk

Computer Engineering, METU

Assist. Dr. İlkey Ulusoy

Electrical and Electronics Engineering, METU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Cüneyt Mertayak

Signature :

ABSTRACT

TOWARD THE FRONTIERS OF STACKED GENERALIZATION ARCHITECTURE FOR LEARNING

Mertayak, Cüneyt

M.S., Department of Computer Engineering

Supervisor: Prof. Dr. Fatoş Tünay Yarman-Vural

September 2007, 64 pages

In pattern recognition, “bias-variance” trade-off is a challenging issue that the scientists has been working to get better generalization performances over the last decades. Among many learning methods, two-layered homogeneous stacked generalization has been reported to be successful in the literature, in different problem domains such as object recognition and image annotation. The aim of this work is two-folded. First, the problems of stacked generalization are attacked by a proposed novel architecture. Then, a set of success criteria for stacked generalization is studied.

A serious drawback of stacked generalization architecture is the sensitivity to *curse of dimensionality* problem. In order to solve this problem, a new architecture named “unanimous decision” is designed. The performance of this architecture is shown to be comparably similar to two layered homogeneous stacked generalization architecture in low number of classes while it performs better than stacked generalization architecture in higher number of classes. Additionally, a new success criterion for two layered homogeneous stacked generalization architecture is proposed based on the individual properties of the used descriptors and it is verified in synthetic datasets.

Keywords: stacked generalization, architecture, ensemble learning, object classification, MPEG-7 descriptors

ÖZ

ÖĞRENME İÇİN YIĞILMIŞ GENELLEME MİMARİSİNİN SINIRLARINA DOĞRU

Mertayak, Cüneyt

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Fatoş Tünay Yarman-Vural

Eylül 2007, 64 sayfa

Örüntü tanımada, “yanlılık-sapma” değişimi öyle zorlayıcı bir konudur ki araştırmacılar daha iyi genelleme performansları elde edebilmek için onlarca yıldır çalışmaktadır. Bir çok öğrenme yöntemi arasından, iki seviyeli homojen yığılmış genelleme, literatürde nesne tanıma ve görüntü açıklama gibi değişik problem tiplerinde yüksek başarıyla rapor edilmiştir. Bu çalışmanın iki temel amacı vardır: Biricisi, yığılmış genellemenin bazı problemlerini çözmek için yeni bir mimari önermektir. İkincisi, yığılmış genelleme için başarı kriterleri bulmaktır.

Yığılmış genelleme mimarisinin bir problemi, niteliği gereği *boyut lanetine* açık olmasıdır. Bu problemi çözmek için, “ortak karar” olarak isimlendirilmiş yeni bir mimari önerilmiştir. Bu mimari düşük sınıf sayısında iki katmanlı homojen yığılmış genelleme mimarisiyle benzer performansı göstermekte, ancak yüksek sınıf sayısında yığılmış genellemeden daha iyi olmaktadır. Buna ek olarak, iki katmanlı homojen yığılmış genelleme için kullanılan betimleyicilerin bireysel özelliklerine dayanan başarı kriterleri ortaya konulmuş ve sentetik veri kümelerinde doğrulanmıştır.

Anahtar Kelimeler: yığılmış genelleme yapısı, toplu öğrenme, nesne sınıflandırma, MPEG-7 betimleyicileri

ACKNOWLEDGMENTS

First of all, I would like to thank Prof. Dr. Yarman-Vural for her great patience and guidance through my work. I feel that one of my biggest opportunities during this work is to share experiences with the members of Image Processing Laboratory of Computer Engineering Department. Among the members of the laboratory, I express gratitude to İsmet Yalabık for his helps with some software tools and supply of more computing power when needed, Özge Öztimur for feedback in good quality and Emre Akbaş for explanations on extraction of MPEG-7 descriptions and helpful comments. Also, I appreciate of Mehmet Gülek and Onur Çobanoğlu's helpful comments. I am pleased to get help on latex from Onur Tolga Şehitoğlu. In addition to these, I cannot go without expressing gratitude to my family and Fulya Tunçer for their magnificent support, without which this work would not have been completed. This work is partially supported by TÜBİTAK (The Scientific and Technical Research Council of Turkey) BAYG-2210 Research Scholarship.

to my mother

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	v
ACKNOWLEDGMENTS	vi
DEDICATON	vii
TABLE OF CONTENTS	viii
LIST OF FIGURES	x
LIST OF TABLES	xii
CHAPTER	
1 INTRODUCTION	1
1.1 Problem Definition	1
1.2 Challenges of Pattern Recognition in Computer Vision	4
1.3 Ensemble Learning and Questions about its Performance	6
1.4 Contributions of the Thesis	8
2 RELATED WORK IN ENSEMBLE LEARNING	9
2.1 Ensemble of Classifiers: Is It Better to Use More Than One Classifier?	9
2.1.1 Individual Classifiers	9
2.1.2 Major Individual Classifiers	10
2.1.3 Ensemble of Classifiers	12
2.1.4 Major Classifier Combination Methods	15
2.2 Stacked Generalization	19
2.2.1 The Architecture	19
2.2.2 Problems of the Stacked Generalization Architectures	23
2.3 Chapter Summary	24

3	UNANIMOUS DECISION	26
3.1	Motivation	26
3.2	Unanimous Decision	27
3.2.1	Basic Block	27
3.2.2	The Overall Unanimous Decision Architecture	30
3.2.3	The Individual Classifiers at Each β_i	30
3.3	Chapter Summary	32
4	EMPIRICAL STUDY	33
4.1	Data Set	33
4.2	Base and Weighting Classifiers	33
4.2.1	Parameter Estimation	34
4.3	Feature Extraction	35
4.4	Parameter Estimation	36
4.5	Comparison of Stacked Generalization and Proposed Method	37
4.5.1	Classification Performances	37
4.5.2	Computational Complexities	38
4.6	Analysis of Stacked Generalization	39
4.6.1	The Mapping	39
4.6.2	A Set of Success Criteria to Select Descriptors for Homogeneous Stacked Generalization	40
4.7	Implementation	41
4.8	Chapter Summary	41
5	CONCLUSION	55
	REFERENCES	57
	A SOME SAMPLES FROM THE DATASET	62

LIST OF FIGURES

FIGURES

Figure 1.1	A sample from Corel Dataset, adapted from [36].	2
Figure 1.2	An overview diagram of a prototyped classification system.	3
Figure 2.1	Illustration of boosting’s ability to create new hypothesis that is not in the hypotheses space of the classifier, adapted from [18].	14
Figure 2.2	Illustration of a general two-layered stacked generalization architecture.	20
Figure 2.3	Some of the images from one of the datasets used in [48] with their class label. The dataset includes 10 different categories.	22
Figure 2.4	The weights of the mean recognition node for the samples of the “bus” category in [48]. This figure is adapted from [48].	22
Figure 2.5	The architecture proposed in [48]. The figure is adapted from the same work.	23
Figure 2.6	Some example images used in the work [2]. These images from one of the “Europe” categories with the annotation words: “Europe, house, landscape”. This figure is adapted from [2].	24
Figure 2.7	Supervised Ensemble of Visual Descriptors, proposed in [2]. This figure is adapted from the same work.	25
Figure 3.1	The number of classes versus classification performance comparison in [48] in order to show the performance decrease as the number of classes increases.	27
Figure 3.2	The First Basic Block: β_1	28
Figure 3.3	The k^{th} Building Block: β_k	29
Figure 3.4	Unanimous Decision Architecture.	31

Figure 4.1	The comparison of the descriptors in terms of classification performance in the test set, in the presence of different number of classes.	46
Figure 4.2	The comparison of unanimous decision to stacked generalization with four descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure. .	46
Figure 4.3	The comparison of unanimous decision to stacked generalization with five descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure. .	47
Figure 4.4	The comparison of unanimous decision to stacked generalization with six descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure. .	47
Figure 4.5	The comparison of unanimous decision to stacked generalization with seven descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure. .	48
Figure 4.6	The comparison of unanimous decision to stacked generalization with eight descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure. .	48
Figure 4.7	The training and the testing set in descriptor space-1; and the likelihoods from the corresponding base descriptor.	51
Figure 4.8	The training and the testing set in descriptor space-2; and the likelihoods from the corresponding base descriptor.	52
Figure 4.9	The likelihoods of the stacked generalization architecture corresponding to the testing set.	53
Figure 4.10	The representation of an instance of imaginary data in four different descriptor spaces.	54

LIST OF TABLES

TABLES

Table 4.1	Estimated k Values and Corresponding Performances for the Descriptors Color Structure and Scalable Color	42
Table 4.2	Estimated k Values and Corresponding Performances for the Descriptors Color Layout and Edge Histogram	43
Table 4.3	Estimated k Values and Corresponding Performances for the Descriptors Haar Coefficients and Homogeneous Texture	44
Table 4.4	Estimated k Values and Corresponding Performances for the Descriptors Dominant Color(s) and Region Shape	45
Table 4.5	Comparison of Unanimous Decision (UD) and Stacked Generalization (SG)	49
Table 4.6	The Class-Based Performances of Stacked Generalization Architecture in 40-Class Problem with 4 Descriptors	50
Table 4.7	The Class-Based Performances of Unanimous Decision Architecture in 40-Class Problem with 4 Descriptors	50
Table 4.8	The Average Performance of Individual Descriptors and the Average Per- formance of Their Triple Combinations with Stacked Generalization . . .	51

CHAPTER 1

INTRODUCTION

In this chapter, we discuss pattern classification problem and its challenges. Then, the major aims of the thesis will be given in line with the challenges and the state of the art.

1.1 Problem Definition

Pattern classification task distributes the objects into categories to which they belong based on some optimality criteria. In the computer world, which is impressed by the performance of humans, the classification has been a deep research subject for a few decades knowing the fact that the human beings are able to classify objects, starting from childhood. It is known that even a six-year-old infant has the ability of recognizing more than a thousand categories [6]. For instance, the sample image shown in Figure 1.1, can be easily classified into the bus category by a human observer. Today, developing a classification system that is comparable to the human's classification competence is one of the challenging problems. The need for such a system constitutes the compulsive power for creation of several new fields including pattern recognition, machine learning and data mining. In pattern recognition and machine learning, the classification is defined to be a mapping from samples to a set of object categories. The automated tools specialized for this task are called as classifiers [41].

Classifiers can be grouped into two main categories: *syntactic* and *statistical*. Syntactic classifiers are based on logical relationships of structural parts. To illustrate, grammar induction is as an example for these types of classification methodologies [44]. On the other hand, in statistical classification, the samples are placed into sorts of quantitative measurements, where the samples from each category are assumed to be drawn from a statistical model. Some examples of statistical pattern recognition are k -nearest neighbors method, Bayesian belief networks and decision trees [18]. Although it is intuitive that structural relationships



Figure 1.1: A sample from Corel Dataset, adapted from [36].

are more important than statistical relationships, from the progress of the literature, statistical classifiers are more accepted and widely utilized by the pattern recognition community as they serve more than one application domain by preserving its higher performance level compared to structural classifiers'. Mumford [39] states that people from speech domain were agreed that statistics is more crucial in 70s while artificial intelligence community believed in the significance of it in 80s and computer vision researchers were agreed in 90s when computers are evolved to be powerful enough to handle megabytes of image data. In this thesis, we focus on statistical classifiers, hereinafter, the “classifiers” are statistical in nature.

Based on the definition of Duda et. al. [18], the classification task has three major stages, which are preprocessing, feature extraction and classification, as shown in Figure 1.2. As indicated in the figure, these methods do not have to be only feed-forward and may also include feedback from the forth coming parts [18, 34]. Preprocessing stage is the removal of unrelated and redundant information with domain dependent methods. For example, in most of the problems the background carries irrelevant information about the object and segmentation is one of the methods to eliminate this type of information. Features are quantitative measures related to samples and feature extraction is the method of measuring the quantities. For instance, edges in an image is a feature and can be extracted with algorithms such as Canny edge detector [10]. Feature measurements create a space of vectors called feature space. Finally, classification step involves a mapping from the feature space to the set of categories to find out the category to which each sample belongs. A simple example for the classification algorithms is the nearest neighbor method [18]. In this method, the distance of a new sample distance to any other previously observed sample is calculated by a predetermined distance function. According to these distances, the category label of

the nearest sample is assigned to the new sample.

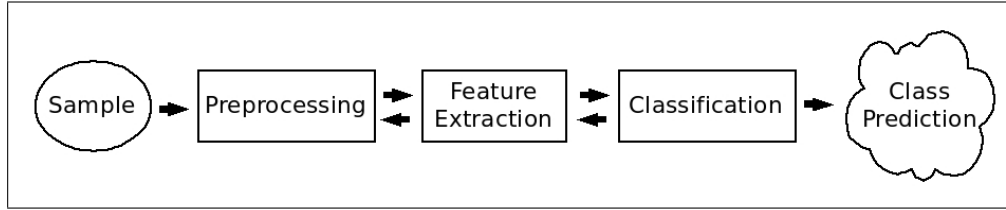


Figure 1.2: An overview diagram of a prototyped classification system.

As aforementioned, feedback can take an important role in classification systems. Evidences for the presence of feedback in neural activity of the brain [9] and the limited efficiency of computer vision algorithms in different areas including visual recognition [24, 55] lead more sophisticated algorithms with feedback mechanisms to build new models which are more successful than the early ones, and to develop new computational cognitive models for understanding how the brain works. Biologists consider the feedback mechanisms mainly as selective attention with biased competition [15]. Although the dominant view in the computational vision field is still feed-forward chain and some of these models can explain some experimental data well [55], there are research groups trying to embed feedback mechanisms in Bayesian inference enhancing the performance of the existing algorithms and attract attentions of vision community to the role of feedback in cognition [16, 34]. Lee et al. [34] proposed a computational model based on Bayesian network with feedback that is claimed to resemble the interrelation between visual cortical fields. The lower part of the system stands the low level visual components, such as retina in the eye and primary visual cortex. Toward the top of the system, the models of higher level information processing units, such as inferotemporal cortex, are arranged. In the network, the edges from both higher to lower, and lower to higher components enable information flow in both directions ensuring the feedback involvement into the processes realized in lower level components. The interpretation of an image data is done when the information flows in the system becomes stable in both directions.

1.2 Challenges of Pattern Recognition in Computer Vision

There are different types of challenges encountered during the different stages of the classification system, indicated in Figure 1.2. One of the biggest challenges in pattern recognition in computer vision field is the visual grouping of regions, that is, *segmentation*. This process is in *preprocessing* stage and involves basically the identification of significant regions of a given image. However, the regions differentiate in properties from different perspectives; for example, color uniformity is a criterion for ideal surfaces while texture is an important aspect for an image containing lots of textured regions. Also, using different scales leads different results in segmentation. As an example if fine representation is used as a scale, leaves on a tree can be extracted whereas if coarse representation is used the tree will be segmented.

Design of feature space, related to *feature extraction* stage of Figure 1.2, is also a challenging problem in pattern recognition systems. The *no free lunch* theorem states that a generic feature space does not exist for all classification problems [52]. This fact leads to research to find the “most appropriate” feature spaces according to a predefined criterion. In order to find such a feature space for a specific classification problem, two different approaches were proposed based on a given set of features: to find a subset of feature set and to find a modified set of it.

Finding a subset of a set of features is named as *feature selection* [14, 26] and can be realized by two different approaches: *wrapper methodology* and *filter methodology*.

- Wrapper methodology utilizes the classifier of interest in order to score subsets of the features according to classification performance with these subsets. The methodology finally chooses the subset with the best performance on a training set. Therefore, the performance of the system depends highly on the training set. If the data in the training set is representative and large enough, then the performance of the system on a test is satisfactory. Otherwise, one may end up with a poor performance depending on the diversity of the test set.
- Filter methodology selects a subset of features, independent of the chosen classifier. In filter based methods, the optimization of a single or a set of parameters depending on the training set is used. These parameters can be redundancy and relevance [40]. These group of techniques do not require a classifier to design the feature space.

Besides the selection of a subset of the feature set, the other method is to modify the feature set. A new set is designed from the provided features. This type of methods is known

as *kernel methods* [13]. Kernel methods approach the problem by mapping the feature space into higher dimensional space. This transformation is realized by a set of functions also termed with the name of the method, i.e. *kernel functions*. One of the most popular algorithm operating with kernel functions is *support vector machine* [43].

An inherent difficulty in pattern recognition is not being able to achieve high classification performances in the presence of high number of categories with relatively low number of samples. The fall of the classification performance emanates from the increase in the number of categories. Although the dataset with 101 visual categories proposed by Fei-Fei et al. [20] contains more categories than other datasets that had been proposed so far, it is still far from the number of category that can be recognized by of a six-year old child. Also, the performance noted in this dataset is considerably lower than child's performance [24, 25, 33, 55]. In order to work with more categories, Griffin et al. [25] provided a 256-category dataset after the work of Fei-Fei et al. [20]. It is observed that the algorithms tested with 256 categories result in lower performances than the performances noted with the 101-category dataset as it is expected [25]. Towards adopting the automated behaviors of human beings, Fei-Fei et al. also anticipates that the number of training samples shall be reduced to low numbers, even only one sample per categories as a six-year-old child is able to categorize the objects by examining only one sample. However, this is a quite challenging problem since performance of the statistical learning algorithms mostly relies on the number of samples when the number of measurements in the feature extraction step is heavy. This problem is known as curse of dimensionality and it will be further discussed in Chapter 2. Regarding to these problems, in the challenging dataset proposed by [20] and in different image classification domains, Zhang et al. [55] obtained considerable generalization performances, using a few training samples with their approach combining support vector machines and k -nearest neighbors classifier

There are many hidden and/or ambiguous parts of signals gathered in visual perception. *Semantic gap* problem is caused by the insufficiency of the concepts represented by low level features. The systems that have been developed so far are far from the expected classification performance.

The human mind is able to complete hidden and ambiguous parts that are present in the signal; however, the systems that have been developed so far only adopt low level features which are not qualified to represent the high-level concepts such as class labels. Human perception of patterns is based on highly complex visual and cognitive processes which includes information based on the expectations about the situation [39]. However, most of the

current recognition systems depends on numerical feature computations and predetermined distance functions. The gap between computer and human visual system leads unsatisfactory results in the current pattern recognition application.

It seems that semantic gap is one of the biggest problems standing on the road from computer vision to human-level vision ability. Therefore, semantic gap should be reduced for boosting the performance of the recognition systems. In the last two decades, to fill this gap, two new approaches were proposed, namely, *unification of bottom-up and top-down inferences* [7, 34, 35] and *ensemble learning* [17, 31]. In this study, we focus on the techniques based on ensemble learning.

1.3 Ensemble Learning and Questions about its Performance

Ensemble learning, a pattern classification approach, creates a new classifier via combining different classifiers. Ensemble learning aims to create a new classifier whose performance exceeds the individual classifiers participating into the formation of it. They are also named as *meta-algorithms* since they subordinate algorithms as variables and replaceable parameters and define a class of algorithms. There is a number of different ensemble learning methods which employ supervision. These include Bayesian averaging, bagging, boosting and stack generalization [17, 31, 53]. In this thesis, we focus our attention to the techniques derived from stacked generalization. Although the theoretical analysis of the performance increase realized by the available ensemble learning algorithms are not made thoroughly, widely known four main reasons to combine different classifiers can be listed as follows [29]:

- The ability for a designer to utilize a number of different classifiers for the same classification task.
- The option to make use of more than one single training set, even different feature types for different training sets.
- The ability of different classifiers to be utilized in different regions of the feature space; these methods are known as dynamic classifier selection and it will be detailed in Section 2.1.4.
- The possibility of using the same classifier, such as artificial neural networks, with different parameter initializations as it inherits randomness in the training procedure.

Three major issues that ensemble learning has to deal with can be summarized as in follows.

- **Classifier Selection:** Selecting the type and the number of classifiers to put together is an art rather than engineering. It requires a lot of trial and error to find an “appropriate” set of classifiers, which improves the performance of the ensemble system. The number of classifiers to be assembled is another crucial parameter. While a small set of classifiers result in poor generalization ability, gathering a large variety of classifiers may cause overtraining. It is clear that the design of the ensemble of classifiers is a combinatorially explosive task.
- **Description Selection:** Deciding on a subset among a number of different descriptors, which attains better generalization ability than other subsets is an optimization problem with combinatorial complexity. Also, the poor generalization and overtraining dilemma is found in the selection of descriptors. Additionally, the “best” set of descriptors can only be formed by an exhausted search.
- **Trainable or Non-Trainable Combination:** The classifiers can be grouped into two categories, either *trainable* or *non-trainable* [31]. A trainable combination needs additional training after the combination is realized. However, a non-trainable combination does not necessitate training after the training of individual classifiers. For instance, bagging and boosting type classifier combination schemes are trainable since both individual classifiers and the whole system is trained, while majority voting and plurality voting are non-trainable because only the individual classifiers are trained.

Among different classifier combination schemes, *stacked generalization architectures* [53] needs special emphasis because of their generalization performances noted in different application domains and over-performing results compared to other classifier combination schemes [2, 48, 49]. To reduce the *semantic gap* between low level features and high level information, a hierarchical system based on the stacked generalization is proposed by Uysal et al. [49]. The proposed system is shown to be remarkably more successful compared to the individual classifiers utilized in the base layer. However, the stacked generalization architectures still continue to suffer from the problem of *curse of dimensionality* whose definition is given in Section 1.2.

1.4 Contributions of the Thesis

The contribution of this thesis is two-folded.

1. We propose a new meta-algorithm, named “unanimous decision”, as an alternative to stack generalization architectures to reduce the effect of the *curse of dimensionality* problem. It is well-known that in the meta-layer of stacked generalization, the number of feature space dimensions increases as the number of categories and the number of individual classifiers increase. Because of this fact, the stacked generalization architectures are sensitive to dimensional curse.
2. During the experiments with stacked generalization, we noted that boosting the performance of the individual classifiers does not necessitate a performance increase in the combined version. Then, we proposed a set of criteria for the necessary set of conditions for the success of stacked generalization architecture.

The rest of the work is organized as follows. Chapter 2 provides a literature survey on ensemble learning with a special emphasis on stacked generalization. In Chapter 3, the newly proposed architecture and the set of criteria for the success of stacked generalization are introduced. The experimental results on both the performance of the new architecture and the new set of criteria for stacked generalization are given in Chapter 4. Finally, the summary of the thesis and further research directions are presented in Chapter 5.

CHAPTER 2

RELATED WORK IN ENSEMBLE LEARNING

This chapter will provide a background for the ensemble learning techniques emphasizing particularly the stacked generalization (SG) in the literature. In the scope of the thesis, the concepts, the historical background of SG architecture will be first presented. Then, the problems which need further research will be discussed.

2.1 Ensemble of Classifiers: Is It Better to Use More Than One Classifier?

In the following subsections we first define individual classifier and major types of individual classifiers. Then, the problems related to individual classifiers and assembling the classifiers will be examined.

2.1.1 Individual Classifiers

Statistical pattern classifier is a function that maps a set of discrete and/or continuous values to a discrete set. The domain of the function is called *feature space* and range of the function is called *label set*. Let the true function be denoted by f ,

$$f : \mathcal{F} \rightarrow \mathcal{S} \tag{2.1}$$

where $\mathcal{F}$ is the feature space and $\mathcal{S}$ is the set of all possible labels in the specific classification problem. It should be noted that f may not be the true function, but the optimal function from $\mathcal{F}$ into $\mathcal{S}$ with respect to some error criterion, mostly 0-1 loss function [18], since exactly

different samples may overlap in the feature space. The regions in the feature space, where the samples overlap is called *overlapping regions*.

The aim of classification is to approximate f with another function h , called *hypothesis*, with finitely many samples $T = \{(x_i, y_i) : x_i \in \mathcal{F}, y_i \in \mathcal{S}, f(x_i) = y_i, i = 1, 2, \dots, N\}$, where T is the training set and each tuple (x_i, y_i) is a training sample with feature vector x_i and label y_i . h is also a function from $\mathcal{F}$ to $\mathcal{S}$.

$$h : \mathcal{F} \rightarrow \mathcal{S} \quad (2.2)$$

Better classifier result, hypothesis, is achieved when h approximates f closer according to a predefined metric. In the literature, the performances of a hypothesis commonly predicted by means of utilizing a test set, $\mathfrak{T} = \{(x_i^*, y_i^*) : x_i^* \in \mathcal{F}, y_i^* \in \mathcal{S}, f(x_i^*) = y_i^*, i = 1, 2, \dots, M\}$, and then counting the number of successfully classified samples:

$$Performance(h) = |\{x : h(x) = y, (x, y) \in \mathfrak{T}\}| \quad (2.3)$$

Let c be the number of classes in $\mathcal{S} = \{w_1, w_2, \dots, w_c\}$. A classifier produces a likelihood value, $p(w_i|x)$, for each class based on the provided sample, x , $i = 1, 2, \dots, c$. Each likelihood value show the probability of the sample to belong to this class. The classification prediction of the classifier is found by evaluating the likelihood values produced by the classifier. The class label with the maximum likelihood assignment is the prediction of the classifier.

$$w^* = \arg \max_{w_i \in \mathcal{S}} p(w_i|x) \quad (2.4)$$

where w^* is the class label to be assigned to the sample x .

2.1.2 Major Individual Classifiers

In the literature, a number of different classifiers are proposed to serve the same purpose. However, most of them differ with respect to statistical properties of the underlying information that they analyze to classify a sample. According to the literature survey reported by A. K. Jain et al. [29], some of the major classification methods are listed as follows.

- **Nearest Mean Classifier:** Each class is represented by the mean vector of its training samples. A testing sample is assigned to the class of the nearest mean vector..
- **k -Nearest Neighbor Classifier:** According to a predetermined k parameter value and a given testing sample, k nearest training samples of the test sample is determined.

The unknown sample is assigned to the majority class of these k samples. For the special case $k = 1$, the classifier is named *nearest neighbor* classifier.

- **Template Matching:** Each class is represented by a template, which does not have to be a real data. A new sample is assigned the label of the template which is closest to the new sample. *Nearest mean classifier* is a special case of template matching.
- **Bayesian Classifier:** Based on the prior belief of the class probabilities and the class conditional priorities that are estimated, the aposteriori probability is calculated with Bayes rule. The class label with maximum posterior probability is assigned as the label of the new sample.
- **Linear Classifier:** This type of classifiers is aimed to find a hyperplane that separates two classes. Class label assignment is done based on the side of hyperplane where the new sample lies.
- **Binary Decision Tree:** It is in a form of tree with branch nodes including boolean comparisons. The right branch is followed if evaluation results in true and the left is followed otherwise. Each leaf node carries the class label that will be assigned to the sample in the case that decision branching ends up in that leaf node.
- **Artificial Neural Networks:** They are weighted directed graphs, nodes of which are artificial neurons. The edges are connections between neurons. Neurons are arranged in layered fashion, first of which is named *input layer* and the last is named *output layer*. According to the query sample, the value on the neurons in the output layer gives the estimated label.
- **Support Vector Classifier:** This type of classifiers uses maximizes the margin between the classes with the selection of a minimum number of support vectors. They are kernel methods utilizing kernel functions in order to assure linear separability.
- **Fisher Linear Discriminant:** It is a linear classifier utilizing *minimum square error criterion* (MSE) for the training.
- **Bayes Plug-in:** This classifier assigns each pattern to the category which gives the maximum posterior probability. It is sensitive to the error resulted from density estimation.

- **Parzen Classifier:** It uses Bayes Plug-in rule with the Parzen window method which is used to estimate the densities for each category. It is optimal as the number of samples goes to infinity, namely, asymptotically optimal.

2.1.3 Ensemble of Classifiers

So far in the literature, individual classifier systems provide considerable performance results in different challenging domains such as optical character recognition and face detection [4, 45]. Nevertheless, an individual classifier fixes the system performance to a single hypothesis and also posed to some well known problems as follows:

- **Sufficient Statistics Problem:** The concept was introduced by Fisher [21]. Sufficient statistics is the state that the statistic contains as much information about the parameters as the full data. As a consequence, while the number of mutually exclusive parameters increases, the number of required samples holding required information about the parameters increases exponentially as proposed by Richard Bellman [5]. This problem is, also, known as *curse of dimensionality*. As the parameters are represented as the dimension in mathematical models, the problem is originally caused by exponential increase in the volume by added new dimension to mathematical space of features.

Another way to explain this problem is through the *bias-variance* dilemma. Classifiers with enough flexibility to fit the training data lead to have lower bias, but higher variance, resulting in high training performance but low testing performance. Bias and variance are such a coupled issue that is in inverse proportion for a fixed training set. This problem is strongly related to the *representational problem*. However, as the number of samples in the training set is increased, the possibility of reducing both bias and variance also increases.

- **Representational Problem:** The hypotheses space must include an individual hypothesis that is able to approximate the true function closer. Otherwise, the model will suffer from high variance. Therefore, the used classifiers must be “good enough” to solve the problem instance. For example, if a linear classifier is used in a two-class classification problem that requires a higher order polynomial hypersurface to separate classes, good generalization performances cannot be expected.
- **Computational Problem:** Finding the optimal hypothesis is a search problem and

sometimes it can be in NP-hard class, e.g. estimating the parameters of some forms of Bayesian networks [12]. In these types of problems, heuristic search methods are preferred instead of exhaustive search.

The performance of an individual classifier is restricted to the *hypotheses space*, which is formed by all possible hypotheses which can be generated by the classifier. Nevertheless, ensemble of classifiers are able to build a new hypothesis, other than the ones in the hypotheses space of the individual classifiers, using a subset of this space [18]. An example from Duda et.al.'s work [18] is based on an ensemble learning method known as boosting, which is explained below, is shown in Figure 2.1. In this illustration, the problem is a simple two-class classification with red and black training samples in two dimensional feature space as illustrated in Figure 2.1.a. The classification algorithm in this example is linear discriminant classifier and three hypotheses are shown in Figure 2.1.b,c,d. Figure 2.1.e shows the result of boosting where the created hypothesis is not a result of linear discriminant function. Although this hypothesis is not in the space of all possible outputs of linear classifiers, it is built by ensemble learning architecture combining the individual classifiers of the same type.

The combiners are grouped according to their structure. They can be in *parallel*, *cascading* or *hierarchical* structural forms. In the parallel combination architectures, all the individual classifiers produce results individually, that is, independent of each other. Then, a combiner aggregates outputs of these individual classifiers and provides the final decision. Mostly known combination methods employing parallel combination architectures in the literature are majority voting, boosting and stacking [31]. Another example for parallel combination architectures is the weighted combination of the outputs of the individual classifiers with a gating device. In this type of combiner, depending on the given sample, the gating device determines the weights of the individual classifiers.

In the cascading type of classifiers, the individual classifiers are invoked linearly [50, 51]. Because of the linear nature of this structure, the combination scheme is also called *serial combination*. For a given pattern, the number of possible classes decreases in every invocation progressively. Therefore, the classifiers that are not accurate, but lightweight are feasible options for the first few steps since the number of possible classes is high, which demand more computing power. However, towards the final classifiers it is better to invoke more accurate and expensive classifiers to obtain the final decision efficiently and accurately [29].

In the hierarchical type architectures, a tree-like line up of the classifiers is realized. Therefore, this kind of combiners is also called *tree-like combination architectures*. This

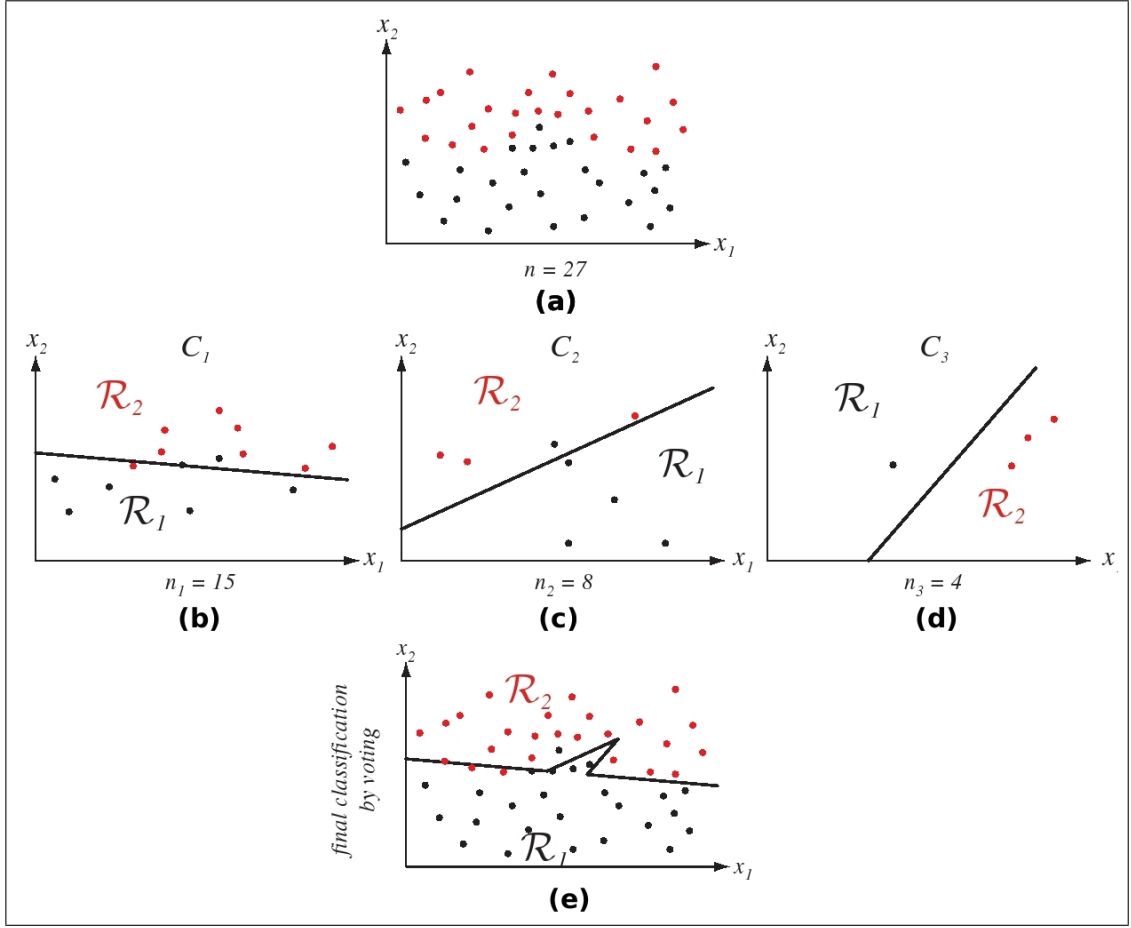


Figure 2.1: Illustration of boosting's ability to create new hypothesis that is not in the hypotheses space of the classifier, adapted from [18].

architecture is similar to the decision trees but it may use more complex classifiers in the nodes with more than one feature. This classifier combination scheme is efficient in utilizing the discriminative power of different types of features.

The non-overlapping nature of sets of misclassified samples by the individual classifiers inspires the idea of combining the classifiers. This idea improves the performance compared to the individual classifiers [53]. Nevertheless, the way to put the classifiers in an ensemble is still an up-to-date research topic with the specified problems which is discussed in Section 1.3. It is a known fact that all combination schemes do not give an increase in the performance. If they do, pattern combiners lead a contradiction with the *no free lunch* theorem, which is mentioned in Section 1.2 [18]. Some theoretical studies make critical assumptions on the individual classifiers. For example, for the Adaptive Boosting technique, the individual classifiers have to perform better than random guessing [22].

2.1.4 Major Classifier Combination Methods

There are a variety of the methods to combine a set of classifiers. In this section, the general overview of the literature on the classifier combination methods will be summarized.

- **Majority voting:** This is a method to get the final result among different classifiers that are trained individually. The predicted label of a given sample is the label which is predicted the same by the most of the classifiers. Let $h_1, h_2, \dots, h_n$ be the hypotheses that are produced by the classifiers. For the sample x , the prediction of majority voting of these hypotheses is

$$MV(x; h_1, h_2, \dots, h_n) = \arg \max_{w_i \in S} \text{histogram}(h_1(x), h_2(x), \dots, h_n(x)) \quad (2.5)$$

where $S = \{w_1, w_2, \dots, w_c\}$ is the label set [31].

- **Plurality voting:** Training of the classifiers are done individually as in majority voting. For hypotheses $h_1, h_2, \dots, h_n$ and given sample x , the prediction is done as follows.

$$PV(x; h_1, h_2, \dots, h_n) = \arg \max_{w_i \in S} \sum_{j=1}^n p_{h_j}(w_i|x) \quad (2.6)$$

where $S = \{w_1, w_2, \dots, w_c\}$ is the label set.

- **Bagging:** It is a “meta-algorithm” that is used to improve the classification accuracy and stability with reducing the variance, known as an acronym for *bootstrap aggregating*. A meta-algorithm is an algorithm having subordinate algorithms as variables and replaceable parameters, in other words it defines a class of algorithms. Bagging can be taken as a special case of model averaging [18]. Given a training set $\mathcal{T}$ of size N , L number of new training sets, $\mathcal{T}'_i$, are created with cardinality N' using uniform sampling from set $\mathcal{T}$, $i = 1, 2, \dots, L$, $N' < N$. Creation of L number of training sets leads L different hypothesis, h_i , $i = 1, 2, \dots, L$. Bagging is nothing but the voting among h_i s, $i = 1, 2, \dots, L$. The model is formalized in Algorithm 1.
- **Boosting:** It is also a meta-level algorithm that is proposed after the question of whether weak hypotheses can create a strong hypothesis, where a weak hypothesis is defined to be a hypothesis that has little correlation with the true hypothesis while a strong hypothesis is one considerably correlated with the true one. For the training set $\mathcal{T} = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, the first step of the algorithm randomly selecting $n_1 < N$ number of samples from $\mathcal{T}$, let it be $\mathcal{T}_1$, and generating a hypothesis h_1 ,

Algorithm 1 Bagging

Require: A training set $\mathcal{T}$ of size N and number of training sets to be resampled, L , and cardinality of each set N' . A classifier C

Ensure: A final hypothesis h

- 1: **for** each $i = 1, 2, \dots, L$ **do**
 - 2: create $\mathcal{T}'_i$ by resampling N' samples from $\mathcal{T}$.
 - 3: produce hypothesis h_i by training classifier C with $\mathcal{T}'_i$
 - 4: **end for**
 - 5: create h by majority voting of h_i for $i = 1, 2, \dots, L$
-

which is called a weak hypothesis. The second training set to be resampled, $\mathcal{T}_2$, must be restricted so that half of the samples in it can be correctly classified by h_1 . The purpose of this step is to generate a subset of $\mathcal{T}_2$ which is “most informative” given h_1 [18].

A major and the most popular variant of boosting is Adaptive Boosting (AdaBoost) which was formulated by Y. Freund and R. Schapire[22]. It works with a classifier that is able to classify weighted samples in such a way that its success is better than chance in all the cases given a training set and weighting of its samples. The term *adaptive* comes from the ability of the algorithm to add new weak hypotheses until the training error is as low as desired. The algorithm is provided below.

- **Stacked Generalization (SG):** SG is a general name for a set of combiners that are ordered in a layered fashion [53]. For two-layered SG architectures, likelihoods produced as the output of the individual classifiers are concatenated. This concatenation yields a vector of real numbers in the range $[0, 1]$. This resulting vector is used as the feature vector for the upper layer. In the SG architectures with more than two layers, the information flows from the bottom layer to the top layer of the architecture layer by layer. In this structured type of stacked combination, the overall structure looks like a tree with a classifier in each node. In this tree-liked organization, leave classifiers obtains real feature measurements. However, non-leaf classifiers get the concatenation of the likelihoods produced by the classifiers that are its children. Since, the method proposed in the thesis is based on the SG architecture, we shall elaborate this issue more in Section 2.2.

Algorithm 2 Adaptive Boosting

Require: Training set $\mathcal{T} = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, classification algorithm $\mathfrak{C}$ to create weak hypotheses, distribution D over $\mathcal{T}$ and the number of weak hypotheses desired K

Ensure: a strong hypothesis h

Initialize the weight vector based on the distribution D , $w_i^1 = D(i)$ for $i = 1, 2, \dots, N$.

Repeat [1-5] **for** $t = 1, 2, \dots, K$

1: Set

$$p^t = \frac{w^t}{\sum_{i=1}^N w_i^t}$$

2: Create a weak hypothesis h_t by $\mathfrak{C}$ with the distribution p^t ; $h_t : X \rightarrow [0, 1]$.

3: Calculate the error of h_t : $\varepsilon_t = \sum_{i=1}^N p_i^t |h_t(x_i) - y_i|$.

4: Set $\beta_t = \frac{\varepsilon_t}{(1-\varepsilon_t)}$.

5: Set the new weights vector

$$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$$

Output the strong hypothesis

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T (\log \frac{1}{\beta_t}) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T (\log \frac{1}{\beta_t}) \\ 0 & \text{otherwise} \end{cases}$$

Dynamic Classifier Selection

There is a class of classifier combination methods called *dynamic classifier selection*. Different from the ensemble learning algorithms mentioned so far, these methods do not fuse the results of the individual classifiers, but instead, there exists an “oracle” to determine an expert among a set of classifiers to classify the provided sample. The decision of this selected expert is taken as the decision of the combined system. Since the selection of the expert is done based on the given sample, this set of methods are called *dynamic*. This type of classifier combination is different from other types inasmuch as this type selects one classifier among a set of them while others uses the predictions of them all.

A dynamic classifier selection method must measure the appropriateness of the classifiers in hand for any sample. For example, the information entropy of the resulting likelihoods of a classifier for a sample can be used as a measure of competence of the classifiers. It would be preferred to select the classifier that outputs likelihood values of which the entropy is the lowest. The classifier with the lowest entropy of likelihoods means that this classifier is more

willing to select a class compared to willingness of other classifiers.

Let C denote the appropriateness of a classifier h_i , for the pattern provided x , then

$$C(h_i|x) = Entropy(\{p_h(w_j|x) : j = 1, 2, \dots, c\}) \quad (2.7)$$

where there are c number of classes, $W = \{w_1, w_2, \dots, w_c\}$. Entropy of a discrete random variable X , with possible values $\{x_1, x_2, \dots, x_n\}$, is defined as follows [46].

$$Entropy(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (2.8)$$

The expert with the maximum competence is selected for the provided pattern. Therefore, for the select classifier, h^* , among the set of classifiers $H = \{h_1, h_2, \dots, h_m\}$, for the pattern x is done as follows.

$$h^* = \arg \max_{i=1,2,\dots,m} C(h_i|x) \quad (2.9)$$

The term *dynamical classifier selection* is introduced by Woods et al. [54] and it was attracted the attention of pattern recognition community to these methods. In this work, the competence of a classifier for a test sample is measured in two different ways. One of them is *decision independent* while the other is *decision dependent*.

- **Decision Independent:** The classifier's performance in a local neighborhood is taken as the competence, and named *overall local accuracy*. The performance is evaluated by considering the nearest k training samples.
- **Decision Dependent:** In this competence measurement, the class prediction is done according to the prediction of the classifier. Let the prediction be w^* . The percentage of the correctly classified local training samples that are assigned to class w^* is utilized as the measure. This measurement is named *local class accuracy*.

In the work of Woods et al. [54], these two selection criteria overperform the individual classifiers. Also, it is noted that the dynamic classifier selection's performance is well enough to outperform classifier fusion algorithms.

Dynamic classifier selection is highly used with classifiers arranged in cascade-like structure [32, 3, 19, 23]. When the competence of the first classifier is high enough, the class prediction of the first classifier is taken as the prediction of the whole system. Unless it is high enough, for example less than a threshold, the competence of the next classifier in the

cascade is calculated. This process goes till a classifier appropriate enough is found. If there is not a classifier competent enough, the system may choose to select the most likely class anyhow.

2.2 Stacked Generalization

In the next two subsections the details and problems of stacked generalization (SG) architecture will be discussed.

2.2.1 The Architecture

SG is an ensemble learning technique which was first proposed by D. H. Wolpert in 1992 [53]. It is a layered architecture working in bottom-up fashion. SG is an ensemble learning technique where layered mechanism plays an important role. There is not a restriction on the number of layers to be used as proposed by Wolpert [53]. In the algorithm, each layer includes a bunch of classifiers. The classifiers at the bottom layer produce likelihood associated to each sample and these likelihood values are fed to the second layer. Each layer gets the likelihoods produced by the architecture of one level below. If there exist more than one classifier in the previous level, the likelihood results are concatenated and provided as the feature vector of the sample for the upcoming level. Figure 2.2 shows the schematic structure of SG. In the representation, a two layered architecture is shown with K number of classifiers and c number of classes. In the figure, there are modules named *descriptor i* , $i = 1, 2, \dots, K$. A descriptor is a group of features which are semantically relevant to each other, for example mean red, green and blue features can be taken as a descriptor. In the SG architecture, these descriptors do not have to be different. The same descriptors can be used with different classifiers.

In the two-layered architecture, the creation of training samples for the second layer is done with N -fold cross validation, where the training dataset is divided into N number of mutually exclusive, collectively exhaustive subsets. These sets include approximately the same number of samples. In the cross validation, one set is kept as a test set, while others used as training set. The training samples for the second layer corresponding to this one fold is produced with training the base classifiers with the other $N - 1$ folds.

The architecture can consist of either the same type or different types of classifiers. The first type of the SG method is called “homogeneous” while the other is called “heterogeneous”.

The SG architecture is applied to two types of pattern recognition problems successfully:

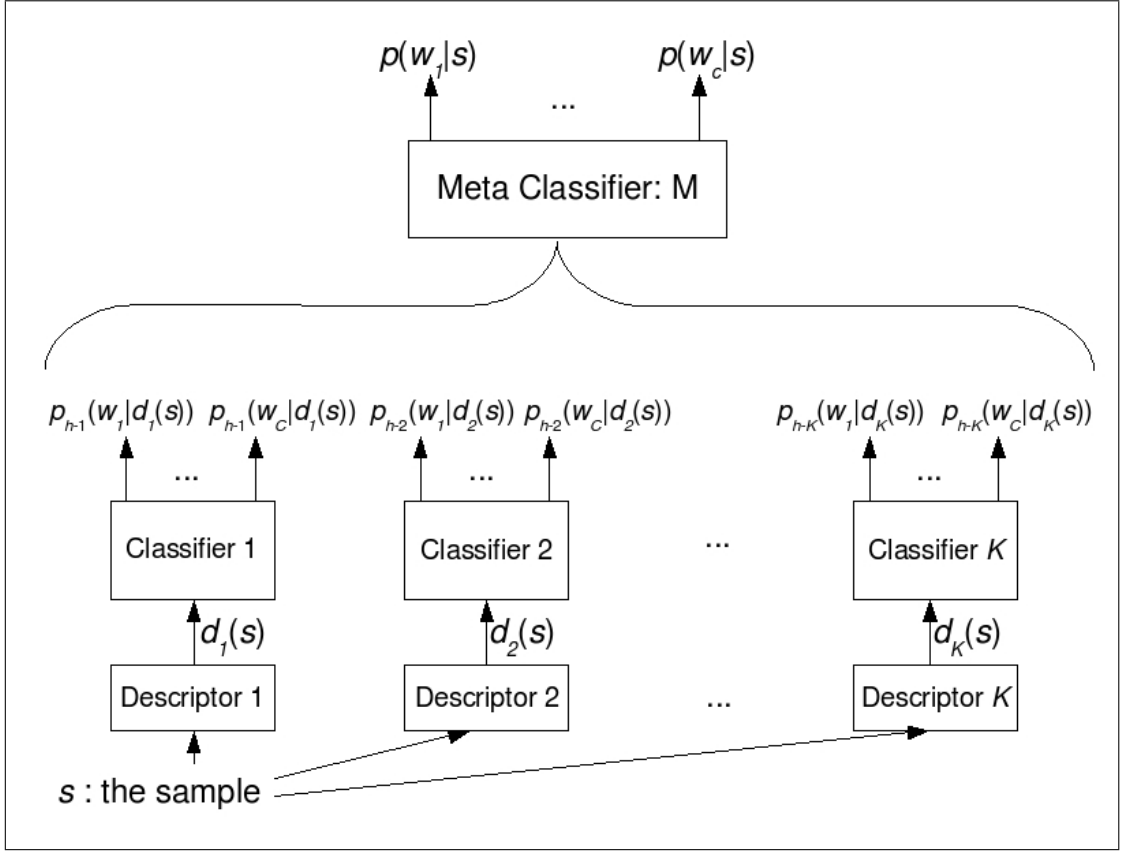


Figure 2.2: Illustration of a general two-layered stacked generalization architecture.

1. **Classification:** Two-layered SG architecture is most widely studied technique and attracted great attention with good generalization performances [42, 47, 48]. Ting et al. [47] try to find an answer with empirical study to the question of what type of level-1 data and level-1 generalizer the stack generalization provide improved accuracy. In the work, ten different datasets from UCI repository are used with three different level-0 classifiers: C4.5, a type of decision tree learning algorithm; NB, naive Bayesian classifier; and IB1, a modification of k -nearest neighbors algorithm. Level-1 training data is created by N -fold cross validation as it is in [53], and used in two different ways: 0-1 classification results and likelihood outputs of level-0 classifiers. As the level-1 generalizers, C4.5, NB, IB1 and a multi-response linear regression (MLR) algorithm are used. MLR is a least square regression algorithm and used as the level-1 generalizer applied to the training data of each class differently. The concluding results from the experiments are as follows.

- Using likelihood values as the level-1 data results in a better generalization per-

formance than using output class predictions.

- Only MLR is appropriate to be the meta classifier, namely, level-1 generalizer.
- It is not necessary to put the constraint of non-negativity to the regression coefficients as opposed to the finding stated in [8].
- When the performance of SG is significantly higher than the individual classifiers participating in the formation of SG, SG architecture tends to perform better than majority voting.

As Uysal [48] stated, the architecture is well enough to overperform state of the art pattern classification combiners such as majority voting and Adaboost. In [48], the base layer consists of modified version of k -nearest neighbors classifier, which finds out the nearest k number of training samples for a given sample. The predicted likelihood for a class is the relative frequency of the class. And these values of the classifiers are concatenated and fed to the meta-classifier, which is a Fuzzy ARTMAP architecture. Fuzzy ARTMAP is a type of artificial neural network and it is proposed as biologically plausible learning model [11]. In the model, there are two clustering modules named ART-a and ART-b; and a mapping field which associates each cluster with one of the clusters in ART-b.

In this work, one of the experimental setup is on 10-class classification problem with four descriptors: “Color Structure”, “Edge Histogram”, “Gabor coefficients”, “Haar coefficients”. The categories are adapted from Corel image dataset. An image per category used in the work is shown in Figure 2.3.

The architecture proposed in [48] is shown in Figure 2.5. The model consists of m different descriptions fed to the k -NN classifiers while Fuzzy ARTMAP architecture is used as the meta classifier with the labels provided for ART-b module. The model is interesting in terms of the learned internal weights in ART-a part of Fuzzy ARTMAP. The system is shown to be able to learn the degree of importance of each descriptor and for each class. In Figure 2.4, mean recognition nodes for the class “bus” is shown. The peak point corresponding to each descriptor belongs to the weight of “bus” category in this figure. The weights dedicated to this category shows that Haar coefficients are worse than other descriptors in order to discriminate “bus” category.

2. **Automatic Image Annotation:** Automatic image annotation is the problem of assigning a set of words to a query image. It is like captioning or attaching keywords



Figure 2.3: Some of the images from one of the datasets used in [48] with their class label. The dataset includes 10 different categories.

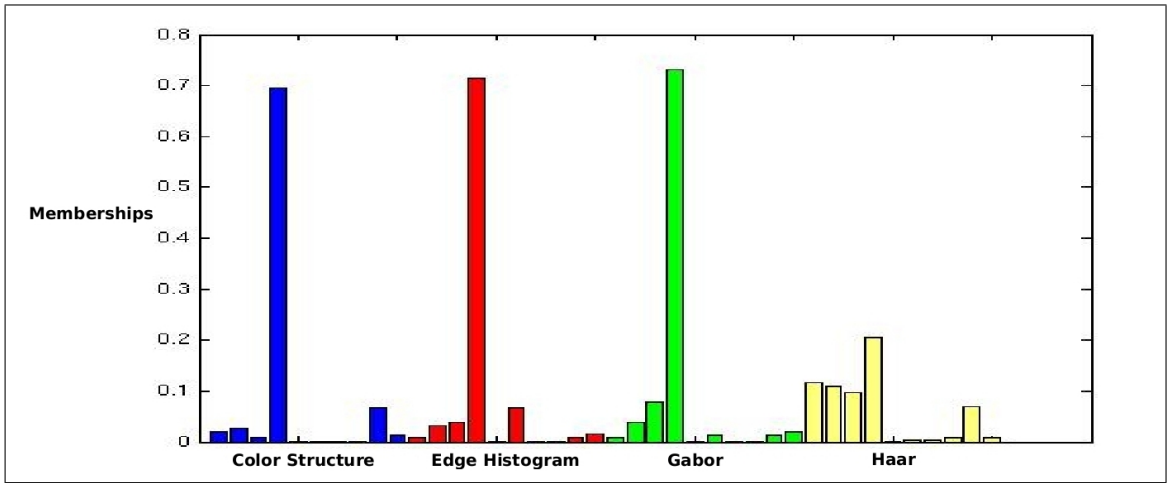


Figure 2.4: The weights of the mean recognition node for the samples of the “bus” category in [48]. This figure is adapted from [48].

to images to explain its content. Although SG architectures are proposed for classification combination technique, it is successfully applied to automatic image annotation problem with supervised learning [1, 2]. In [2], the annotation problem is approached from two different ways: supervised and unsupervised. Supervision is done by SG architecture. In the approach, each image belongs to a category and each category has a set of annotation words. Since the annotation of all images in the same category done with the same words in the training images, this approach reduces human effort to annotate each training images. However, using the same words leads non-trivial annotations for some images. For example, it is not easy to guess the annotation words of some images in Figure 2.6. In this image database, the category named “Europe”

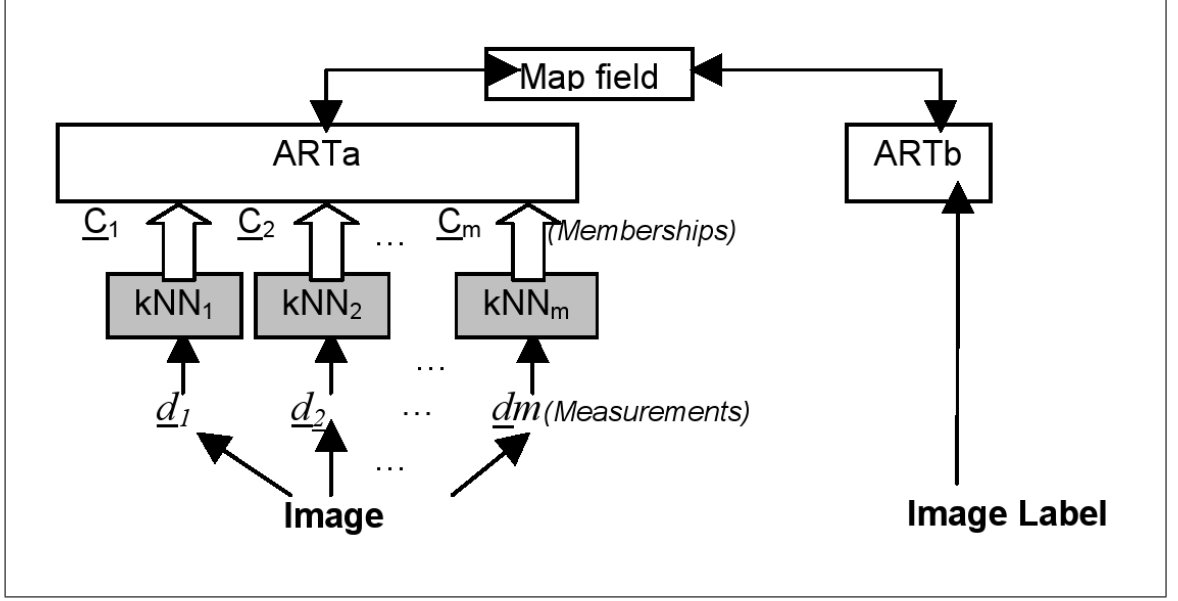


Figure 2.5: The architecture proposed in [48]. The figure is adapted from the same work.

with the manual annotation words “Europe, house, landscape” is shown. For supervision, a two-layered homogeneous SG architecture is used with fuzzy k -NN classifier in both layers. As it is shown in Figure 2.7, the architecture feeds different descriptors of an image to the individual learners, at level-0 classifiers. The resulting likelihood values from level-0 are appended and given to the meta classifier. The best k number of classes are selected by the meta layer. Each word of these categories put into a list with how many times it appears. According to a global metric on the training set, appearance of each resulting word is categorized as either *by chance* or *by not chance*. The words categorized by *not chance* are presumed to be the annotation words of the image. The performance of SG architecture used for automatic image annotation is shown to outperform the methods proposed before this architecture [1, 2].

2.2.2 Problems of the Stacked Generalization Architectures

Based on empirical analysis on ensemble learning, Dietterich et.al [17] stated that a necessary and sufficient conditions for an ensemble learning architecture can be stated as follows:

- Its base classifiers must be more accurate than chance, in other words, individual classifiers must be better than random guessing.
- Its base classifiers must be diverse from each other. This means that two individual hypotheses lead different errors on at least one new sample.

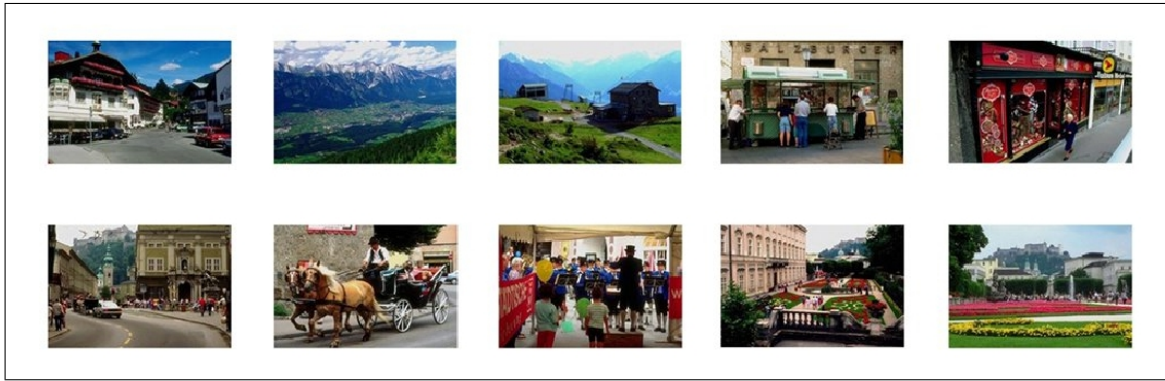


Figure 2.6: Some example images used in the work [2]. These images from one of the “Europe” categories with the annotation words: “Europe, house, landscape”. This figure is adapted from [2].

However, SG is a “black art” as stated in [53]. Additionally, neither a complete theoretical analysis of the performance nor the cases in which it is superior to the other ensemble learning algorithms has been studied yet.

Another serious problem of the SG architectures, is the curse of dimensionality. Since the dimension of the feature space is increased in the top level, the training data is far short for statistical stability. For the two-layered architecture, the dimension of top level is $K|S|$, where K is the number of base classifiers and S is the label space. Therefore, SG suffer from the curse of dimensionality at the meta layer, specifically, when the number of classes are high. The number of label space of base classifiers increases, the system tends to be affected more and more from the curse of dimensionality. This fact is verified in the work of Uysal, where the performance of the system decreases dramatically as the number of classes increases [48]. A similar inclination is valid for the increased number of individual classifiers.

2.3 Chapter Summary

Stacked generalization architectures, the related ensemble learning methods are discussed in this chapter. Also, the problems of SG are stated. Another notification of this chapter is that although great performances of ensemble learning techniques have been noted in the literature, there is not a complete theoretical background on these methodologies. The facts and findings are all based on experimental studies.

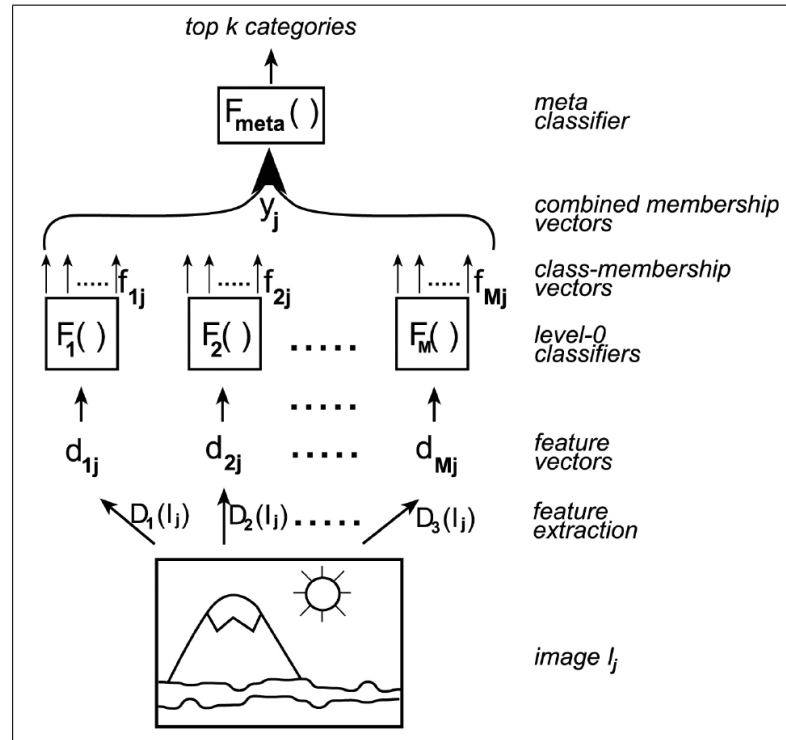


Figure 2.7: Supervised Ensemble of Visual Descriptors, proposed in [2]. This figure is adapted from the same work.

CHAPTER 3

UNANIMOUS DECISION

In this chapter, a novel architecture named “unanimous decision” will be presented. It is proposed as an alternative method to two-layered stacked generalization (SG).

3.1 Motivation

For the classification problem of c classes, two-layered stacked generalization (SG) architecture with K number of individual classifiers faces with the problem of classification in the vector space of cK dimension of features. As a known fact, exponential dependence of the volume to the number of dimensions leads to the problem called “curse of dimensionality”. It is a serious obstacle in pattern classification problems that requires in very large training data set. Since the dimension of the upper layer in SG depends linearly on both the number of class labels and the number of individual classifiers, it is highly possible to be affected by this problem as both factors about to increase. As it is noted in the work of Uysal [48], the performance of SG decreases as classification problem involves new classes. Figure 3.1 shows the drop of performance of the system as the number of classes increases. Although the general trend is reduction on the performance, there are some cases in which the classification performance increases. This is explained as that in some newly added classes the samples are easy to be classified compared to the samples of other classes. Even, the algorithm shows hundred percent accuracy in classification of some specific classes.

Increased number of classes requires increased number of base level classifiers to meet a satisfactory classification performance. This fact further elevates the curse of dimensionality problem.

Based on this problem of SG, trying to shrink the vector space with which the upper classifier deals is noteworthy. Therefore, the fundamental aim of this novel architecture is

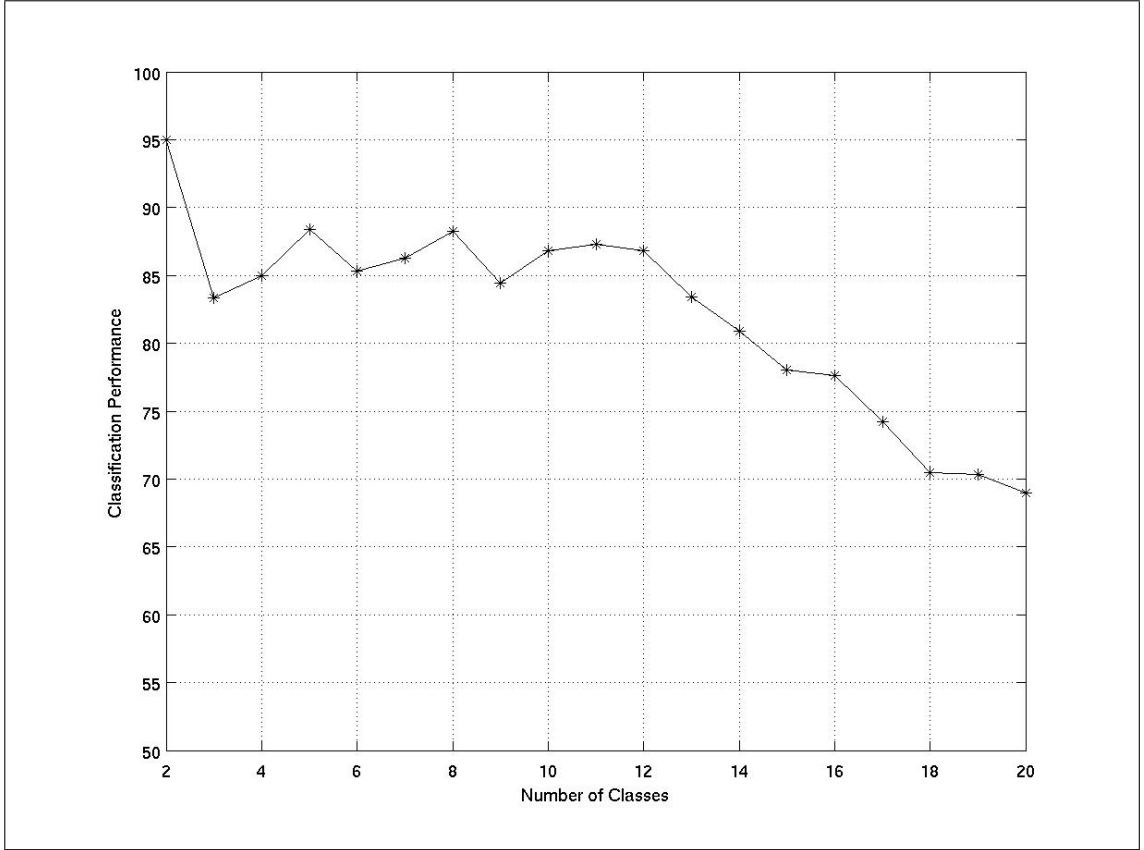


Figure 3.1: The number of classes versus classification performance comparison in [48] in order to show the performance decrease as the number of classes increases.

to attack the problem of high dimensionality in the upper part in order to diminish the tendency of the method to be trapped in curse of dimensionality.

3.2 Unanimous Decision

The architecture consists of a set of basic blocks each of which is itself an ensemble learning architecture with base classifiers and a weighting gate.

3.2.1 Basic Block

The basic block, shown in Figure 3.2 is called β_1 . In β_1 , the classifier C_1 is used as the weighting mechanism of all other classifiers, which are $C_2, C_3, \dots, C_K$. For a training sample s , each classifier receives a different description of s , indicated as $d_1(s), d_2(s), \dots, d_K(s)$. Then, each classifier provides likelihood values based on the description of the sample provided. For the testing sample, s , the likelihood prediction of the block is done based on the following

equation.

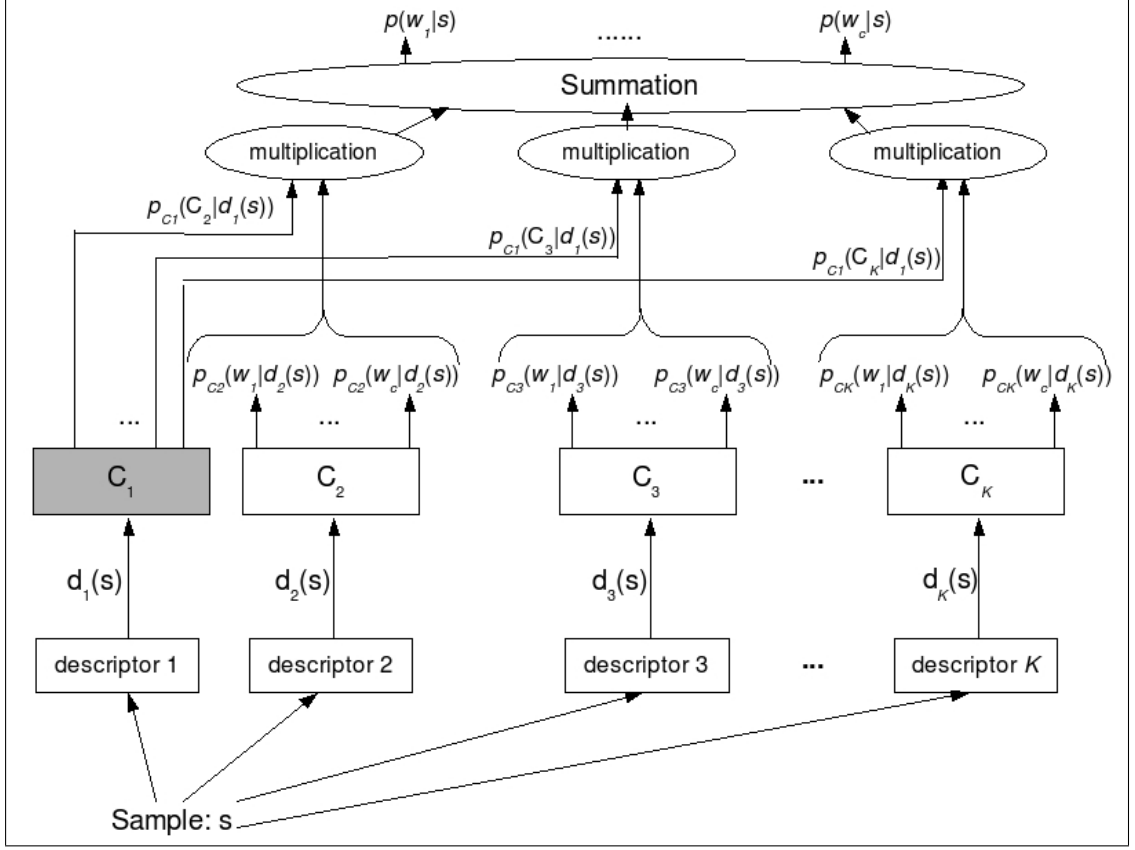


Figure 3.2: The First Basic Block: β_1 .

$$p_{\beta_1}(w_i | s) = \sum_{j=2}^K p_{C_1}(C_j | d_1(s)) p_{C_j}(w_i | d_j(s)) \quad (3.1)$$

where $\mathcal{S} = \{w_1, w_2, \dots, w_c\}$ is the set of all labels; $p_{\beta_1}(w_i | s)$ is the likelihood of the class w_i from the building block β_1 while $p_{C_1}(C_j | d_1(s))$ denotes weighting of the classifier C_j , by the prediction of classifier C_1 , for $j = 2, 3, \dots, K$. $p_{C_j}(w_i | d_j(s))$ is the likelihood of the class w_i from the classifier C_j for $j = 2, 3, \dots, K$.

The definition of $\beta_2, \beta_3, \dots, \beta_K$ are similar to the definition of β_1 . As a general case, the formula of β_k can be given as follows, for $k = 1, 2, \dots, K$ and β_k is depicted in Figure 3.3.

$$p_{\beta_k}(w_i | s) = \sum_{j=1, 2, \dots, K; j \neq k} p_{C_k}(C_j | d_k(s)) p_{C_j}(w_i | d_j(s)) \quad (3.2)$$

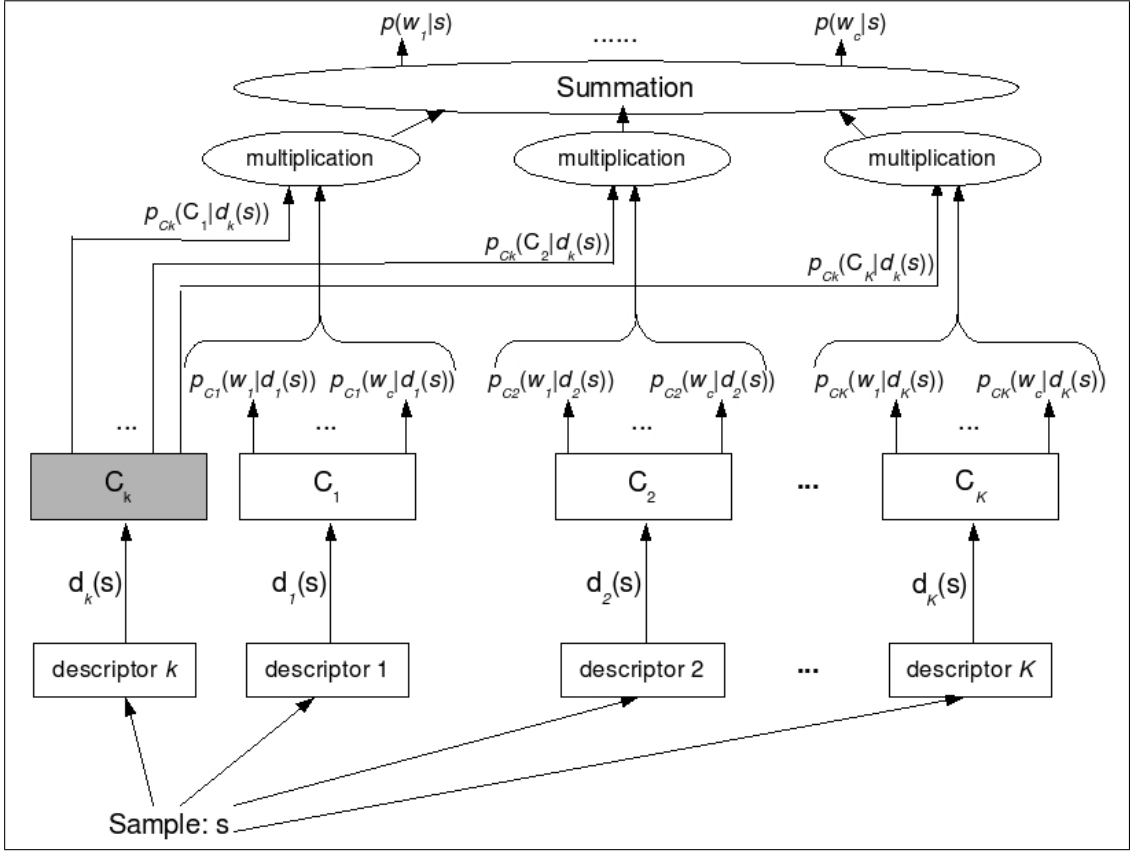


Figure 3.3: The k^{th} Building Block: β_k .

where the weighting of decisions of the base classifiers are determined by the weighting mechanism. According to the mechanism, the weights are assigned based on the previous samples. In the equation $p_{\beta_k}(w_i|s)$ is the prediction likelihood of the class w_i from the building block β_k , while $p_{C_k}(C_j|d_k(s))$ is the weight of the classifier C_j , which is produced by p_{C_k} , based on the descriptor space $d_k(s)$; $p_{C_j}(w_i|d_j(s))$ is the likelihood of the class w_i from the classifier C_j , based on the descriptor space $d_j(s)$ based on the descriptor space $d_k(s)$ for $k = 1, 2, \dots, K$, $i = 1, 2, \dots, K$ and $j = 1, 2, \dots, K$; $j \neq k$.

Training of a Block

For the training of each block β_k , with training sample s and label w_j , each individual classifier other than C_k is trained as usual, for $j = 1, 2, \dots, c$ and $k = 1, 2, \dots, K$. C_k is trained so that it can produce a similar result to $p_{C_i}(w_j|d_i(s))$ for the weighting value corresponding to i -th classifier. This is exactly the same thing as the likelihood of the classifier C_i for the true label, w_j , for s in the description space d_i . Intuitively, the probability value $p_{C_i}(w_j|d_i(s))$ is a meaningful measure to be used for the reliability of classifier C_i for the training sample

$(d_i(s), w_j)$. A classifier with low accuracy for that sample leads low reliability for the classifier C_i , and as a result, C_i gets reduced attention for that sample. Of course, the reverse is held true, if C_i is able to produce high likelihood for label w_j when $d_i(x)$ is provided.

Since all output values are aimed to be probability values, the necessary normalization operations are important to be done; for instance, for the training of C_k in β_k , the provided vector $[p_{C_1}(w_j|d_1(s)), p_{C_2}(w_j|d_2(s)), \dots, p_{C_{k-1}}(w_j|d_{k-1}(s)), p_{C_{k+1}}(w_j|d_{k+1}(s)), \dots, p_{C_K}(w_j|d_K(s))]$, has to be normalized for the training sample (s, w_j) .

3.2.2 The Overall Unanimous Decision Architecture

The overall system is the weighted combination of all building blocks, $\{\beta_1, \beta_2, \dots, \beta_K\}$. The likelihood can be formulated as follows for an unknown sample s .

$$p(w_i|s) = \sum_{j=1}^K \alpha(j) p_{\beta_j}(w_i|d_j(s)) \quad (3.3)$$

$\alpha(j)$ values are weighting values for the likelihood value corresponding to building block β_j , for $j = 1, 2, \dots, K$. These values can be estimated dynamically with training-testing mechanisms. However, a simpler approach is taken into account in the thesis for the sake of simplicity, which is *equal weighting*, in other words, *plurality voting*: $\alpha(j)$ is taken $1/K$, $j = 1, 2, \dots, K$.

The whole architecture is shown in Figure 3.4. The weighting factors, $\alpha(j)$'s, are shown on the directed edges from the likelihood values of each building block to the final phase, which is weighted summation with this coefficients.

Unanimous decision (UD) architecture proposes a solution to the curse of dimensionality problem because the number of measurements that is input to a classifier is linearly depend on merely the number of dimensions of feature space at the base layer descriptor. But, in SG architecture, the dimension of the second layer is linearly depended on both the number of classes and the number of base classifiers, i.e. the number of descriptors. As the number of classes and/or the number of base classifiers increase this does not cause the dimensional curse problem in UD, whereas SG is sensitive to these numbers.

3.2.3 The Individual Classifiers at Each β_i

As individual classifiers, fuzzy k-nearest neighbors classier is used and from this perspective, the system can be said to be homogeneous.

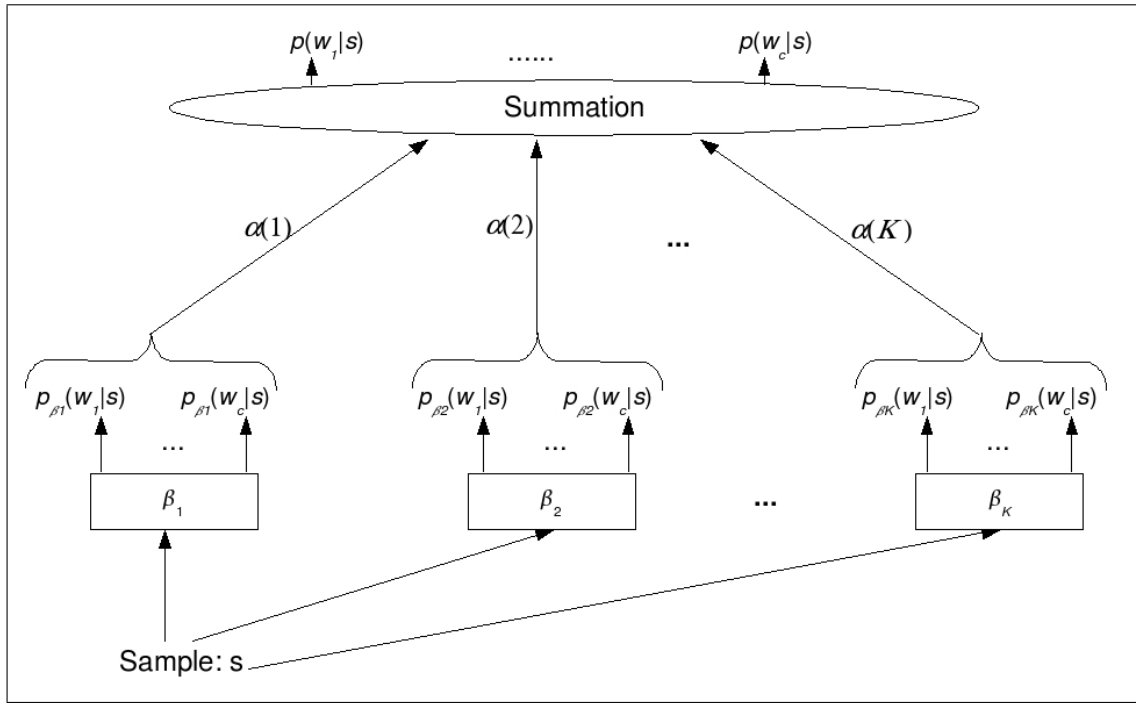


Figure 3.4: Unanimous Decision Architecture.

Fuzzy k -Nearest Neighbors Classifier

Fuzzy k -nearest neighbors classifier was proposed by Keller et al. in 1985 [30]. It is a fuzzified extension of classical k -NN.

The classical k -NN classifier is an instant-based learning algorithm, which is also known as *lazy learning*. In this classifier, the given query sample x 's k nearest samples from the training set, T , are determined according to a predetermined distance metric. After the identification of k points, among the labels of these training points, the most frequent label is assigned as the label of the new sample. Whereas in fuzzy k -NN, each training sample does not have an exact label but there is a membership of being labeled by each classes. Based on this information and inversely weighting the distance of k nearest samples to x , the output is a likelihood vector.

The formulation of Keller et al. is as follows, where x is the sample whose label wanted to be learned, $T = \{(x_i, y_i) : x_i \in \mathcal{F}, y_i \in \mathcal{S}, i = 1, 2, \dots, N\}$ is the training set with $\mathcal{F}$ feature space and $\mathcal{S} = w_1, w_2, \dots, w_c$ label set. Class memberships vector $y(x)$ is computed as

$$y(x) = [y_1(x), y_2(x), \dots, y_c(x)] \quad (3.4)$$

$$y_i(x) = \frac{\sum_{j=1}^k y_i(x_j) \|x - x_j\|^{\frac{-2}{p-1}}}{\sum_{j=1}^k \|x - x_j\|^{\frac{-2}{p-1}}}, i = 1, 2, \dots, C \quad (3.5)$$

where the k nearest points in T is denoted by x_j , $j = 1, 2, \dots, k$ and $y_i(x_j)$ is the membership value of sample x_j to the label to class w_i . In the formulation, p is known as the scaling factor to control the effect of the distance between x and x_j on calculated weight values. p is taken 3 in this work.

3.3 Chapter Summary

In this chapter, a novel classifier combination method, “unanimous decision”, is proposed as an alternative to two layered SG. The aim of the system is to get rid of curse of dimensionality problem in the upper part of SG architecture. It is notable that as SG’s dimensionality depends on the multiplication of the number of classifiers and the number of classes, new algorithm’s number of dimensions is the same as dimensions of the feature spaces that individual classifiers have to deal with.

CHAPTER 4

EMPIRICAL STUDY

The experimental setup and the empirically verified facts will be presented in this chapter. This chapter has aimed:

1. To analyze the performance of “unanimous decision” (UD) compared to the stacked generalization (SG) using the same classifiers with different datasets.
2. To analyze the proposed set of criteria for the better classification performance of SG.

4.1 Data Set

In this work, some subsets of Corel Draw image database are used. This dataset is published by Corel Corporation. It contains nearly 60000 images kept in 600 CD-ROMs each of which contains approximately 100 images. Images in each CD-ROM are related to a concept, so each CD-ROM can be assumed to be a category. However, there are similarities between some classes while there is not a conceptual relation within some classes. Because of this reason, some subsets of these classes are included in the experiments as the working datasets.

Each category contains nearly 100 images, half of which are taken as the training set. The other half of images in each category are kept for testing. “Unanimous decision” architecture is tested with 5, 10, 15, ..., 40 categories. Sample images from these categories are presented in Appendix A. The categories used in the experiments are taken in the same order as they presented there.

4.2 Base and Weighting Classifiers

Fuzzy k -nearest neighbors classifiers are used in the experiments as base and weighting classifiers. The formulation of the classifier can be found in Section 3.2.3. The reasons to

select fuzzy k -NN classification algorithm can be given as follows.

- **Easy to Implement:** Fuzzy k -NN classifier is easy to implement.
- **Having Less Number of Parameters:** In general, the classification algorithms have lots of parameters to estimate. For instance, in artificial neural networks, number of internal nodes, type of activation function, and type of training method are all parameters to be specified. However, fuzzy k -NN has merely two parameters which are k : the number of neighbors to be taken into account, and p : the scaling factor. For the estimation of k , leave-one-out cross validation is used while p is fixed to 3.
- **Easy to Train:** Fuzzy k -NN classifier does not have training phase. The algorithm relies on just the data that is in hand.

In order to compare the proposed architecture with stacked generalization (SG) architectures, fuzzy k -NN classifiers are also used as the individual classifiers and the meta-classifier for SG.

4.2.1 Parameter Estimation

In order to achieve the maximum performance, the parameters of the classification algorithms is estimated using resampling theory. Some of the methods to estimate the parameters are N -fold cross validation, jackknife method and bootstrapping [18]. In this work, leave-one-out cross validation is used. Leave-one-out cross validation is a special case of N -fold cross validation where N is equal to the number of training samples.

In N -fold cross validation, the training set, T , is divided into N number of subsets $F = \{T_1, T_2, \dots, T_N\}$. The cross validation performance of a system is defined to be the average of the N cases where T_i is kept for testing and others are used for training for all $i = 1, 2, \dots, N$. The estimation of the parameter θ using cross validation is done via calculating cross validation performance for the finite subset of possible values for θ . θ does not have to be a single parameter; it can be a vector of parameters. The algorithm is provided below.

In the fuzzy k -NN classifiers of both SG and UD architectures, leave-one-out cross validation is used to estimate the parameter k . The other parameter p in the classifiers is taken fixed at 3.

Algorithm 3 Parameter Estimation with N -Fold Cross Validation

Require: A training set T , the classifier C with the parameter θ to be estimated, finite set of possible values for θ : $\{\theta_1, \theta_2, \dots, \theta_m\}$

Ensure: Estimation result for the parameter θ , $\theta = \theta^*$

- 1: Divide the training N nearly equal sized subsets $\{T_1, T_2, \dots, T_N\}$.
 - 2: **for** each $j = 1, 2, \dots, M$ **do**
 - 3: **for** each $i = 1, 2, \dots, N$ **do**
 - 4: Train C with the unification of sets $T_1, T_2, \dots, T_{i-1}, T_{i+1}, T_{i+2}, \dots, T_N$ and produce hypothesis h_i .
 - 5: Test h_i with the set T_i and keep the performance, p_i .
 - 6: **end for**
 - 7: Set the cross validation performance of θ_j , $p(\theta_j) = (\sum_{i=1}^N p_i)/N$.
 - 8: **end for**
 - 9: assign the θ_i with the maximum cross validation performance as the estimation of θ ,
 $\theta^* = \arg \max_{\theta_i \in \{\theta_1, \theta_2, \dots, \theta_N\}} p(\theta_i)$
-

4.3 Feature Extraction

In the experiments, different subsets of MPEG-7 visual descriptors are used. Total 7 of MPEG-7 descriptors, namely, Color Structure, Scalable Color, Color Layout, Edge Histogram, Homogeneous Textures, Dominant Color, Region Shape are used. In addition to these descriptors, Haar Coefficients [38] are also employed in the experiments.

Moving Picture Experts Groups (MPEG) developed MPEG-7 standard aimed to describe the multimedia content to be interpreted and manipulated by computational device [37]. For the full description and the extraction of the MPEG-7 descriptors, the technical report [28] of the Moving Picture Experts Group can be referred. And Mulcahy [38] gives the details of Haar coefficients. In this work, a brief explanation of each feature groups is provided:

1. **Color Structure:** This color feature descriptor captures two different types of information: color content and the structure of this content. The extraction of this information couple utilizes 8x8 structuring element.
2. **Scalable Color:** Scalable Color constructs the histogram of colors in HSV color space. The Haar transformation is used in order to encode this histogram.

3. **Color Layout:** It uses discrete cosine transformation in order to represent spatial distribution of colors in the image in the YCrCb color space.
4. **Edge Histogram:** It describes the spatial distribution of four directional edges and a non-directional edge information.
5. **Homogeneous Textures:** This descriptor uses Gabor wavelet coefficients which are extracted in six orientations within five scales.
6. **Dominant Color(s):** Dominant Color descriptor uses color quantization to find out a small number of representative colors in the image.
7. **Region Shape:** It is a shape descriptor of set of regions for the image, which is robust to minor deformations along the boundaries.
8. **Haar Coefficients:** This descriptor is extracted by Haar wavelet transformation which is mainly used for image compression and filtering. It uses multi-resolutional simple intensity difference information with different filters applied to the image with convolution.

4.4 Parameter Estimation

The parameter k is estimated utilizing 10-fold cross validation. The estimated k value, 10-fold cross validation performance in the training set and performance in the test set corresponding to the fuzzy k -NN with each descriptor with different number of classes are shown in Table 4.1, Table 4.2, Table 4.3 and Table 4.4. The classification performance of the base classifiers with each of these descriptors in the classification problems with different number of classes are provided in Figure 4.1. The structured nature of the Corel Image Dataset in terms of colors brings higher performances with features describing color distributions in the image. However, the images do not contain a few colors that are dominant and unique to a class, so Dominant Color descriptor results bad performance compared to other color based descriptors. Edge Histogram and Haar Coefficients are notable in terms of relative performances, since they are able to grasp the shape information and Corel dataset consists of classes in each of which has similar shape information. However, as a shape descriptor, Region Shape results in the worst individual performance in all the cases because it is appropriate to describe the shape of binary silhouette images.

As it is notable from the fluctuations in the estimated k values with respect to the number of classes for each descriptor, the estimated value is more stable in the individual classifiers which is with the descriptor resulting in better performance. For example, the fluctuation of the estimated value is in between 8 and 13 for Color Structure descriptor while it is in between 5 and 126 for Region Shape descriptor.

The fuzzy k -NN architectures in the meta-layer part of SG architecture and in the weighting gates of UD architecture are estimated using 10-fold cross validation on the training data generated by the bottom layer. The training data for the upper part of the architecture is generated by leave-one-out cross validation.

4.5 Comparison of Stacked Generalization and Proposed Method

4.5.1 Classification Performances

The performance comparison of UD with SG architecture is done with five different sets of descriptors among the set of possible descriptors, {Color Structure (CS), Scalable Color (SC), Color Layout (CL), Edge Histogram (EH), Haar Coefficients (HC), Homogeneous Textures (HT), Dominant Color (DC), Region Shape (RS)}. These five different sets are taken as {CS, SC, CL, EH}, {CS, SC, CL, EH, HC}, {CS, SC, CL, EH, HC, HT}, {CS, SC, CL, EH, HC, HT, DC} and {CS, SC, CL, EH, HC, HT, DC, RS}. The experiments are conducted with 5, 10, ..., 40 number of classes.

The resulting performances of both architectures in these 40 different settings are presented as a tabular form in Table 4.5 and as a graphic in Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5 and Figure 4.6. In the table, the cases that the performance of UD is greater than SG are indicated with typing the performance bold. As it can be seen, the performance of UD architecture is comparable to SG for the lower number of classes while its performance is better in the most cases for the higher number of classes.

There are increases in the performance of individual classes from 30-class problem to 35-class problem and from 35-class problem to 40-class problem. It is because of the ease to classify the newly added classes compared to the previously used ones. These increases in the performance of individual classifiers leads greater jumps in the classification performances of both UD and SG architectures. As an example for this case, the class-based accuracy of the SG and UD architectures are provided in Table 4.6 and Table 4.7, respectively. It is observable that the last ten classes are relatively easy to be classified except for the 37th class.

4.5.2 Computational Complexities

Besides the classification performance comparison, the comparison of the architectures in terms of computational necessities is crucial. The computational complexities are given from two different points of views.

Training

Although in Figure 3.4 there are K number of different blocks each of which contains K number of different classifiers, a simplification is possible for the architecture. For two different blocks, there are $K - 2$ pairs of classifiers each of which is exactly the same. Therefore, in the simplified version of the architecture, $2K$ classifiers are needed to be trained. K number of base classifiers trained with K different descriptors while other K classifiers are trained with the same set of K different descriptors.

Since the fuzzy k -nearest neighbors classifiers are lazy learning algorithms, the only computational need of both SG and UD is estimating the likelihoods produced by the base classifiers. For one fuzzy k -NN to produce the likelihood of a class for a sample in the training set with leave-one-out cross validation, the computational complexity is

$$O(dN) + O(kN) + O(k) + O(dkC) = \max\{O(dN), O(dkC)\} = O(dN) \quad (4.1)$$

where there are C classes, in the d dimensional feature space with N number of training samples. In the calculation, the first term is to computing $\|x - x_j\|^{-2/(p-1)}$ for all j values in Equation 3.5. The second term is needed to identify k nearest numbers among N different scalar values, while the third and fourth ones needed for the calculation of the denominator and the nominator, respectively for all C classes. In approximately all cases, $k < d$ and $kC < N$. Therefore, to produce level-1 training set, one fuzzy k -NN classifier needs $O(dN)$ time for one sample. So, it needs $O(dN^2)$ time for all training samples. For all K different fuzzy k -NN with different descriptors, the computational time is $O(N^2(\sum_{i=1}^K d_i))$ where d_i is the dimension of the i^{th} descriptor.

For both SG and UD, there are exactly the same base descriptors. Therefore, both architectures computational complexity for training phase with leave-one-out cross validation is $O(N^2(\sum_{i=1}^K d_i))$ where N is the number of training samples with K different descriptors with d_i dimensions, $i = 1, 2, \dots, K$.

Testing

Getting the likelihoods of a query sample from fuzzy k -NN, takes $O(dN)$ time as stated above. In SG, if there exist K base classifiers, the required time for the level-1 feature vector to be produce is $O(N(\sum_{i=1}^K d_i))$ where d_i is the dimension of the i^{st} descriptor. The time necessary for the second layer to output the final likelihoods is $O(NKC)$ where K is the number of base classifiers and C is the number of classes. Therefore, the total amount of time needed for SG in order to find the category of a test sample is $O(N(\sum_{i=1}^K d_i) + NK C)$.

In UD with K building blocks, there exist K base classifiers and K weighting classifier. Total time required for the base classifiers is $O(N(\sum_{i=1}^K d_i))$, again. It is the same for the weighting classifiers. Therefore, the total time complexity is $2O(N(\sum_{i=1}^K d_i)) = O(N(\sum_{i=1}^K d_i))$.

From the analyzes above, it is shown that the computational needs of the both architecture's training are identical. However, the time requirement of the testing steps, differs depending on the ratio $KC/(\sum_{i=1}^K d_i)$. Obeying the aforementioned assumptions, it is true to say that *as the number of classes increases, the asymptotic time necessity of SG increases linearly while this is not the case for UD.*

4.6 Analysis of Stacked Generalization

In this section, SG is analyzed in order to understand what it lays beneath the architecture and what can be done to select the set of descriptors give the best performance for homogeneous SG architecture.

4.6.1 The Mapping

First of all, it must be noted that SG utilizes some classifiers in the base level and each classifier provides a mapping from a feature space to a likelihood space. In order to inquiry SG's ability of mapping, a synthetic dataset having three categories is generated randomly in a two-dimensional spaces. All categories, red, green, blue, have 250 samples for training and 250 for testing. Each sample has two descriptors of it. The generated samples are assumed to be distributed with circular and banana-like models in these two different descriptor spaces. The samples in two different descriptor spaces are shown in Figure 4.7 and Figure 4.8. As the classifiers, fuzzy k -NNs are used with $k = 10$. The output likelihoods of each base classifier is presented in a three dimensional space. Each dimension corresponds to a likelihood value for

a category; red, green or blue. Since these likelihoods values sum up to one for a sample, the samples are distributed in the plane of $red\ likelihood + green\ likelihood + blue\ likelihood = 1$.

The likelihood values produced from the upper layer of the architecture is shown in Figure 4.9. As it can be seen, there is a big difference between the separability of the samples in these different likelihoods spaces shown; likelihoods of descriptor space 1 and 2; and the output of the whole architecture. The output of SG is more compact than outputs of base classifiers and each category's samples are heaped together in the likelihood space of the SG.

4.6.2 A Set of Success Criteria to Select Descriptors for Homogeneous Stacked Generalization

Since the performance of SG is affected by the number of descriptors, it is important to restrict the number of descriptors. Finding the optimal number of descriptors for homogeneous SG architecture is problematic. Even if the number of descriptors is found, the selection of descriptors is another problem. In the case that the number of descriptors to be selected is fixed, a set of criteria for selecting descriptors that give the best possible performance in homogeneous SG architecture is as follows.

- Keeping the base classifiers with descriptors that are able to classify one or more categories well.
- Keeping the descriptors in such a way that for each category there is at least one descriptor that is able to classify it with a high performance.

For the testing of the validity of this set of criteria, synthetic data with four descriptors in two dimensional space is created. The data contains three different classes. The chosen model for each class is normal distribution with the same covariance matrix for all classes. The problem is to select the set of three descriptors which gives the best performance in homogeneous SG with fuzzy k -NN classifiers.

In Figure 4.10, an instance of the data model is depicted in four different synthetic descriptors. Similar configurations are created with ten different random trials. The average performance of each descriptor and the average performance of each possible configuration with 3 different descriptors are shown in Table 4.8. It is shown that although the fourth descriptor is the best individual descriptor, the combination satisfying the two criteria given above results in better classification performance than other combinations.

4.7 Implementation

All algorithms except MPEG-7 feature extraction are realized in MATLAB programming language using MATLAB 7.1 on GNU/Linux 2.6, Pentium 64-bit 2.4 GHz Dual core platform. For the extraction of MPEG-7 descriptors mentioned above, XM (eXperimentation Model) software [27] is used.

4.8 Chapter Summary

This chapter provided the experimental results in both the comparison of the novel method, unanimous decision (UD), with stacked generalization and analysis of stacked generalization architecture. The results of the empirical study showed the followings.

- UD architecture is comparable with SG in low number of classes while its performance is better than the performance of SG.
- Two criteria of selecting descriptors for the homogeneous SG architecture are proposed and shown to be valid.

Table 4.1: Estimated k Values and Corresponding Performances for the Descriptors Color Structure and Scalable Color

Color Structure			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	12	78.80%	77.20%
10	8	62.00%	62.40%
15	8	55.06%	55.20%
20	8	48.60%	49.20%
25	10	45.36%	46.00%
30	13	44.33%	44.14%
35	8	45.71%	44.91%
40	8	46.75%	46.65%

Scalable Color			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	3	72.80%	73.20%
10	4	57.40%	57.80%
15	6	49.06%	50.14%
20	6	42.20%	41.90%
25	10	37.60%	36.96%
30	16	36.80%	36.67%
35	15	37.42%	37.71%
40	10	38.05%	39.60%

Table 4.2: Estimated k Values and Corresponding Performances for the Descriptors Color Layout and Edge Histogram

Color Layout			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	5	70.00%	69.60%
10	8	50.40%	50.40%
15	8	42.13%	41.86%
20	5	39.10%	39.40%
25	5	35.76%	34.48%
30	12	34.33%	34.20%
35	12	35.25%	35.89%
40	14	36.90%	36.60%

Edge Histogram			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	3	70.40%	69.20%
10	24	49.00%	49.20%
15	12	37.73%	38.00%
20	19	33.90%	34.00%
25	7	29.60%	29.84%
30	7	30.20%	31.07%
35	7	35.60%	34.86%
40	7	37.75%	37.45%

Table 4.3: Estimated k Values and Corresponding Performances for the Descriptors Haar Coefficients and Homogeneous Texture

Haar Coefficients			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	6	57.20%	55.60%
10	7	44.80%	44.20%
15	10	36.00%	35.47%
20	3	32.70%	32.60%
25	8	32.70%	32.48%
30	3	29.73%	29.33%
35	5	31.65%	31.89%
40	5	33.90%	32.80%

Homogeneous Texture			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	5	60.40%	58.80%
10	17	37.80%	36.80%
15	17	28.26%	28.00%
20	35	24.00%	24.20%
25	40	23.28%	23.44%
30	48	20.60%	20.80%
35	58	22.22%	22.51%
40	154	23.55%	23.35%

Table 4.4: Estimated k Values and Corresponding Performances for the Descriptors Dominant Color(s) and Region Shape

Dominant Color(s)			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	1	54.80%	53.60%
10	1	33.60%	33.40%
15	57	29.73%	27.86%
20	67	25.70%	24.50%
25	17	23.12%	22.48%
30	13	23.00%	22.87%
35	6	23.82%	23.20%
40	6	26.30%	26.15%

Region Shape			
# of Classes	Estimated k Value	C.Validation Perf.	Test Perf.
5	5	42.00%	41.20%
10	18	22.00%	21.60%
15	66	16.26%	16.53%
20	31	13.40%	13.00%
25	20	11.60%	11.28%
30	126	10.53%	10.07%
35	55	11.31%	10.91%
40	37	11.65%	11.35%

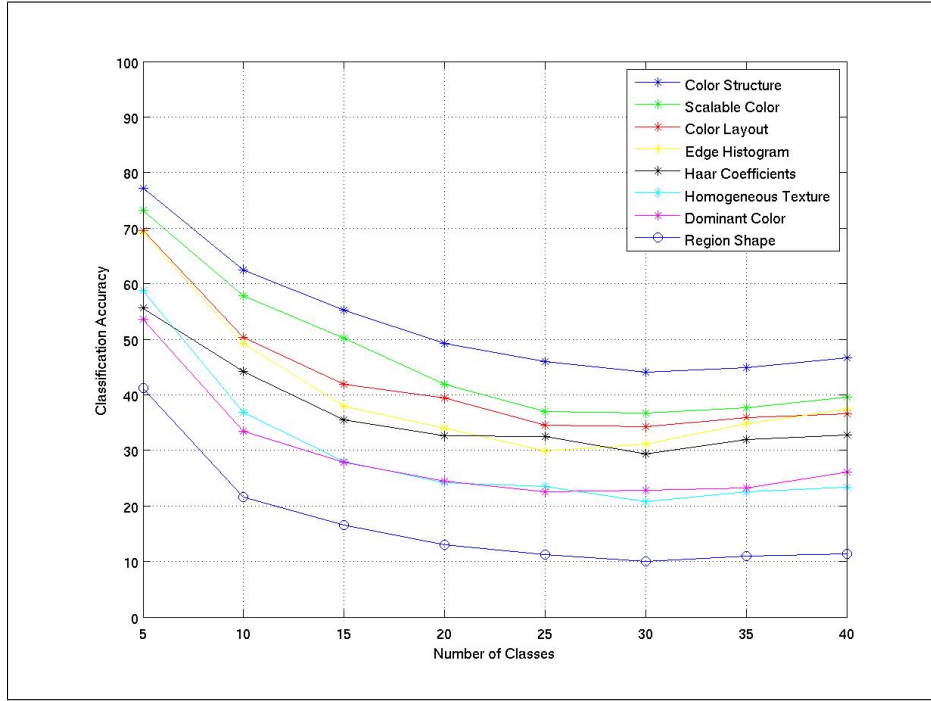


Figure 4.1: The comparison of the descriptors in terms of classification performance in the test set, in the presence of different number of classes.

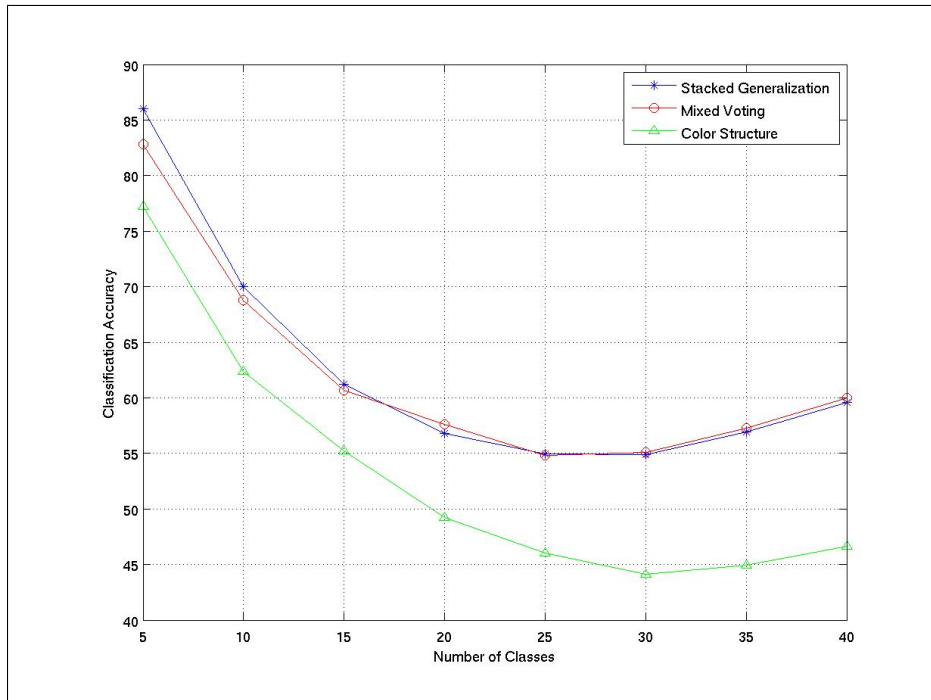


Figure 4.2: The comparison of unanimous decision to stacked generalization with four descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure.

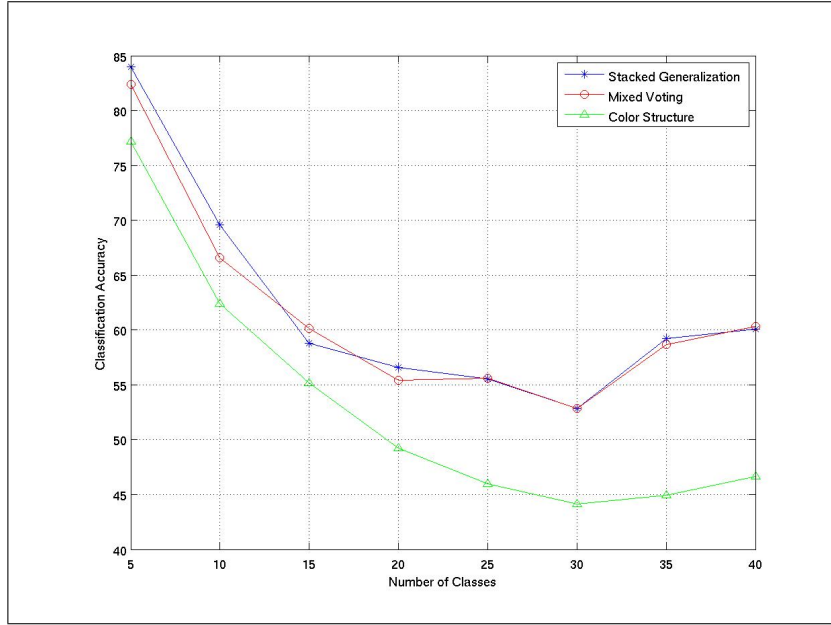


Figure 4.3: The comparison of unanimous decision to stacked generalization with five descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure.

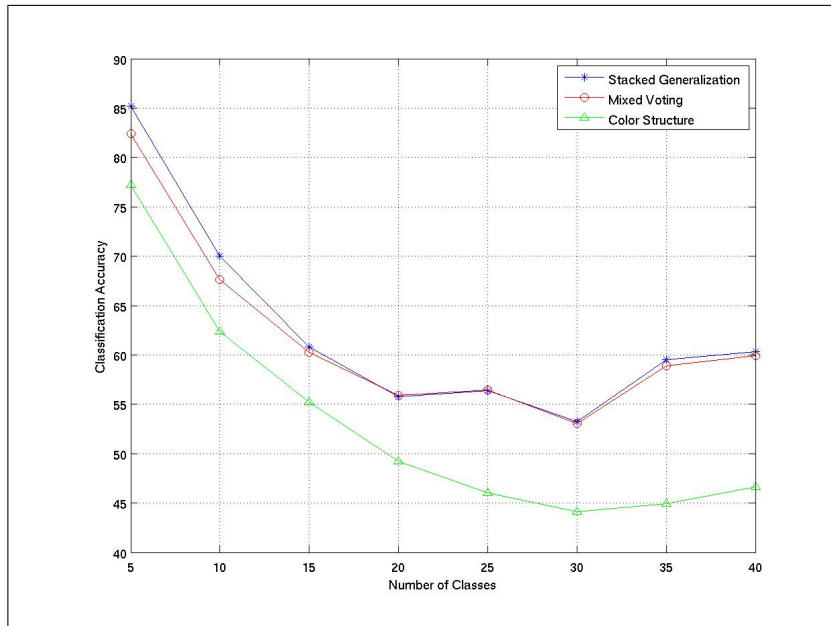


Figure 4.4: The comparison of unanimous decision to stacked generalization with six descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure.

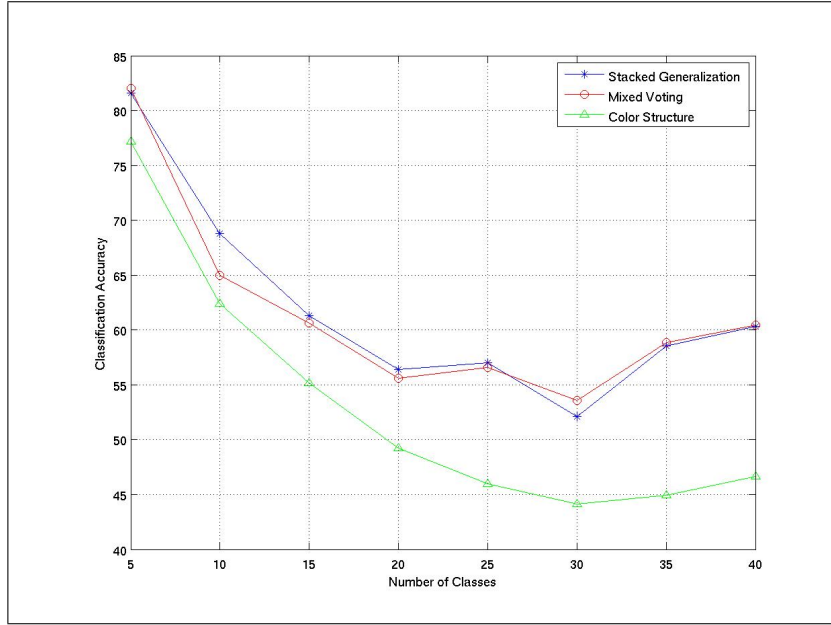


Figure 4.5: The comparison of unanimous decision to stacked generalization with seven descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure.

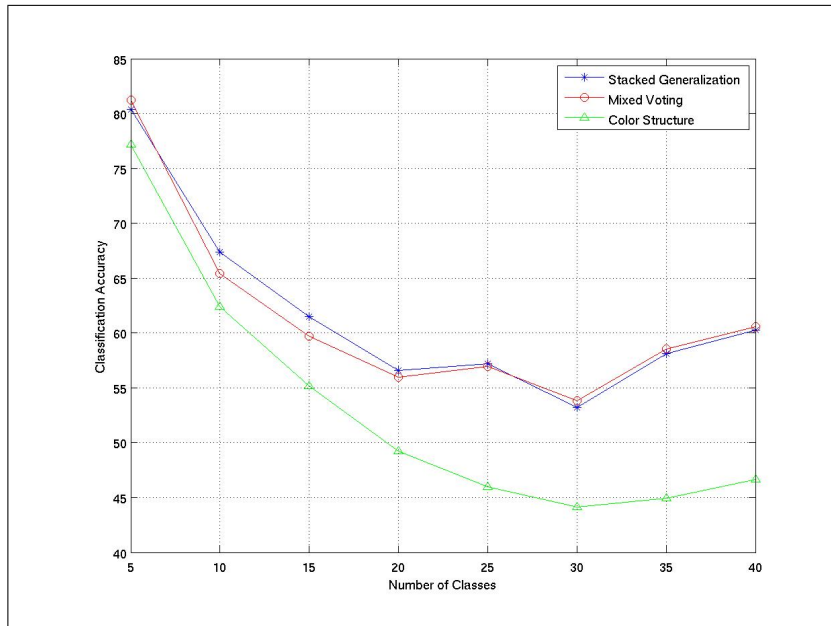


Figure 4.6: The comparison of unanimous decision to stacked generalization with eight descriptors in terms of test performance, in different number of classes. The classification accuracy of the Color Structure, the descriptor with the best individual performance, is also shown in the figure.

Table 4.5: Comparison of Unanimous Decision (UD) and Stacked Generalization (SG)

	Number of Descriptors Used					
	4		5		6	
# of Classes	UD	SG	UD	SG	UD	SG
5	82.80%	86.00%	82.40%	84.00%	82.40%	85.20%
10	68.80%	70.00%	66.60%	69.60%	67.60%	70.00%
15	60.67%	61.20%	60.13%	58.80%	60.26%	60.80%
20	57.60%	56.80%	55.40%	56.60%	55.90%	55.80%
25	54.80%	54.96%	55.60%	55.52%	56.48%	56.40%
30	55.06%	54.86%	52.86%	52.86%	53.06%	53.26%
35	57.25%	56.91%	58.68%	59.20%	58.91%	59.54%
40	60.00%	59.60%	60.35%	60.10%	59.95%	60.35%

	Number of Descriptors Used			
	7		8	
# of Classes	UD	SG	UD	SG
5	82.00%	81.60%	81.20%	80.40%
10	65.00%	68.80%	65.40%	67.40%
15	60.66%	61.33%	59.73%	61.46%
20	55.60%	56.40%	56.00%	56.60%
25	56.56%	57.04%	56.96%	57.20%
30	53.60%	52.10%	53.80%	53.20%
35	58.85%	58.57%	58.57%	58.11%
40	60.45%	60.35%	60.55%	60.25%

Table 4.6: The Class-Based Performances of Stacked Generalization Architecture in 40-Class Problem with 4 Descriptors

Class Number									
1	2	3	4	5	6	7	8	9	10
34%	60%	36%	62%	82%	60%	76%	22%	46%	38%

Class Number									
11	12	13	14	15	16	17	18	19	20
62%	58%	32%	16%	52%	52%	52%	48%	92%	26%

Class Number									
21	22	23	24	25	26	27	28	29	30
72%	52%	20%	72%	76%	92%	34%	48%	44%	66%

Class Number									
31	32	33	34	35	36	37	38	39	40
86%	84%	90%	94%	78%	72%	28%	84%	86%	100%

Table 4.7: The Class-Based Performances of Unanimous Decision Architecture in 40-Class Problem with 4 Descriptors

Class Number									
1	2	3	4	5	6	7	8	9	10
32%	54 %	34%	46%	90%	70%	68%	30%	44%	54%

Class Number									
11	12	13	14	15	16	17	18	19	20
70%	52%	36%	20%	44%	54%	54%	36%	92%	34%

Class Number									
21	22	23	24	25	26	27	28	29	30
64%	66%	18%	68%	74%	98%	38%	40%	40%	70%

Class Number									
31	32	33	34	35	36	37	38	39	40
86%	82%	92%	94%	82%	72%	26%	86%	90%	100%

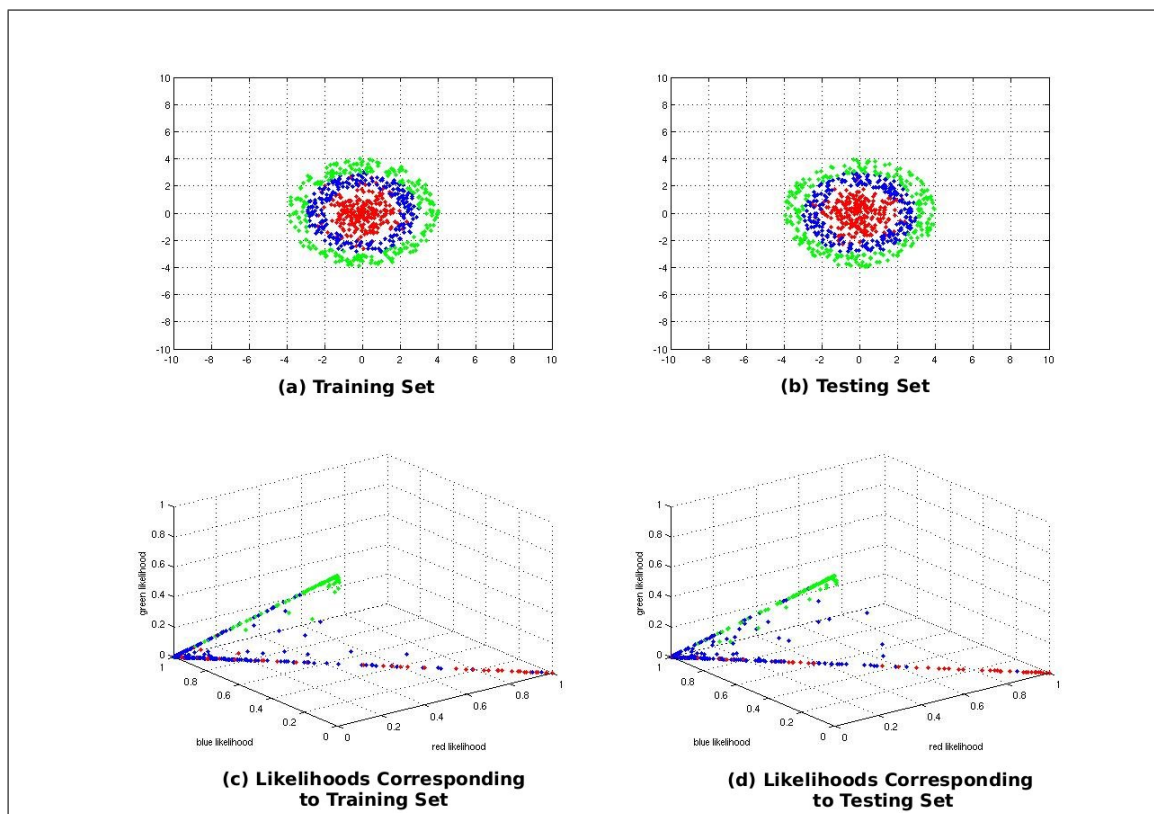


Figure 4.7: The training and the testing set in descriptor space-1; and the likelihoods from the corresponding base descriptor.

Table 4.8: The Average Performance of Individual Descriptors and the Average Performance of Their Triple Combinations with Stacked Generalization

Desc-1	Desc-2	Desc-3	Desc-4	SG(1,2,3)	SG(1,2,4)	SG(1,3,4)	SG(2,3,4)
85.093%	85.213%	84.84%	92.467%	99.387%	99.040%	99.133%	99.307%

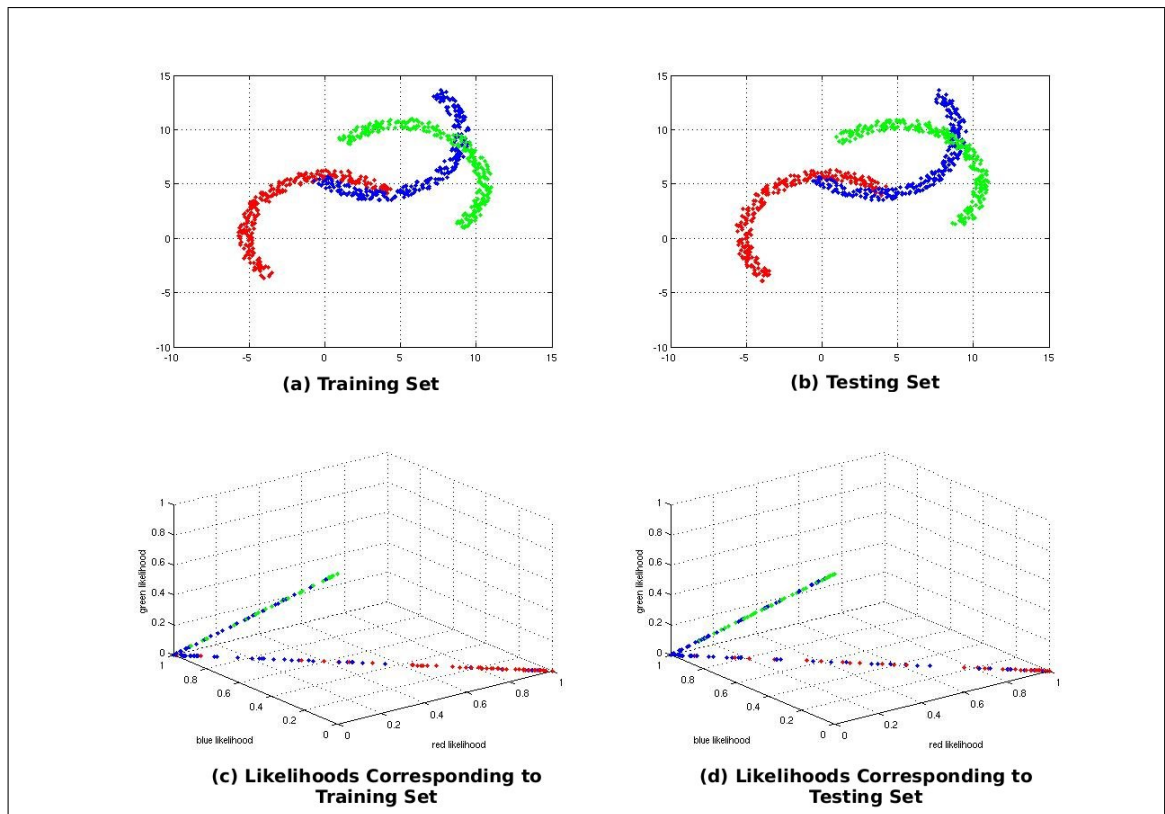


Figure 4.8: The training and the testing set in descriptor space-2; and the likelihoods from the corresponding base descriptor.

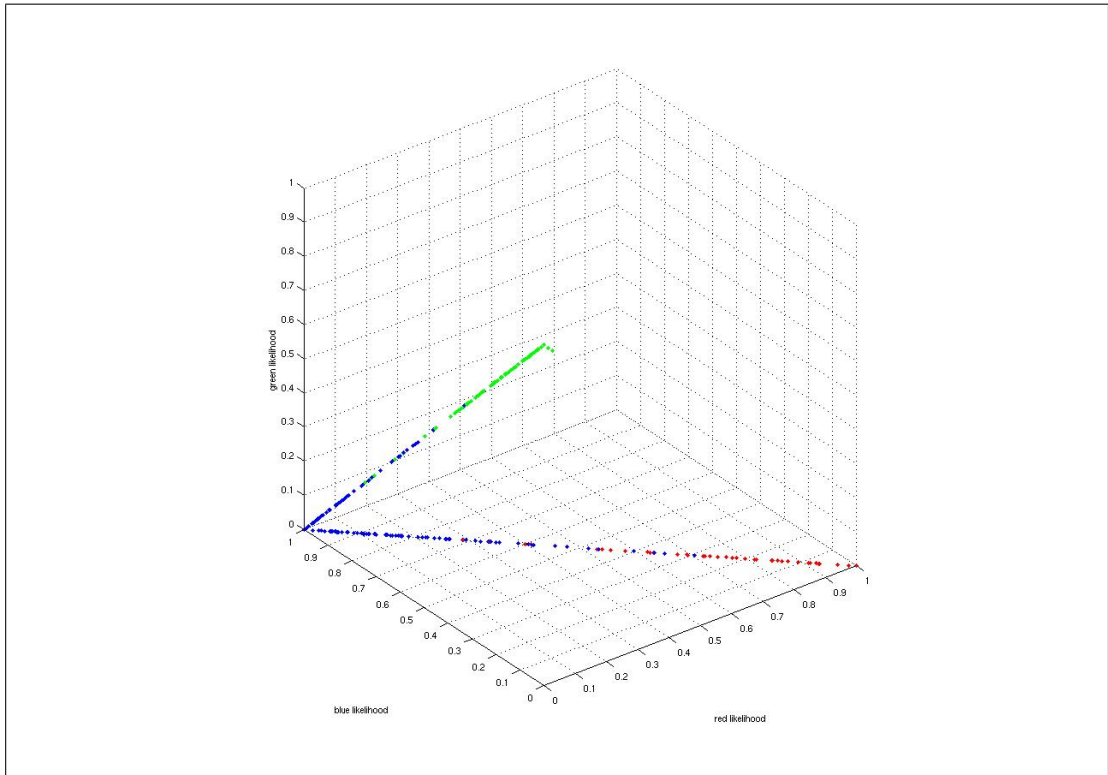


Figure 4.9: The likelihoods of the stacked generalization architecture corresponding to the testing set.

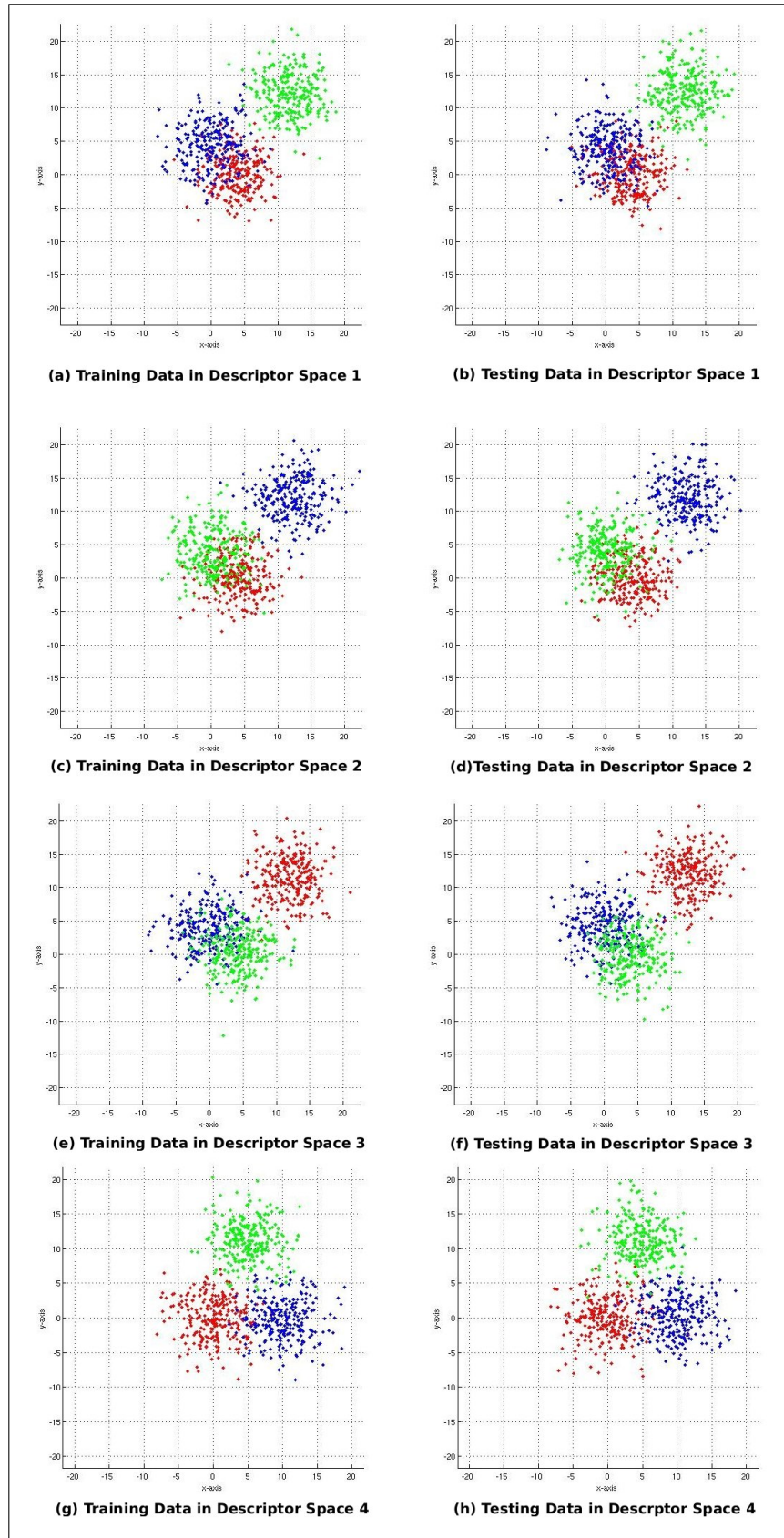


Figure 4.10: The representation of an instance of imaginary data in four different descriptor spaces.

CHAPTER 5

CONCLUSION

One of the problem of stacked generalization (SG) architecture is to be sensitive to the curse of dimensionality problem as the number of classes and the number of base classifiers increase. These increases grow out of concatenation of the likelihoods produced by classifiers in the first layer. A likelihood value is produced for each class by each classifier in the first layer and the concatenation of these likelihoods is presented to the second layer. The second layer of the architecture gets feature vector of the length the number of classes times the number of base classifiers. A solution proposed to the problem is a new meta-learning algorithm called “unanimous decision”. The solution addresses a restricted problem domain of SG which is homogeneous and with fuzzy k -nearest neighbors classifier.

SG architecture is known as a “black box”, the type of mapping that the architecture does has not been investigated in the literature. One part of this thesis is the investigation of the type of mapping of the architecture.

In the homogeneous SG, the selection of the descriptors effects the performance. The selection consists of two parts: the selection of the number of descriptors and the selection of these descriptors. Even the number of descriptors is fixed there is not a criterion to select these descriptors. To attack this problem of homogeneous SG architecture, a set of criteria is proposed and verified on a synthetically created dataset with fuzzy k -NN classifiers. These criteria are:

- Keeping the base classifiers with descriptors that are able to classify one or more categories well.
- Keeping the descriptors in such a way that for each category there is at least one descriptor that is able to classify it with a high performance.

By paying attention to these two factors it is shown experimentally that better results are

obtained than other combinations including the same number of base classifiers. This result reduces the effort that must be put to decide which combination of the available descriptors shall be selected for nearly optimal result.

Further research directions can be stated as follows.

- Experiments on the success criterion can be extended to more number of classes and more number of base descriptors with a good way to visualize the outputs produced by the first layer. This visualization technique can be Fisher linear discriminant analysis [18].
- In order to find a set of success criteria for all ensemble learning techniques, more combination schemes can be utilized in the experiments. Also, it is a necessity for the further research to put the results into a theoretical framework.
- The new architecture and SG shall be experimented with other dedicated individual classifiers such as artificial neural networks and Bayesian networks. In this way, the new architecture can be compared with SG by employing a plenty of other base classifiers.
- The relationship between SG and graphical methods, i.e. Bayesian networks, shall be investigated for further research and improvement. Since both models somehow works on belief propagation, it may result in remarkable facts enlightening the similarities and the differences between them. In addition to this, a method unifying these perspectives may result in high performance by combining discriminative and generative models.

REFERENCES

- [1] E. Akbas and F.T. Yarman-Vural. Automatic image annotation by ensemble of visual descriptors. In *SLAM07*, pages 1–8, 2007.
- [2] Emre Akbas. Automatic image annotation by ensemble of visual descriptors. Master’s thesis, Middle East Technical University, Ankara, Turkey, 2006.
- [3] E. Alpaydin and C. Kaynak. Cascading classifiers. *Kybernetika*, 4:369–374, 1998.
- [4] N. Arica and F.T. Yarman-Vural. An overview of character recognition focused on off-linehandwriting. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 31-2:216–233, May 2001.
- [5] Bellman. *Adaptive Control Processes*. Princeton University Press, 1961.
- [6] Irving Bierderman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.
- [7] Eran Borenstein, Eitan Sharon, and Shimon Ullman. Combining top-down and bottom-up segmentation. In *CVPRW ’04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’04) Volume 4*, page 46, Washington, DC, USA, 2004. IEEE Computer Society.
- [8] L. Breiman. Stacked regression. *Maching Learning*, 24:49–64, 1996.
- [9] Edward M. Callaway. Feedforward, feedback and inhibitory connections in primate visual cortex. *Neural Netw.*, 17(5-6):625–632, 2004.
- [10] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

- [11] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–71, 1991.
- [12] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.
- [13] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000.
- [14] M. Dash and H. Liu. Feature selection for classification, 1997.
- [15] Gustavo Deco and Tai Sing Lee. A unified model of spatial and object attention based on inter-cortical biased competition. *Neurocomputing*, 44-46:775–781, 2002.
- [16] Thomas D. Department. Scalable inference in hierarchical generative models.
- [17] Thomas G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.
- [18] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
- [19] M. Egmont-Peterson, W. R. M. Dassen, and J. H. C. Reiber. Sequential selection of discrete features for neural networks a bayesian approach to building a cascade. *Pattern Recognition Letters*, 20(11-13):1439–1448, 1999.
- [20] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *cvprw*, 12:178, 2004.
- [21] Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London A*, 1922.
- [22] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.
- [23] J. Gama and P. Brazdil. Cascade generalization. *Machine Learning*, 41(3):315–343, 2000.

- [24] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1458–1465, Washington, DC, USA, 2005. IEEE Computer Society.
- [25] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- [26] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(1):1157–1182, 2003.
- [27] Stephan Herrmann. MPEG-7 eXperimentation Model (XM). http://www.lis.ei.tum.de/research/bv/topics/mmdb/e_mpeg7.html.
- [28] International Organization for Standardisation: Coding of Moving Pictures and Audio. Multimedia content description interface, part 3 visual. Technical Report ISO/IEC JTC1/SC29/WG11/N4062, 2001.
- [29] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
- [30] J.M. Keller, M.R. Gray, and J.A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transaction on Systems, Man and Cybernetics*, 15(4):580, 1985.
- [31] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, July 2004.
- [32] Louisa Lam. Classifier combinations: Implementations and theoretical issues. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 77–86, London, UK, 2000. Springer-Verlag.
- [33] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.
- [34] T. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex, 2002.

- [35] Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *ECCV (4)*, volume 3954 of *Lecture Notes in Computer Science*, pages 581–594. Springer, 2006.
- [36] J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach, 2003.
- [37] MPEG (Moving Picture Experts Group). MPEG-7 overview. <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>.
- [38] C. Mulcahy. Image compression using the haar wavelet transform. *Spelman College Science and Mathematics*, 1:22–31, April 1997.
- [39] D. Mumford. Pattern theory: the mathematics of perception, 2002.
- [40] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.
- [41] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 2003.
- [42] Petr Savický and Johannes Fürnkranz. Combining pairwise classifiers with stacking. In *IDA*, pages 219–229, 2003.
- [43] B. Sch, o Burges, and A. Smola. Advances in kernel methods Support Vector Learning", text = "B. Scolkopf, C.J.C. Burges, and A.J. Smola, editors. Advances in Kernel Methods support vector learning. mit press, 1999., 1999.
- [44] Robert J. Schalkoff. *Pattern recognition: statistical, structural and neural approaches*. John Wiley & Sons, Inc., New York, NY, USA, 1991.
- [45] Henry Schneiderman and Takeo Kanade. Object detection using the statistics of parts. *Int. J. Comput. Vision*, 56(3):151–177, 2004.
- [46] Claude E. Shannon and Warren Weaver. *A Mathematical Theory of Communication*. University of Illinois Press, Champaign, IL, USA, 1963.
- [47] Kai Ming Ting and Ian H. Witten. Stacked generalizations: When does it work? In *IJCAI (2)*, pages 866–873, 1997.

- [48] Mutlu Uysal. *A Hierarchical Object Localization and Image Retrieval Framework*. PhD thesis, Middle East Technical University, Ankara, Turkey, 2006.
- [49] Mutlu Uysal, Emre Akbas, and Fatos T. Yarman-Vural. A hierarchical classification system based on adaptive resonance theory. In *ICIP*, pages 2913–2916, 2006.
- [50] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, 2001.
- [51] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance, 2003.
- [52] D. Wolpert. The relationship between pac, the statistical physics framework, the bayesian framework and the vc framework.
- [53] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.
- [54] Kevin Woods, W. Philip Kegelmeyer Jr., and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.
- [55] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2126–2136, Washington, DC, USA, 2006. IEEE Computer Society.

Appendix A

SOME SAMPLES FROM THE DATASET



(1) Dogs

(2) Lizards



(3) Models



(4) Mushroom



(5) Pyramids



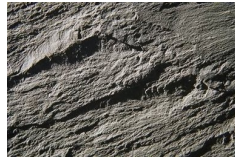
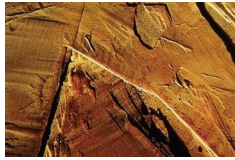
(6) Rome



(7) Sailing



(8) Sub sea



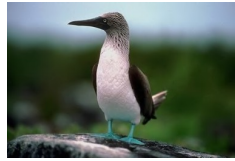
(9) Natural Texture

(10) Wild Cats



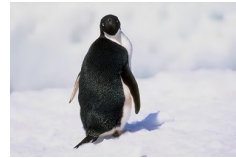
(11) Eagles

(12) Nest



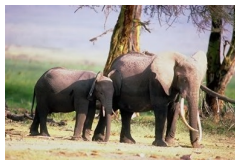
(13) Turtles & Birds

(14) Deers



(15) Lions

(16) Penguins



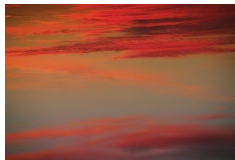
(17) Elephants

(18) Hippopotamus



(19) Horses

(20) Squirrel



(21) Sky & Cloud

(22) Artificial Textures



(23) Cartoon

(24) Stalactite



(25) Ski

(26) Antiques



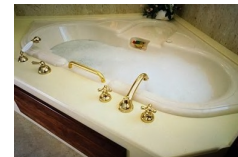
(27) Aviation

(28) Balloons



(29) Beaches

(30) Bobsled



(31) Bonsai

(32) Bathroom & Kitchen



(33) Cars

(34) Cards



(35) Cats

(36) Drinks



(37) Desert

(38) Firearms



(39) Fireworks

(40) Fitness