

Incorporation of Relational Information in Feature Representation for Online
Handwriting Recognition of Arabic Characters

Sara Izadi Nia

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

February 2010

© Sara Izadi Nia, 2010



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-67374-4
Our file *Notre référence*
ISBN: 978-0-494-67374-4

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Sara Izadi Nia

Entitled: Incorporation of Relational Information in Feature
Representation for Online Handwriting Recognition of Arabic Characters

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

complies with the regulations of the University and meets the accepted standards
with respect to originality and quality.

Signed by the final examining committee:

Dr. Chair

Dr. Peter Grogono Examiner

Dr. William E. Lynch Examiner

Dr. Ahmed K. Elhakeem Examiner

Dr. Ching Y. Suen Supervisor

Dr. Nawwaf Kharma Supervisor

Approved By _____
Chair of Department or Graduate Program Director

Dr. Robin A.L. Drew , Dean
Engineering and Computer Science

Date _____

ABSTRACT

Incorporation of Relational Information in Feature Representation for Online Handwriting Recognition of Arabic Characters

Sara Izadi Nia

Interest in online handwriting recognition is increasing due to market demand for both improved performance and for extended supporting scripts for digital devices. Robust handwriting recognition of complex patterns of arbitrary scale, orientation and location is elusive to date because reaching a target recognition rate is not trivial for most of the applications in this field. Cursive scripts such as Arabic and Persian with complex character shapes make the recognition task even more difficult. Challenges in the discrimination capability of handwriting recognition systems depend heavily on the effectiveness of the features used to represent the data, the types of classifiers deployed and inclusive databases used for learning and recognition which cover variations in writing styles that introduce natural deformations in character shapes. This thesis aims to improve the efficiency of online recognition systems for Persian and Arabic characters by presenting new formal feature representations, algorithms, and a comprehensive database for online Arabic characters. The thesis contains the development of the first public collection of online handwritten data for the Arabic complete-shape character set. New ideas for incorporating relational information in a feature representation for this type of data are presented. The proposed techniques are computationally efficient and provide compact, yet representative, feature vectors. For the first time, a hybrid classifier is used for recognition of online Arabic complete-shape characters based on the idea of decomposing the input data into variables representing factors of the complete-shape characters and the combined use of the Bayesian network inference and support vector machines. We advocate the usefulness and practicality of the features and recognition methods with respect to the recognition of conventional metrics, such as accuracy and timeliness, as well as unconventional metrics. In particular, we evaluate a feature

representation for different character class instances by its level of separation in the feature space. Our evaluation results for the available databases and for our own database of the characters' main shapes confirm a higher efficiency than previously reported techniques with respect to all metrics analyzed. For the complete-shape characters, our techniques resulted in a unique recognition efficiency comparable with the state-of-the-art results for main shape characters.

Table of Contents

List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Thesis Objective	3
1.2 Research Contributions	5
1.3 Statement of Originality	7
1.4 Thesis Outline	7
2 Background	10
2.1 Offline and Online Handwriting Recognition Systems	11
2.2 A Typical Online Handwriting Recognition System	12
2.3 Challenges in Handwriting Recognition	13
2.4 Direct Approaches	15
2.5 Indirect Approaches: Classification in Online Handwriting Systems .	16
2.5.1 Structural Methods	17
2.5.2 Rule-based Methods	18
2.5.3 Prototype-based Methods	19
2.5.4 Statistical Methods	22
2.5.5 Hybrid Methods	23
2.6 Indirect Approaches: Online Data Representation Methods	25
2.6.1 Structural Representations	26

2.6.2	Representation in Dissimilarity Space	26
2.6.3	Representation in Feature Space	28
2.6.3.1	Global/High-level Feature Representation	28
2.6.3.2	Local/Low-level Feature Representation	30
2.6.4	Combining Online and Offline Features	35
2.7	Studies on Feature Selection for Online Handwriting Recognition . . .	36
2.8	Persian/Arabic Scripts	38
2.8.1	Characteristics of Persian/Arabic Handwriting	38
2.8.2	Classification and Data Representation for Online Persian/Arabic Handwriting	41
3	A New Online Arabic Database	46
3.1	Online Arabic/Persian databases	48
3.2	Design of the New Database of Complete Arabic Characters	50
3.3	Tools Developed for Data Collection and Visualization	52
3.3.1	Data Collection Software	52
3.3.2	Viewing Tools	53
3.4	Policies used for sample collection	55
3.5	Characteristics of the new database	56
3.5.1	Data Variation	57
3.5.1.1	Complementary Parts	59
3.5.2	Data Format and Verification	61
3.5.3	Sections of the Database	64
4	Relational Histogram Feature Representation in a Neural-Network- Based Recognition System	67
4.1	Data Preprocessing	69
4.2	Relational Histogram Representation	71
4.2.1	The Neural-Network-Based Recognition System	75
4.3	Empirical Evaluations	78
4.3.1	Experimental Setup	78

4.3.2	Results	79
4.4	Conclusion	81
5	Relational Context Representation and an SVM-Based Recognition System	82
5.1	Relational Context Representation	84
5.2	The SVM-Based Recognition System	87
5.3	Qualitative Analysis for the Proposed Features	88
5.3.1	Invariance and Uniqueness	89
5.3.2	Flexibility	90
5.3.3	Ease of Implementation	90
5.3.4	Memory Issue	91
5.4	Empirical Evaluation	91
5.4.1	Experimental Setup	92
5.4.2	Results	92
5.5	RC in Comparison with Existing Features for Online Handwriting Recognition	94
6	Separability Analysis and Evaluation of Feature Representations	99
6.1	Separability Measure	101
6.2	Distance Metrics	102
6.2.1	Test of Normality	103
6.2.2	Geometric-based measures	105
6.2.3	Information-Based Measures	107
6.3	Related Application Areas	108
6.3.1	Feature Selection	109
6.3.2	Clustering	111
6.4	Experimental Results	112
6.5	Conclusion	117

7 A Hybrid System for Recognition of the Complete Shape in Arabic Letters	120
7.1 Segmentation Method	124
7.2 A Hybrid System for Recognition of Complete Online Arabic Characters	128
7.3 Bayesian Networks	130
7.4 The BN Hybrid Classification	133
7.5 Experimental Results	136
7.5.1 Erroneous Cases	143
7.6 Conclusion	145
8 Conclusions and Future Work	147
8.1 Summary	148
8.2 Future Extensions	151
Bibliography	154

List of Tables

2.1	Online features used in the literature.	34
2.2	Summary of methods, performances, and comparability	44
3.1	Summary of database specification used in	49
3.2	The statistics of our database.	56
3.3	The statistics about the number of strokes in	59
4.1	Performance comparison of different features in	80
4.2	Confusion matrix for a sample run using the RH representation.	80
4.3	Performance of RH features using our database.	81
5.1	Comparative experiments on the database in [113] of Arabic letters.	93
5.2	Comparison of different features tested on an Arabic dataset	95
6.1	Separability measures for three features on a dataset	113
7.1	The recognition results for multi-stroke Arabic	136

List of Figures

2.1	Diagram of a typical online handwriting recognition system.	13
2.2	Directional features and turning angels for a discrete curve.	33
2.3	Four different shapes of letters in the Persian alphabet.	40
2.4	Different allographs for writing the same 3-letter word.	40
3.1	Isolated letters from the Arabic alphabet and	51
3.2	Photograph of the type of tablet and accessories used	51
3.3	Screen shot of our developed software application	53
3.4	A snapshot of the software application developed for	54
3.5	A real example of an Arabic character	57
3.6	Three samples of the Arabic letter Taa in	58
3.7	(a) Arabic letter Shiin in typewritten form	60
3.8	(a) Arabic letter Jiim in typewritten form	62
3.9	Similarly written letters: (a) Samples of the Arabic	63
3.10	Different letters which are written similarly	63
3.11	Each row shows a sample of multi-stroke Arabic	65
4.1	The Arabic letter dal in its (a) original	71
4.2	Diagram of the log-polar bins around the center	74
5.1	The Arabic letter “ye” in a relative context feature representation.	86
5.2	Some misclassified Arabic letters from our database	97
5.3	Nine instances of the Arabic letter waaw misclassified as	97
6.1	Comparison of the separability measure for classes $C1$ to $C17$	114

6.2	The scatter plot for some samples of class <i>C1</i> in the <i>RC</i> feature space. . .	115
6.3	The scatter plot for some samples of class <i>C2</i> in the <i>RC</i> feature space. . .	116
6.4	QQ-plots of some of the 17 classes of Arabic character	118
7.1	Different forms of secondary components: (a) one dot	123
7.2	Segmentation of the Arabic letter <i>Zay</i>	125
7.3	The block diagram of our BN hybrid classification approach	129
7.4	Bayesian network used in our hybrid system for	134
7.5	The probabilities for a sample of the Arabic letter <i>taa</i>	138
7.6	The probabilities for a sample of the Arabic letter <i>nuun</i>	139
7.7	The probabilities for a sample of the Arabic letter <i>nuun</i>	140
7.8	The probabilities for a sample of the Arabic letter <i>taa</i>	141
7.9	The probabilities for a sample of the Arabic letter <i>waaw</i>	142
7.10	Some misclassified samples for which the source of error is	144
7.11	Some misclassified samples for which the source of error is	144
7.12	Some misclassified samples for which the source of error is	145
1	QQ-plots of all 17 classes of Arabic character shapes	176
2	QQ-plots of all 17 classes of Arabic character shapes	177
3	QQ-plots of all 17 classes of Arabic character shapes	178
4	QQ-plots of all 17 classes of Arabic character shapes	179

Chapter 1

Introduction

Writing by hand is the most natural way for humans to communicate and exchange information. The desire to combine the convenience of handwriting with the need to use, maintain and communicate digital information has led the industry to integrate handwriting input into devices such as hand-held computers, smart phones, personal digital assistants (PDAs), tablet PCs, and smart boards. These advances make computers more useful to people in many circumstances. While the hardware is simple to implement using touch-sensitive pads, there is a pressing need for software to effectively recognize handwritten data. The main component of the required software is an online handwriting recognition unit. In online handwriting systems, the recognition and handwriting happen at the same time, as opposed to offline handwriting recognition, in which handwriting occurs prior to recognition.

Handwriting recognition is a difficult task in general due to the huge variety of writing styles among different individuals, or *variation*; and different samples at different times under different conditions and moods in each individual, or *variability*. In addition, allographs (different ways of writing the same letter) increase the effective

size of an alphabet. In online systems, variations in the number and the order of strokes introduce more variations in handwriting, while not all of these subject-specific parameters carry information about the character's identity. Recognition of scripts such as Arabic, with its complex characters' shapes and many allographs, is even more difficult than the recognition of Latin scripts.

Research interest in online handwriting recognition has increased due to the market demand for both improved performance and for extended supporting scripts for digital devices. Many handwriting recognition algorithms and techniques have been introduced in the last few decades, however, online handwriting recognition has still remained an active area of research to this date [162, 18, 8, 47, 69, 77, 76]. This is due to the three following main challenges: low recognition rate; high recognition time; and lack of comprehensive databases which represent the variation and the variability exhibited in handwritten data.

For some scripts such as Persian and Arabic, the lack of sufficient data sets (which are a representative collection of natural handwriting) is a serious obstacle for developments and evaluation of efficient handwriting recognition methods. Currently, the recognition rate on average is 80%-85% for words and 90%-93% for symbols, on highly restrictive classes of patterns which do not cover the variations and variability that exist in real handwriting data. From a practical perspective, research efforts in the field to tackle the three above-mentioned challenges in online handwriting recognition should address the following: the design of feature representations that describe handwritten data more intelligently and more efficiently for recognition algorithms; faster and more accurate algorithms to classify handwritten data; and databases of

handwritten data in different scripts, taken from an unbiased collection of people with multiple entries from each person. The database collection and data extraction are not considered to be part of the recognition phase. However, the problem of the lack of real and naturally representative data makes the recognition evaluation fragmentary with restricted data samples.

1.1 Thesis Objective

In this thesis, we address the challenges for improving an online recognition system for isolated Arabic characters. The development of a comprehensive database of online handwritten samples with arbitrary variations is the focus of the first part of this thesis. This first part tries to address the aforementioned issue of the lack of complete databases of the Arabic characters. Therefore, the development of our database will contain the first public collection of online handwriting for the complete Arabic character set and this collection can be used independently for future research on this script.

In the second part of this thesis, our focus is on the recognition of online isolated characters in the context of the Arabic/Persian languages. We investigate new approaches in online recognition by designing new feature descriptors for online data and by proposing a hybrid system for the recognition of the characters in their main shapes. Our objective in the first part of this investigation is to gain new, more representative, and more sophisticated features for handwriting recognition in order to tackle the huge inter-class variability problem. This includes improving the accuracy

of recognition and improving the speed of an online recognition system for online Arabic characters. The first proposed feature representation is called the *relational histogram feature* representation. This representation augments the visual realism of the shape of a character by incorporating physical and geometric characteristics of the data. We use this feature together with a tangent feature for character classification. In a previously published work [82], we illustrated that the ability of this feature is promising when used with an artificial neural network recognition engine. The second proposed feature representation is called the *relational context feature* representation. In this representation, relative interrelationships of temporal data points of a character are captured in an arbitrary length. Experimental evaluations in our previously published study showed that our approach outperforms the best reported recognition rates for Arabic characters, in a writer-independent system by using support vector machines as the classification engine [84]. The recognition time when using this approach is also significantly less than that of the state-of-the-art approach for the same database [114]. Experimental results for the newly developed database and for a previously existing database show that these feature enhancements represent a notable improvement in the recognition accuracy and in the recognition time. Our objective in the second part of this investigation is to design a method for the recognition of Arabic letters in their complete shapes by using the advantages of the proposed representation method. For this purpose, we used the combinations of three such systems used for the recognition of the segments of an online character. The combination had to be performed in a probabilistic way for reducing the generalization error. In a new design, we integrated a Bayesian classifier for combining the

results of SVM classifiers. This classification approach was successful when tested for our database of Arabic characters in their complete shapes. Despite the increase in the number of classes by ten, the recognition rate had only a minor decrease.

1.2 Research Contributions

There are four major research contributions in this thesis. The first is the design and demonstration of a complete online database of Arabic characters collected from people with different demographics. Our database represents a comprehensive collection of information about the writers as well as the data collected from them. The design of handwritten databases is a complex endeavor and requires input from many different perspectives including the technology used, user behavior, and human factors. Our database is unique in that it provides information about the users, and may therefore offer insights into task-specific or user-specific recognition system requirements and constraints.

The second contribution of this thesis is the design of new feature representations for online Arabic characters. In particular, we introduce relational representation, and histogram representation. To have a powerful character representation which tolerates both variations and deformations of patterns, our system needs to learn and maintain both the local and global information related to the characters. Most representations are combinations of different local and global measures. On the other hand, there has been relatively little work on features interacting with the direct shape of a character. The idea behind our new feature representations is to consider measurements of local

and global characteristics which can capture information related to the character's shape.

The third contribution relates to our findings regarding class separability analysis of a data set, by using only the feature representation to describe the data before classification. These findings provide insights not only for the design of feature representation methods and classification, but also for the design of new distance metrics which can be applicable and potentially useful in the separability of online handwritten data. In particular, separability is a basic behavioral phenomenon of the data which recognition systems need to consider and address. Our study reveals different levels of separability analysis in the same data samples and this observation leads to an interesting conjecture that merits further investigation.

The final contribution is concerned with the important and almost unexplored issue of Arabic complete-shape character recognition. Our approach includes the design of a segmentation method for extracting different parts of the character shape, and the design of a new sophisticated combination of several classifiers in a hybrid system. While the proposed classification method, benefits from the proposed representation, the method possess a novelty of using a Bayesian network for reasoning about a second-level classification from the classification results of different components in building the main-shape characters. This contribution reveals the fact that considering the main-shape characters can provide more benefits for classification of some characters, while it makes the recognition more difficult in others. Our new methodology and unique results are significant advances to online Arabic character recognition. These results can be used as the first comprehensive references for future

research in this field.

1.3 Statement of Originality

This thesis incorporates the outcome of research performed independently for my Ph.D. A portion of the background material on Persian script recognition and an in-depth study of the literature were previously published as a survey in [161] and [80]. The material presented for relational and histogram feature representation was published in peer-reviewed conference proceedings [82, 83, 84] and was presented at workshops [81]. This thesis also contains a significantly more extensive discussion of the proposed techniques and problems investigated compared to the content of the published papers. We also provide additional experimental results and different evaluations with the new database as well as with existing databases. Furthermore, we present techniques for the recognition of Arabic complete-shape characters.

1.4 Thesis Outline

The rest of this thesis is organized as follows:

Chapter 2 presents a brief literature review of online handwriting recognition systems. In this chapter, we also highlight the classification methods and feature representations generally used for online handwriting recognition. Since the focus of this thesis is on Arabic/Persian scripts, we then continue this chapter with a background discussion on Arabic scripts and we explain the challenges involved in the recognition of Arabic/Persian handwriting. This discussion is followed by a review of the current

state of research on these scripts.

Chapter 3 provides the specification of the database developed in this thesis and consisting of a complete online character set in the Arabic alphabet. In this chapter, we introduce the tools designed for data collection and the visualization of online data patterns. We also explain the policies developed for data collection in order to increase the variation and variability in this new database.

Chapter 4 introduces the first feature representation proposed in this thesis for online character data. In this chapter, we also describe a feature extraction technique and we describe our proposed character recognition system in detail, utilizing artificial neural networks. The recognition experiments were conducted on two different data sets: an online Arabic isolated characters database in [110] and the new database developed in Chapter 3.

Chapter 5 describes the second approach proposed for feature representation. The complete character recognition system using this representation and support vector machines for classification of patterns is presented in this chapter. Comprehensive experimental results on the evaluation of this method in comparison with a wide range of other approaches for feature representation are demonstrated.

Chapter 6 considers the general problem of class separability in the context of online Arabic characters by using different feature representations. This chapter tries to evaluate a feature representation method independent of the choice of classifiers. Separability analysis of our Arabic character database and the evaluation results using this analysis are provided.

Chapter 7 describes an innovative hybrid classification method for the recognition

of multi-stroke Arabic letters. The components of the hybrid classifier and the recognition methodology are presented in detail. This method requires segmenting the characters into appropriate building block strokes. This chapter introduces a segmentation algorithm which results in three stroke types, used later on in this chapter by three classifiers. The recognition performance in a 28-class character set is evaluated by using the database that was developed in Chapter 3. Experimental results of this evaluation are also provided in this chapter.

Chapter 8 concludes with a discussion about key findings of the research, potential impacts and possible limitations. This discussion is followed by suggestions for further improvements, such as possible future extensions of this work, and also immediate directions for future research.

Chapter 2

Background

Chapter Outline

In this chapter, we provide background information about online handwriting recognition, and the challenges involved in this area of research. Then, we present a literature review on approaches used in the design and development of systems to tackle those challenges. Since Arabic and similar scripts such as Persian and Urdu are the main focus of this thesis, we briefly review some of the common characteristics of these scripts. At the end, we present a survey of different approaches that have been used in the literature for Arabic/Persian online handwriting recognition.

Handwriting machine recognition has been an ongoing research topic since the early sixties [49]. Handwriting recognition research was initially limited to machine reading of paper documents, which is now called *offline handwriting recognition*. With the advent of digital tablets as a new medium of writing, handwriting recognition has branched into a new category called *online handwriting recognition*. Major differences between online and offline handwriting systems lie in the areas of data acquisition,

data representation, and applications. In Section 2.1 we discuss these differences in more detail.

2.1 Offline and Online Handwriting Recognition Systems

Offline handwriting recognition systems take as input either handwritten or machine printed texts that are recorded on a piece of paper. The recognition process starts from scanning a paper document, which produces a digitized image of the text. The image is then delivered to the offline system for some further processing and recognition. Patterned, crumpled, or old paper documents demand more preprocessing operations before recognition. In a trained offline system, capturing the handwriting and the processing of captured data take place at two different points in time. So far, offline handwriting recognition has been used in a wide range of applications such as bank cheque processing, mail sorting, fax filing, and machine reading aids for visually impaired people.

The online data acquisition environment consists of a pointing device (e.g. a corded or cordless pen) and a sensing device (e.g. the surface of a digitizer)¹. The sensing device can accurately locate the center position of the pointing device. The trace of the pen tip is recorded as a sequence of points, sampled in equally-spaced time intervals $(x(t), y(t))$. The sequence of the spatial positions and the pen-up/pen-down occurrences are usually the digital data that are recorded by a software application.

¹The electronic tablet, invented in late 1950's, is a typical capturing device for online data. Often, an orthogonal grid of wires is used to locate the pen-tip in the digitizer.

Any sequence of points from a pen-down state to the next pen-up state is called a *stroke*. The accuracy of the digital pattern depends on the following: the skill of the writer, the nature of the pointing device, and the resolution of the digitizer's grid².

The main difference between online and offline data is the availability of dynamic (temporal) information in online data. In online handwritten recognition, the process of automatic recognition takes place as a person writes on the digitizer, and the user can interact with the recognizer during the writing process.

The renewed research interest in online handwriting recognition has been escalating mainly because of the mass production of devices such as tablet PCs, hand-held computers, personal digital assistants (PDAs), smart phones, and white-boards and the demand for a non-keyboard-based data entry.

2.2 A Typical Online Handwriting Recognition System

A typical online handwriting recognition system consists of three main building blocks: preprocessing, pattern representation, and classification. Preprocessing operations are applied in order to reduce the noise introduced by the digitizing device in online data. Pattern representation includes design and extraction of representative features for classification purposes. The final decision on the class of the input pattern is made using features (and system models) by a recognition module. A diagram of a typical online system is shown in Figure 2.1.

²The typical resolution rate and sampling rate of the electronic tablets are 200 dots per inch and 130 points per second, respectively. There is a good review of digitizing technology in [164].

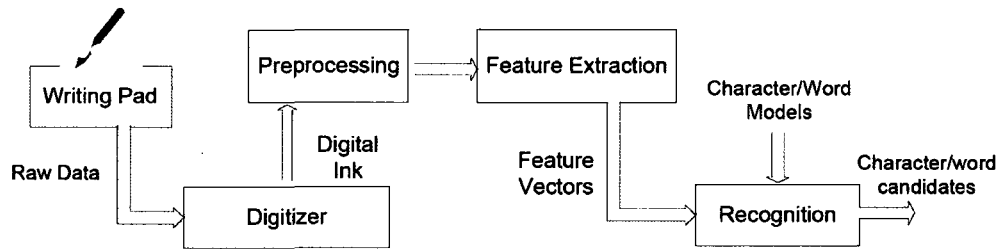


Figure 2.1: Diagram of a typical online handwriting recognition system.

2.3 Challenges in Handwriting Recognition

The difficulty of automated handwriting recognition is rooted in the *variation* and *variability* in handwriting as pointed out by Schomaker [151]. Variation refers to the idea that each individual has a unique way of writing. Variability addresses the changes of a specific individual in producing different handwritten samples over time. The main challenge in the design of efficient features is the search for invariance. The error rate for recognizing unconstrained handwriting in a writer-independent online system is still too far from being practical for real applications due to the huge variation in writing styles. Schomaker categorized sources of variety in handwriting into four different groups:

- Affine transforms;
- Sequence variety;
- Neuro-biomechanical variety; and
- Allographic variety.

Affine transformation, also known as geometric transformation, refers to the size and slant variations between writing styles. Neuro-biomechanical refers to the variations due to the speed or concentration of a writer. Sequence variety addresses the variation in number and order of the arcs in graphemes. An allograph is defined as a letter of an alphabet in a particular shape. Allographic variety addresses the number of shapes used by writers for a given grapheme. This source of variation has been pointed out as the toughest problem in handwriting [151]. The allographs that one uses depends on some factors such as the region and generation that the writer belongs to, and the early education received by an individual for writing. According to Schomaker, even five million characters in the UNIPEN dataset [62] do not seem to be enough data for expressing all allographical variations.

Overcoming the challenges mentioned can be categorized into two main groups of direct and indirect approaches. Direct approaches which try to simplify the problem at the data level are explained in Section 2.4. The indirect approach to the variation challenge attempts to increase the system's discriminative power by improving the system modeling. The two system modules which have the most impact on the overall system performance are *pattern classification* and *pattern representation*. Those approaches that concentrate on the pattern classification part require making a right choice of classifier, or combining different classifiers. They aim at overcoming the variation by forming more generalized classification boundaries for the data at hand. Whereas, those approaches that concentrate on the pattern representation part address the question of how to describe the pattern better, so that the complexity of the classification is reduced. In the following sections, we review direct and indirect

approaches in the literature for designing a more accurate handwriting recognition system. Indirect approaches are explained in Sections 2.5 and 2.6.

2.4 Direct Approaches

A direct approach to the variation challenge consists of breaking down the variation by some data manoeuvring such as categorizing similar ways of writing. *Allograph modeling* clusters the writing styles, and models them separately. *Writer adaptation* narrows down the variation by reducing the generality of the system from being user-independent (free-style) writing to be customized to a specific user/writing style. Direct approaches to the variation problem consist of allograph modeling and writer adaptation methods. Allograph modeling methods handle varieties of writing styles by dividing one class of patterns into more classes depending on how different they may be. If the writing styles are too different, they are considered as separate classes. In this approach, usually different writing styles for the same symbol are grouped by clustering the training instances. For this method to work well, a lot of training data that covers different writing styles is needed. Writer adaptation is the process of converting a writer-independent handwriting recognition system, which models the characteristics of a large group of writers, into a writer-dependent system, which is tuned for a particular writer [38]. A system becomes specialized for a specific person through adjusting parameters for achieving a better classification. There are two main approaches for writer adaptation. The first approach is to build a system which

includes a classifier for each category of similar writing styles (identified by the clustering of free-style training data). This system is tested using the user's handwriting in order to associate the user to a category. The classifier which gives the most confident recognition for the user is chosen to be further tuned. The second approach is to train a new classifier for the user while incorporating a priori over the classifier's parameters, learned from the generic dataset [163]. Adaptation strategies are presented in the literature for fuzzy systems [117], Hidden Markov Models (HMM)-based [25], HMM/Neural Network (NN)-based [163], and for prototype-based systems [120]. A comparison between some adaptation techniques was conducted in [26]. Although adapting to a writer provides a significant performance gain, the demanded time and effort from the user until the performance gain is achieved can be unappealing.

2.5 Indirect Approaches: Classification in Online Handwriting Systems

Classes can be defined as a number of mutually exclusive sets, which are identified by evaluating some similarly valued common characteristics among a set of patterns. Classification is the task of assigning an unknown pattern to one of the classes by evaluating the similarity of the pattern to members of all existing classes. Such mapping from the pattern space to a discrete set of class labels is performed by classifiers. In general, a flawless classification performance is hard to achieve. The main reasons for the classification error include: the samples used in designing the classifier are not representative, the features characterizing patterns are not adequate,

the method used for separating classes is not efficient, or there is an intrinsic overlap of the classes that no classifier can resolve [43]. Classification emerges in any handwriting recognition task. A variety of techniques ranging from structural and rule-based methods to statistical modeling have been attempted in the development of classifiers, each of which possesses some pros and cons. In the following sections, we briefly explain each group of classifiers while we refer to the related online handwriting research in the literature.

2.5.1 Structural Methods

Structural (syntactic) classifiers are designed for a specific pattern representation, which is composed of sub-patterns (primitives) plus the relationship between them. Patterns with different levels of complexity can be built from primitives in the form of discrete mathematical models of formal language theory, by what is called a *grammar*. A grammar is usually defined as a four-tuple $G = \{V_T, V_N, P, S\}$. The entities are a set of terminal symbols (primitives) denoted by V_T , a set of nonterminal symbols (variables) which are disjoint from V_T , denoted by V_N , a set of production rules denoted by P , and a string symbol (root) denoted by S , where $S \in V_N$ [56]. Classification in such approaches is just parsing³. In other words, the structure of the test pattern is compared to the existing class structures or grammar. The test pattern is classified if it is syntactically accepted by the grammar, and rejected otherwise [56]. Structural classifiers have been used for recognition of online Chinese handwriting [103], Indian digits [55], alphanumeric characters [33], and online mathematical expressions [34].

³Parsing is the process of structuring a representation of a natural language sentence usually with respect to a given grammar.

The interest in using structural methods in pattern recognition applications has decreased due to several shortcomings such as the difficulty in constructing the capable syntactic grammars for discriminating among classes [130]. The grammar construction is often performed manually by a domain expert, however, creating grammars becomes unmanageable for complex classification problems. Structural methods are computationally expensive because the matching of character structures is a nontrivial combinatorial problem [102]. More details about structural methods can be found in [29].

2.5.2 Rule-based Methods

The rule-based method is a heuristic approach to classification which defines each class by constructing a set of rules about the class features. Most of the training instances of each class should satisfy the corresponding rules. Fuzzy logic provides a good ground for an extension of rule-based classification by offering a multi-valued logic that formalizes imprecision and uncertainty in reasoning. In fuzzy rule-based classification, decision boundaries are assumed to be fuzzy. In other words, the overlapping class definitions are allowed. In a typical fuzzy classification system, where each instance is described by a feature vector X ($X = [X_1, X_2, \dots, X_n]^T$), a fuzzy if-then rule R_i , that describes the class C_i might be:

$$R_i : \text{if } (X_1 \text{ is } A_1) \wedge (X_2 \text{ is } A_2) \dots \wedge (X_n \text{ is } A_n) \text{ then } k \in C_i \text{ with } m = m_{ki},$$

where A_1, A_2, \dots, A_n are linguistic terms characterized by appropriate membership functions for describing the features. The parameter m_{xi} represents the degree of belonging for an instance k to belong to a class C_i . In order to classify an unseen

input, first the degree of activation of each rule is computed, then the rule of the highest degree of activation determines the output of the classifier. Fuzzy rule-based approaches have been broadly used for online handwriting recognition, and are mainly used for online character recognition [70, 141, 148, 71, 5].

Rule-based techniques do not require a large amount of training data, and allow the number of features used to describe one class to be different from those required for another class. Fuzzy rule-based classification has succeeded in classification tasks for a small number of classes with a limited variety. However, their success in handwriting recognition applications has been restricted due to their inability to cover larger variations of handwriting styles or a bigger number of symbols. As the symbol size or variety grows, more rules are required to define the classes as separate. Increasing the number of rules leads to the difficulty in choosing the values for model parameters and in turn, leads to a slow inferencing or even a combinatorial explosion. The performance of rule-based methods is limited by the designer's abilities to reliably devise the set of rules. If too many free parameters are used, there is a danger of overfitting; conversely, with too few parameters the training set may not be learned. This issue has an impact on model complexity and the classification performance [93].

2.5.3 Prototype-based Methods

Prototype-based or template-based methods are more sophisticated versions of instance-based methods, in which an input sample has to be compared to all training samples for classification. Obviously, the time cost of these comparisons is a drawback of instance-based methods. To overcome this inefficiency, prototype-based methods use

a smaller yet representative set called prototypes (exemplars) instead of all training instances. There are different ways of choosing prototypes. Prototypes can for example be selected among training samples using selection techniques, or they can be formed by averaging the training samples of a class, on the intuition of representing the most likely or average characteristic of a class instance using replacement techniques. Sometimes instead of an instance, a prototypical description of the class is used. Clustering is usually useful in forming prototypes that represent different varieties in the data.

Often the classifier used in conjunction with rule-based methods is the k -nearest neighbor (k -NN). The k -NN classifier which was first introduced in [39], uses the majority votes of the k nearest labeled training instances to the test pattern for predicting the output class. As the number of training samples approaches infinity and $k = 1$, the error rate can be no worse than twice the Bayes error rate [68]. If k also approaches infinity, the error rate approaches the Bayes error rate. Additionally, lazy learners such as k -NNs do not use training data to form a class model or to estimate the underlying class probability distributions. Rather, they memorize all training instances and store them. Therefore, there is no separate learning phase for this type of classifier and computational costs pertain to the classification phase. The classification relies on the assumption that the input pattern is most likely to belong to the same class as instances which lie close to each other in the reference space.

The reference space for prototype-based methods can also be the feature space.

The similarity between two samples is defined as the distance between their corresponding feature vectors with respect to a distance metric (e.g. Euclidean or Manhattan). Alternatively, dissimilarity spaces, which are further discussed in Section 2.6.2, can be used. Given a similarity measurement between two patterns, an input pattern can be classified by its matching scores against prototypes of each class, for both feature and dissimilarity spaces. Mode detection between handwriting and a few basic geometrical hand-drawn shapes was performed in [179] by using a system consisting of k -NN classifiers in the feature space.

The fact that template matching-based systems do not require training, makes it easy to incorporate adaptivity and user customization into those systems. However, they have a number of limitations. Some of these limitations are rooted in k -NN shortcomings such as memory requirements for storing all training samples, computational cost (speed issue), and sensitivity to noisy data or outliers. In order to cover more writing styles, a larger number of prototypes are required. On the other hand, recognition time is linear in the cardinality of the prototype set. This is where prototype selection becomes useful. There have been some attempts for using prototype selection methods in the dissimilarity space [140]. Ordering and pruning of the prototypes prior to the final matching in combination with a two-stage k -NN classification was applied for reducing the recognition time in [173]. Prototype-based methods offer an easy user adaptation, however achieving acceptable error rates within a reasonable time frame is still a challenge for such systems.

2.5.4 Statistical Methods

Statistical classification is a supervised approach to classification. In this approach to classification, usually patterns are first mapped into a feature space, by measuring some of their characteristics. Ideally, in this new space, patterns of different classes lie in disjoint and compact areas. The feature space is partitioned into a finite number of regions corresponding to classes. The separating boundaries (decision boundaries) are used for performing classification. In statistical classification algorithms, typically two sets of data are used: training and testing. The training set of previously labeled instances are used for building decision boundaries (mapping functions), which are required for the labeling of unseen instances of a test set.

A supervised classification problem can be formally stated as follows: given a set of training data $X = (x_1, \dots, x_n)$ characterized by class labels $C = (c_1, \dots, c_n)$, produce a classifier (also called a hypothesis or model) $h : \{X\} \rightarrow \{C\}$, which maps an object $x \in \{X\}$ to its corresponding classification label $c \in \{C\}$.

There are two approaches for a supervised classification to generate a decision rule: *generative* and *discriminative*. A generative classifier tries to model the underlying class distributions, which produces the observed data. Alternatively, a discriminative classifier seeks to identify decision boundaries which provide an optimal separation of classes. In other words, generative models estimate the joint probability of feature vectors and the class label, while discriminative approaches directly model the conditional density of the class, given the feature vectors, without any assumption about the feature distribution. Discriminative models are often more robust than generative

models, because of making fewer assumptions [123].

There has been a shift of focus from prototype-based classification techniques towards statistical methods for recognition applications [159]. Hidden Markov Model (HMM) is a common generative classifier used in representing both the online handwriting primitives models (e.g. strokes or substrokes) [74] and the holistic pattern models [104, 17]. Artificial Neural Networks (ANN) is a common discriminative classifier which has been used for developing online handwriting recognition systems [180, 109, 105]. Support Vector Machines (SVMs) are state-of-the-art classifiers, in which the margin between classes is maximized. SVMs have been used for online handwriting recognition in [12, 1].

The effectiveness of the statistical classification depends on the quality of the feature space or the representational power of the used features. The more the selected feature set is representative, the easier it becomes to separate regions in the feature space. In general, a weakness of statistical classifiers is that they are tightly dependent on having representative training data. Statistical approaches generally require a larger amount of data for training. Besides, such classifiers must use a fixed number of features for all patterns in the multidimensional feature space. In general, statistical methods offer higher speeds under a larger memory in comparison to structural methods.

2.5.5 Hybrid Methods

Using a combination of classifiers instead of one classifier is a well-known method for improving the accuracy of recognition systems, although the question of how

to optimally combine multiple classifiers still remains. Some hybrid classifiers have been constructed for online handwriting recognition [15]. The methods used in the literature to combine classifiers are divided into two main categories: *abstract-level* and *measurement-level*. Abstract-level techniques address the idea of combining a level of information such as class labels and class ranking. For applications with a large number of classes, however, combining the labels of different classifiers may result in many confusing cases. Measurement-level techniques, on the other hand, combine such information as estimates of posterior probabilities and similarity measures. They have been proven as useful techniques for large classification problems.

Inspired by the idea that features and classifiers of different types could complement each other and consequently provide a better performance, some researchers have combined offline and online recognizers for online handwriting recognition [128, 106, 172]. Such systems are usually composed of two subsystems that use offline and online features for recognition. The final decision is usually made by combining the output of the two recognizers using probabilistic techniques. An advantage of such systems is that the stroke order variation does not affect the offline subsystem. In [172], the authors combined two separate HMM-based recognizers for online and offline features. In [128], the offline recognizer used the Modified Quadratic Discriminant Function (MQDF), and the online recognizer used elastic matching. A heuristic-based integration method was presented in [106] to combine the output of the two k -NN classifiers for recognition of one stroke characters. Aside from these classification methods, other indirect approaches will be discussed in the next section.

2.6 Indirect Approaches: Online Data Representation Methods

Online data can be represented qualitatively by structural methods or they can be represented (often quantitatively) by mapping the data to the feature space or the dissimilarity space. Each of those methods have a different philosophy behind the way they represent the pattern for facilitating the classification. Structural methods try to use the description of the pattern structure for classification. A pattern is encoded with some structures by using a finite set of building elements.

From the point of view of feature-based methods, similar patterns have similar measurable features, and feature-based methods can perform classification by mapping patterns to a suitable feature space and using the patterns' proximity in that space. From the point of view of dissimilarity-based methods, a suitable pairwise similarity measure⁴ can characterize a pattern by mapping it to a space called the dissimilarity space, in which similar objects are more likely to appear close to each other. The main part of designing a feature-based approach is the problem of extracting the most relevant information from the raw data (feature extraction). For dissimilarity-based approaches, devising a pairwise similarity measure is of equal importance. Sections 2.6.1, 2.6.2, and 2.6.3 have more details about structural, dissimilarity-based, and feature-based representations.

⁴A Similarity Measure is a function that associates a numeric value with (a pair of) patterns or sequences, with the idea that a higher value indicates greater similarity.

2.6.1 Structural Representations

Syntactic pattern representation, inspired by formal languages, is one way to represent patterns. In any formal language, sentences are formed from alphabets and grammars and rules. Similarly, in the syntactic approach, patterns are described in terms of primitives (sub-patterns) and relations among them. To represent a complex pattern, it is continually decomposed until it is transformed into primitives. The primitives are then represented hierarchically by strings, trees, or graphs.

With the increase in primitives, a more powerful pattern representation with a higher computational cost is achieved. The type of grammar used in syntactic algorithms may be deterministic, stochastic, fuzzy or hybrid. In [89], the authors used structural recognition techniques combined with some heuristics as a part of their system for recognition of isolated letters in the Devanagari script.

The most common structural representations for online trajectories are piecewise curves and strings of directional codes. Structural approaches require robust extraction of the primitives. Lack of a general approach for extracting primitives is one of the difficulties in implementing this approach. Difficulties in detecting primitives from noisy patterns and inferencing the grammar from training data are other limitations of this approach.

2.6.2 Representation in Dissimilarity Space

It has been argued that dissimilarity representation is more fundamental than feature representation [94]. Theoretically, dissimilarity classification does not operate on the class conditional distributions, hence the classification accuracy can exceed the Bayes

error. The Bayes error rate of a classifier is the probability of misclassification.

Elastic matching (EM), also called nonlinear/deformable template matching, is a vastly used pairwise similarity measure which is known for its robustness against pattern transformation. Dynamic time warping (DTW) is an EM method for measuring the overall similarity of two time-varying sequences of different sizes based on a measure of local minimum distance. DTW has been frequently used in speech recognition after it was introduced by Sakoe and Chiba [146]. DTW has been proven to be an efficient method to calculate the distance in online handwriting recognition [11].

In the area of online handwriting recognition, DTW has been used for digit recognition [138], Tamil character recognition [124], automatic extraction of handwriting styles [125], and verification of a large human-labeled data set of online handwriting [174].

DTW is an $O(N^2)$ algorithm, because every cell in the cost matrix must be filled to ensure an optimal answer is found, and the size of the matrix grows quadratically with the size of the time series [147]. Being an $O(N^2)$ algorithm is the major weaknesses of standard DTW. One solution to this problem is avoiding the DTW point-to-point matching between samples by simplifying the trajectory representation [79]. An inherent drawback of DTW (and all EM approaches), known as overfitting, results from their ability to tolerate pattern deformation. This problem has been denoted by some researchers [169, 116]. Overfitting happens when a test symbol is scored higher towards a wrong class rather than the class it belongs to (for example, digit 1 may be matched better to class 7). Overfitting cannot be excluded completely [116], although it can be less influential by incorporating probabilistic techniques such as statistical

dynamic programming.

2.6.3 Representation in Feature Space

Features are defined as measurements, attributes or primitives derived from patterns, that may be useful for their characterization [43]. The goal in feature extraction is to minimize the within-class pattern variation while enhancing the between-class pattern variation [45]. Features should be as invariant as possible to the expected distortions. Selecting a feature extraction method has been stated as the most important factor in the recognition performance [168].

Features can be broadly divided into global (high-level) features, and local (low-level) features. In order to learn about the online substrokes and their sequencing, several different features of both categories have been used in the literature. In this section, we cover important local and global features such as: coordinate, slope, length, significant points-related features, bounding box related-features, start/end points-related features, zone features, loop, curve approximation features, number of strokes, and length of strokes. We provide the definitions and variations of these features as well as their corresponding research.

2.6.3.1 Global/High-level Feature Representation

In computing a global feature, the whole sequence of trajectory points is used to extract a property related to the topology of the pattern. Some of the frequently used global features for a stroke include: number of ascenders, descenders, cusps, loops, mass center, start or end points, and bounding box information.

Global features may provide more descriptive information about a character than local features. However, these features usually come with a high computational cost. Global features are more powerful but less robust. The results of high-level feature extraction tend to be highly erroneous due to large shape variations in natural cursive handwriting, especially among different writers [75]. Therefore, methods that rely only on high-level features do not work well on unconstrained handwriting. The following list contains a summary of the six most commonly used global features:

- **Significant points-related features:** Some methods extract features from the identified significant/critical points such as high-curvature points or extreme points, instead of the whole trajectory⁵. The positional or directional features about such a point are used for describing a stroke/substroke.
- **Bounding box-related features:** Some methods consider using information provided by the stroke/substroke bounding box. The aspect ratio, which is the ratio of width to height of the original symbol, is a bounding box-related feature. The aspect ratio can be used as a global characteristic of the shape. The center of a stroke/substroke can be considered either as the bounding box center or the center of mass for the point sequence, described in the equations below:

$$(x_c, y_c) = \frac{1}{N} \left(\sum_{i=0}^N x_i, \sum_{i=0}^N y_i \right), \quad (2.1)$$

$$(x_c, y_c) = \frac{1}{2} ((x_{max} - x_{min}), (y_{max} - y_{min})), \quad (2.2)$$

⁵Trajectory is the path that the pen follows on the tablet.

- **Start/end points-related features:** The start point/end positional or directional information for a stroke/substroke can be used as features. Sometimes the difference between start and end point coordinates or the direction of the line connecting them are used as features. Sometimes the start/end positions are compared against a critical or maximal point of the corresponding stroke.
- **Loop:** The number of loops included in a stroke can be used as a global feature. The occurrence of a loop can depend on the style of writing. False loops may also form, when a user re-traverses a previously written loop.
- **Number of strokes:** The number of strokes or the number of pen-down/pen-up events can be used for multi-stroke symbol recognition.
- **Length:** Curve length is approximately measured by the sum of the Euclidean distances between consecutive points on the trajectory.

2.6.3.2 Local/Low-level Feature Representation

Local features are features which are calculated on a local area of a pattern. Local features for online handwriting are computed only by considering the trajectory points in a certain vicinity of a point. Local features for online handwriting are also called *point features* as they are assigned to each point along a symbol trajectory. Generally, local features are less informative but more reliable than global features [75].

Some typical local features include: digital curvature, cumulative curve length, point aspect ratio, and ink-related features (such as point density). Pen pressure and the time derivative of pen pressure are also used as local features [119]. Among

local features, the most applied ones are pen directional features and pen coordinate features. The following list contains a summary of the six most commonly used local features:

- **Positional and directional features:** *Positional features*, also called *co-ordinate features*, consist of the original or normalized values of the x and y positions of points in a sequence. *Directional features*, also called *tangent features*, usually refer to angles of every two consecutive points along the trajectory with respect to the horizontal axis. Often the sine and cosine of those angles are used as directional features.

The combination of directional and positional features has been proven to be a useful technique [3]. In [166], the authors compared direction features with the combination of direction and position features for Japanese characters, and concluded that the combination can vastly improve the recognition accuracy.

Point-to-point differences of directional and positional features have been used by some researchers [38, 25]. In [38], four point features were used for character recognition: the delta-positional feature (the changes in x and y coordinates from the previous point to the current point) and the delta-directional feature (the sine and cosine of the angle of incline for the line segment between two consecutive points). In [25], the authors used the directional features and their first order difference in combination with some other features for word recognition.

- **Coordinate-related features:** The normalized horizontal and vertical coordinates of the symbol, usually after re-sampling, have been widely used for

describing online handwriting data. Changes in the coordinates with regard to the previous point, or the maximum point, are other forms of using coordinates as features. Coordinate features are also known as positional features.

- **Slope-related features:** Direction of the pen movement is one of the essential information pieces embedded in online data. Different samples of a symbol, regardless of their sizes, should have similar pen directions. The slope is usually measured as an angle that a line forms with the horizontal axis (see Equation 2.3).

$$\alpha_i = \cos^{-1}((x_i - x_{i+1})/dist(P_i, P_{i+1})) \quad (2.3)$$

In case that slope is assigned to each point on the trajectory, as a time series of writing directions, it is called a *pen direction* or a *directional* feature. Directional features are sometimes quantized and presented in four or eight portions of a full circle in the form of chain codes. Another way to describe the writing direction is to measure the change of slope between two consecutive points or strokes, which is called a *turning angle*. A turning angle measures how far the angle is from being a straight angle. Turning angles and directional features are shown in Figure 2.2 as β_i s and α_i s, respectively.

- **Curvature:** Curvature of a curve is a measure closely related to the curve direction. If the direction remains almost the same when we move along a curve, the curvature is small. Whereas, if the curve undergoes a sudden and sharp turn, the curvature is large. The curvature at a point P_i on a planar curve has a magnitude equal to the reciprocal radius of an osculating circle at P_i .

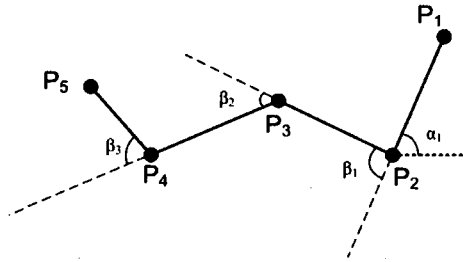


Figure 2.2: Directional features and turning angles for a discrete curve.

Curvature has been clearly defined in classical mathematics for continuous parametric curves ($c(t) = (x(t), y(t))$) or explicit curves ($y = f(x)$). For the discrete case of online handwriting trajectories, one approach is to transform the discrete to a continuous representation and calculate the curvature using the classical curvature definitions. This approach is done by applying curve fitting methods such as quadratic or B-spline curves to the sequence of points in order to extract a piecewise parametric representation. Another approach for approximation of the total signed curvature is the sum of the turning angles (see Figure 2.2), which has been formulated in Equation 2.4 [59].

$$k_{total} = \sum_{i=1}^N \beta_i \text{Sign}(\beta_i) \quad (2.4)$$

where

$$\text{Sign}(\beta) = \begin{cases} -1, & \text{if } \beta \text{ clockwise} \\ +1, & \text{if } \beta \text{ counter clockwise} \end{cases}$$

If the curvature is considered as a point feature, then for point P_j the summation in Equation 2.4 is calculated for $(i = j - 2)$ to $(N = j - n - 1)$ in order to account for the change of angle with regards to the n previous points.

- **Zone features:** Zone features can be computed by dividing the normalized

symbol into a certain number of zones. The total number of points in each region, as a replacement of the 2-D histogram in the image domain, are extracted as features. The zone information is useful in handwriting recognition [22]. The zone feature has also been used as a point feature [152].

Table 2.1: Online features used in the literature.

Feature	Research
pen coordinate	[3, 138, 21, 175, 106, 38, 11, 107, 115, 90]
pen direction	[3, 13, 21, 26, 27, 175, 157, 152, 106, 38, 11, 107, 115, 90]
pen pressure	[26, 27, 13]
curvature measures	[21, 67, 158]
bounding box	[176, 9, 10]
start/end points	[176]
zone information	[152]
significant points	[172, 38]
length	[9, 10, 63, 66, 65, 64]
loop	[126, 176]
curve approximation	[67, 36]
strokes count	[67]

- **Curve approximation features:** Some methods try to approximate the handwriting trajectory by piecewise curve fitting and by using the approximation coefficients as features. Polynomials and splines are usually used as approximating functions.

A summary of online handwriting features is shown in Table 2.1. In this table, the most widely used local and global features are listed. Another approach for representing online handwriting is using offline features in combination with online features. This approach to online handwriting representation is explained in Section 2.6.4.

2.6.4 Combining Online and Offline Features

Offline handwriting research is relatively more developed than online handwriting research. Furthermore, online data can be transformed to offline data by interpolating the point sequence and performing some morphological operations. This fact has inspired the application of the previous offline research to the online area [165]. Still some other researchers have worked on combining online and offline features. By the idea that putting the information of both categories of features together we obtain a more complete description of the handwriting input. Offline features extracted from an image often provide information about the strokes' spatial structures, whereas online features often provide dynamic information about strokes.

In [172], the authors used the pixel values of the image as offline features. In [128], offline features were extracted from the projection images in cardinal directions (north, south, east, and west). The four directional sub-patterns were used in [106] for offline feature extraction. A quad tree image decomposition was used in [165] for offline feature extraction. In [3], the authors demonstrated that a predetermined combination of online and offline features is not necessarily optimal, and they proposed a model that uses a weighted combination of offline and online features.

In this section, we have reviewed pattern representation methods and features used for online handwriting recognition. The variety of features demands for studies to investigate which ones are more effective. In the next section, we review studies on selecting the features.

As presented in previous studies, local, global, or offline features can be used

for online handwriting recognition. The importance of selecting existing features are studied in the topic of feature selection. In Section 2.7, we review the related research.

2.7 Studies on Feature Selection for Online Handwriting Recognition

In machine learning, feature/variable selection is the general name for methods that try to select a good combination of global and local relevant features for a particular learning task. There have been a few research methods on this matter in the literature, summarized in this section.

In [171], the authors used twelve mixtures of local and global features in a wrapper approach using a genetic algorithm (GA) search engine. Subsets of the UNIPEN dataset consisting of lowercase characters, uppercase characters, and digits were used in this study. Another feature selection method for recognition of online handwriting strokes was presented in [77]. This feature selection method was also tested on the digits, lowercase letters and uppercase letters of the UNIPEN database. The search engine was a sequential floating search engine, and the evaluating function was a hybrid HMMs-MLP recognizer. Using local features, a set of left-right HMMs produced a set of different labels with their associated maximum probabilities. This set of labels/probabilities, together with global features, were propagated to an MLP for recognizing unknown inputs. The feature selection procedure was performed in two phases for optimizing the local features, and for optimizing the combination of global and HMM features. A set of 19 different global and local features were chosen for this

study. Local features consisted of: substroke slope, turning angle, three inner angles, 8-directional chain codes, directional changes, aspect ratio, length, curvature, mass point and pen up/down. Global features included: character aspect ratio, number of strokes, character mass center, length of the mass center, character start and end points, and zone information. Among local features, the combination of five features: 8-directional chain codes, changed direction, aspect ratio, length, and mass point were identified as the best local feature combination by the system. Among global features, a combination of three features proved to be the best: number of strokes, length of the mass center, and zone information.

In [108], the authors studied 25 global and local features for whiteboard notes using a sequential forward search engine and an HMM classifier. The features were not independent and they might have provided redundant information. Their results showed that the best subset included 16 global and local features. An interesting observation was that a set of only five features could perform just as well as a combination of all 25 features. The best five features were: the cosine of the slope, the normalized y-position, the density in the center of the 3x3-matrix, the pen-up/down information, and the sine of the curvature. Some features such as the cosine of the curvature, the ascenders and descenders, the linearity, the curviness and the aspect of the online vicinity did not increase the performance and often made it worse.

2.8 Persian/Arabic Scripts

Most of the research in handwriting recognition has concentrated on Latin, Chinese and Japanese scripts. Arabic and its derivative scripts, which are used by a third of a billion people worldwide for writing, have had considerably less attention in the research. This shortage has to do with the difficulty of recognizing these scripts, lack of proper and publicly available databases, and lack of funding in this area. The shape of handwriting, regardless of the type of the script used, may vary due to several factors as mentioned previously (such as the writing style, writer individuality, writing input device, and others). However, Persian and Arabic scripts have specific characteristics which make the recognition task even more difficult. We discuss these characteristics in the following section. We will also provide a review of the literature for online handwriting recognition of Persian/Arabic scripts.

2.8.1 Characteristics of Persian/Arabic Handwriting

Persian/Arabic characters in both printed and written forms are cursive, meaning that the letters of a word are connected. In these scripts, disconnected block letters written in a sequence do not stand for a word, as in Latin. Most but not all of the letters in a word connect directly to the immediately following letter, along a writing line or baseline⁶. As a result of these characteristics, separating the letters is challenging.

Persian script, also known as Perso-Arabic script, is an important variant form

⁶Some Arabic letters such as Alif, Daal, Thaal, Raa, Zaay and Waaw do not connect to the following letter.

of the Arabic alphabet. Persian and its dialects have official language status in Iran, Afghanistan, and Tajikistan. Persian is also spoken by minorities in Uzbekistan, Turkmenistan, Azerbaijan, Armenia, Georgia, and Southern Russia. The Persian alphabet has thirty-two basic letters, and twenty-eight of them have been adopted from the Arabic alphabet. Several letters in the Persian/Arabic alphabet share the same basic form and differ only by a small complementary part [161]. A complementary part could be a dot, a group of dots or a slanted bar. It should be noted that changing the dots or small marks can turn one character into a different one. In Arabic/Persian, a letter can appear in up to four different forms: *isolated* (in a detached form), *initial* (those coming at the beginning of a word), *medial* (those forming a connection from the left and the right), and *final* (those forming a connection from the right, at the end of a word). These characteristics increase the effective size of the alphabet. The position of the character in the word and the previous character of the word (if there is any), are determining factors for the shape of a character. Figure 2.3 shows these variations for the Persian alphabet.

In Persian/Arabic, there are several styles for writing a letter or cursively connecting a group of letters. In other words, there are different allographs for letters and words. For example, Figure 2.4 shows six different ways of writing the same word. As this figure shows, both letters and their connections can appear in different shapes. Allographic variety is considered to be the toughest one among four different sources of variety denoted by L. Schomaker [151], as discussed in Section 2.3. Moreover, the affine transformation variation or geometrical distortion of Arabic/Persian scripts includes main shape distortion, as well as complementary part distortion. Main shape

Isolated	Initial	Medial	Final	Roman	Name	Isolated	Initial	Medial	Final	Roman	Name
ا	ا	ا	ا	á	alef	ص	ص	ص	ص	ş	sád
ب	ب	ب	ب	b	be	ض	ض	ض	ض	đ	zád
پ	پ	پ	پ	p	pe	ط	ط	ط	ط	ţ	tá
ت	ت	ت	ت	t	te	ظ	ظ	ظ	ظ	z	zá
ث	ث	ث	ث	th	se	ع	ع	ع	ع	'	ayn
ج	ج	ج	ج	j	jim	غ	غ	غ	غ	gh	ghayn
چ	چ	چ	چ	ch	che	ف	ف	ف	ف	f	fe
ح	ح	ح	ح	h	he	ق	ق	ق	ق	q	qáf
خ	خ	خ	خ	kh	khe	ك	ك	ك	ك	k	káf
د	د	د	د	d	dál	گ	گ	گ	گ	g	gáf
ذ	ذ	ذ	ذ	dh	zál	ل	ل	ل	ل	l	lám
ر	ر	ر	ر	r	re	م	م	م	م	m	mím
ز	ز	ز	ز	z	ze	ن	ن	ن	ن	n	nún
ژ	ژ	ژ	ژ	zh	zhe	و	و	و	و	v/ú	váv
س	س	س	س	s	sin	ه	ه	ه	ه	h	he
ش	ش	ش	ش	sh	shin	ی	ی	ی	ی	y/i	ye

Figure 2.3: Four different shapes of letters in the Persian alphabet.

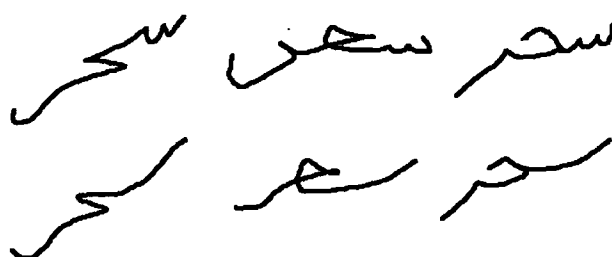


Figure 2.4: Different allographs for writing the same 3-letter word.

distortion refers to the slope of the main shape/stroke of the handwriting. Complementary part distortion refers to the change in the angle or position of one or more complementary parts with respect to the main stroke. These two types of distortion, which cannot be corrected in a single normalization operation, add up to even more variation. In [145], the authors presented more details about state-of-the-art methods in Persian script recognition. In the following subsection, a brief review of what has been done specifically in the field of online Arabic/Persian handwritten recognition is presented.

2.8.2 Classification and Data Representation for Online Persian/Arabic Handwriting

The earliest work in the field of online Arabic recognition started in 1989 ([51, 2, 50]). A rule-based classifier was used in this research. Among different classifiers that were discussed in Section 2.5, rule-based classifiers remained the most popular classifier for online handwriting recognition of Arabic/Persian scripts in the literature. A review of the currently existing recognition systems for online Arabic/Persian recognition, in terms of the employed classification methods, is presented below.

A hierarchical rule-based method was proposed for online isolated Arabic character recognition in [50]. The first level of the hierarchy was based on the number of strokes of the characters. Further division happened by using rule-based classification that relied on an error-free segmentation. This assumption made the system unable to handle noisy data or writing variations. In [52], a rule-based online recognizer of cursive Arabic handwriting was also proposed for simultaneous segmentation

combined with recognition of word portions by dynamic programming.

Fuzzy theory has also been very popular for recognition of Arabic/Persian online scripts, even recently. Bouslama et al. introduced a system using geometrical and structural features for describing Arabic letters in a piecewise manner, and using fuzzy techniques for recognition [24], [23]. More elaborate fuzzy-based systems have been reported more recently for Persian online recognition [9, 10, 63, 66, 65, 64]. All of these systems share a similar pattern representation approach. The approach is to segment a stroke into some curves (line, arc, or semicircle as in [64]; line, arc, highly curved arc, or half circle as in [63]; arc, semicircle, or circle as in [10]; line, arc, or semicircle as in [65]; and line, arc, or loops as in [9]). The segmentation is mainly performed by monitoring how the slope and the change of slope vary from their average values along the trajectory, and comparing those changes against two practically chosen threshold values. Some researchers have used a prior loop detection to this segmentation procedure [9],[10]. After segmentation, each segment is described by using some fuzzy linguistic terms for such geometrical measures of the segment, such as: segment start-to-end direction, segment type, curvature, length, aspect ratio, and relative positional change. Then, the stroke representation is produced by concatenating the segment into fuzzy terms in the original time order. The recognition part is based on a set of fuzzy rules [9],[63], or variations of the traditional fuzzy theory as such as fuzzy LVQ [10], and fuzzy elastic [64]. The performance of some of the fuzzy systems is compared with dynamic programming-based recognition systems in [65].

An error minimization method was proposed in [4] for recognition of online isolated Arabic characters, by using a bank of prototypes developed for the coded characters. The representation was based on directional codes and positional codes. A prototype-based system was also proposed in [79], using DTW for recognition of Persian subwords. This work also presented a segmentation method that suits the cursive nature of Persian scripts.

Using an MLP-NN classifier, the recognition of isolated Persian characters was studied in [143]. Isolated Persian letters were divided into 12 classes based on the number of dots, shapes, and locations of their diacritics. After recognition of the complementary parts and their locations by two MLP-NN's, the character's main shape was analyzed by another NN, if further recognition was needed. The reported recognition rate was 93.9%, using the database in [142]. This method assumes that the character's main body is written in the first stroke, followed by complementary parts. This assumption, although valid in most cases, is not always guaranteed. In [95], the authors compared the performance of genetic programming-based recognition, and an MLP classifier for the main strokes of Arabic letters.

A clustering method based on Self-Organizing Maps (SOM) was used to recognize the basic shapes of online isolated Arabic characters in [112]. SOM is a type of neural network, known for producing a discretized low-dimensional (usually one or two dimensions) representation from high-dimensional data, by providing a map that places similar data in close positions [96]. The SOM is usually a two-layer, fully connected NN that is trained by using unsupervised learning. In order to improve the system performance in [110], a combination of two separately trained SOMs was

considered, one using a tangent vector and the other using a Fourier descriptor [110]. For a better accuracy, the two SOMs were pruned and filtered. As the similarity measure used by the SOM was computed as a distance, two other distances specifically the Kullback-Leibler divergence and the Hellinger distance were used as replacements for the Euclidean distance, for training SOMs [113]. Results showed that the Hellinger distance improved the performance by 0.5%. All SOM-based methods developed in [112, 110, 113] showed lower performances than the 1-NN clustering methods that used the same features. SOMs get more computationally expensive as the dimensions of the data increase. SOMs are also sensitive to initial conditions.

In Table 2.2, the most recent work on Arabic/Persian online handwriting is presented. This table summarizes information about the methodologies defined by features and classifiers used, performances with respect to recognition rates and recognition times reported, and the comparative results in the literature (if any).

Table 2.2: Summary of methods, performances, and comparability of previous research in Arabic/Persian online handwriting recognition.

Ref.	Classifier	Feature	Accuracy%	Recog. time	Compared results
[112]	SOM	trajectory Fourier descriptor	88.4	not reported	1-NN clustering
[110]	SOM	trajectory Fourier descriptor	93.5	not reported	authors' own work [112]
[111]	SOM	empirical directional feature histogram	95.3	not reported	authors' own work [112, 110]
[113]	SOM	empirical directional feature histogram	95.3	26 s	authors' own work [112, 110, 111]
[114]	Bays	empirical directional feature histogram	92.6	not reported	authors' own work [112, 110, 111, 113]
[95]	SOM,MLP,GP	coordinate,time	95.0	not reported	none
[20]	HMM	directional,loop	96.3-word	not reported	none
[7]	DTW	coordinate,direction,curvature,aspect ratio	91.0	not reported	none
[53]	string matching	directional,length	95.0-char	not reported	none
[9]	Fuzzy	directional,length,curvature,aspect ratio	75.0	not reported	authors' own work [66]
[63]	Fuzzy	directional,length,curvature	77.0-char	not reported	none
[10]	Fuzzy LVQ	positional,directional,curvature,aspect ratio	90.0	not reported	authors' own work [66, 9]
[64]	Fuzzy Elastic	directional, length,curvature	78.0	not reported	none

In this chapter, we presented the basics in the field of online handwriting recognition, as well as the commonly used classification and feature extraction strategies for online handwriting recognition in light of the methods for improving a system for online handwriting recognition. We also reviewed a number of works in online handwriting recognition focusing on the Arabic/Persian scripts. As our survey shows, so far the features used are either purely local or global, and the relation of the local and global attributes has not been taken into consideration for developing a descriptive representation.

Furthermore, we saw that most of the methods and strategies proposed for those scripts used small databases. The fact that there is no comprehensive database for Arabic is an obstacle. In the next chapter, we will discuss our developed tools for collecting online data samples, and the database developed based on such tools for online Arabic characters.

Chapter 3

A New Online Arabic Database

Chapter Outline

In this chapter, we present the design and development of a new database for Arabic online handwritten character patterns. We describe the characteristics of this database and provide illustrative examples from the real samples. We also discuss the tools and policies used to collect the patterns. We also provide the statistics of the contributors to our database.

The Arabic Language is the native language of more than 230 million speakers and is the official language of fifteen countries in the world, mainly in the geographical areas of the Middle East and Africa. It is one of the six official languages of the United Nations [46] and the tenth language in the world with respect to the number of internet users [60]. Arabic is the second most widely used alphabetic writing system in the world after the Latin alphabet [132]. Despite this popularity, in the context of handwriting recognition research, there has been little effort in building an online handwriting database for Arabic and similar scripts (Persian, Pashto, and Urdu). The reason behind this may be due to the lack of research funds, and the delay of

technology growth in the regions of the Middle East and Africa.

The databases developed in the literature for Arabic and Persian scripts are mainly small in the number of contributors and/or include a limited number of collected samples (see Table 3.1). A comprehensive database should consist of a large number of samples written by a large number of writers to capture different variations and variabilities ¹ exhibited in real handwriting patterns.

It is important to have standard databases available to all researchers for the performance comparison of their developed methods and models, to be used in a consistent manner. Currently, almost all researchers have examined their ideas and reported their results on small private databases collected by the contribution of a small number of writers. To the best of our knowledge, there is no publicly available online Arabic/Persian data repository for researchers to use, as a benchmark to evaluate the recognition methods and compare the results. Some researchers do not wish to share their databases even upon request (e.g. our attempt to obtain the databases of some reported outstanding systems such as in [20] for comparison purposes failed). Therefore, in most works, results are never compared with other available techniques in the literature, or are only compared with the previous publication from the same authors, using their own databases. We present the current state of online Arabic/Persian databases for the purpose of comparison with the database developed in this thesis.

¹See Section 2.3 for definitions of variation and variability.

3.1 Online Arabic/Persian databases

Many recognition systems have been introduced in the last three decades for online Arabic/Persian handwriting recognition. Research considerations and empirical results suggest that these systems perform differently when applied to different datasets. However, evidence describing the performance of these systems remains fragmentary when tested with different datasets.

Based on our survey of the databases in online Arabic/Persian handwriting, there is no publicly available database with online access. The only two databases available by request from researchers who developed them are described in [142, 114]. Razavi and Kabir's database [142] includes a set of the most frequently used Persian subwords. Prior to database collection, the authors investigated the most repetitive Persian words in daily usage. In this work, a set of 30,000 words which have appeared more than 30 times in an archive consisting of several Persian newspapers were extracted as frequent words. Breaking those 30,000 words into subwords resulted in less than 7,300 subwords, since many of these subwords were common among the words. From these subwords, authors collected the most frequent 1000 subwords. In addition to subwords, this database contains Persian digits and isolated letters (124 samples per symbol). About 124 people contributed to this database. Despite the commendable effort in identifying the most frequent words and subwords in daily usage of Persian script, not a sufficient amount of samples were collected for each subword. On average, just 12 samples for each subword are included in this database. The database by Mezghani et al. [114] is a representative database for the main shape

of isolated Arabic letters. From the 28 characters that complete the Arabic alphabet, one can construct 18 distinct shapes. In an earlier version of this database [112], two of these shapes were combined into one group (letters Fa and Qaf), making 17 classes in total. The former database includes about 7400 samples, and has been constructed by the contribution of 17 writers. In the latter database [114], the letter kaf is divided into two allographs (one or two strokes) and the number of contributors has increased to 22. The minor increase in the number of classes has decreased the recognition rate from 95% in [112] to 92% in [114]. This observation suggests a high degree of similarity between classes, as expected.

Table 3.1: Summary of database specification used in the literature for Arabic/Persian online handwriting.

Research	Script	Data type	Availability	#writers	sample/class
[112]	Arabic	char/uni-stroke	upon request	17	~435/17
[110]	Arabic	char/uni-stroke		17	~432/17
[111]	Arabic	char/uni-stroke		18	~432/17
[113]	Arabic	char/uni-stroke		18	~432/17
[114]	Arabic	char/uni-stroke		22	~528/18
[95]	Arabic	char/uni-stroke	not available	not reported	~107/15
[20]	Arabic	word/multi-stroke	not available	10	not clear
[7]	Arabic	char/uni-stroke	not available	not reported	not reported
[53]	Arabic	word/multi-stroke	not available	8	not clear
[9]	Persian	character	upon request	~124	~124/32
[63]	Persian	character-word	not available	not reported	1
[10]	Persian	character	upon request	~124	~12/32
[64]	Persian	word/uni-stroke	not available	20	not reported/1250

Table 3.1 presents the information about the datasets, which have been used in the literature for the recognition of Persian/Arabic scripts. The data type identifies if the

data consists of characters, words, or subwords, and also if the samples contain a uni-stroke or a multi-stroke. Information about whether or not the database is available online and whether or not it could be obtained are showed in the next column. The number of contributors to each dataset, and the number of samples per class for each symbol are also provided in Table 3.1. Based on the information presented, the lack of a diverse and sufficiently large database is apparent. In the next section, we explain some details about the type of data, device used, and specification of the database developed for Arabic letters.

3.2 Design of the New Database of Complete Arabic Characters

To address the limitations in the current state of the research in online Arabic handwriting recognition in terms of database resources, we have developed a new database of online Arabic characters. The database contains over 10,000 entries from a wide variety of writers to cover the variability and variations that exist in human handwriting as much as possible. Figure 3.1 shows the isolated letters of our database and their corresponding class labels.

We used a digital Wacom Graphire Bluetooth tablet (Figure 3.2), which has a frequency of 100 points/sec and a resolution of 23 points/cm. When the tablet is connected to a computer, every point on the tablet corresponds to a point on the computer screen. Some software tools are developed in order to facilitate and automate data collection, sample visualization, ground truth formation, and necessary

خ	ح	ج	ث	ت	ب	ا
7- Kha	6- Haa	5- Jiim	4- Thaa	3- Taa	2- Baa	1- Alif
ص	ش	س	ز	ر	ذ	د
14- Saad	13- Shiin	12- Siin	11- Zaay	10- Raa	9- Thaal	8- Daal
ق	ف	غ	ع	ظ	ط	ض
21- Gaaf	20- Faa	19- Ghayn	18- Ayn	17- Thaa	16- Taa	15- Daad
ي	و	ه	ن	م	ل	ك
28- Yaa	27- Waaw	26- Ha	25- Nuun	24- Miim	23- Laam	22- Kaaf

Figure 3.1: Isolated letters from the Arabic alphabet and their corresponding class numbers in the database developed in this thesis.

verifications for building the database. These tools will be explained in the next section.

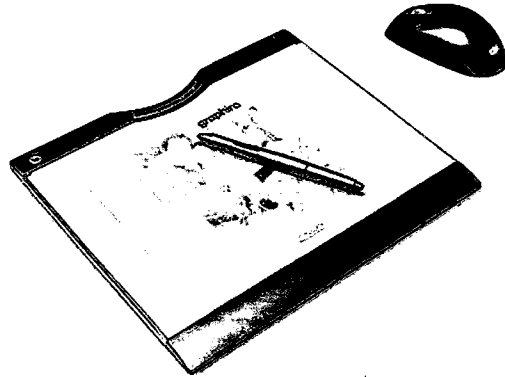


Figure 3.2: Photograph of the type of tablet and accessories used for collecting samples in our database.

3.3 Tools Developed for Data Collection and Visualization

In order to collect the online data from individuals with different backgrounds and different levels of familiarity with computer peripherals, we needed easy and flexible tools which would let the writer work comfortably. As a software application for recording the coordinates was not provided by the manufacturer, we had to develop our own software for recording and visualizing the data. These tools are explained in Sections 3.3.1 and 3.3.2, respectively.

3.3.1 Data Collection Software

Our data collection software included a user-friendly interface, which allowed us to save the data with the appropriate ground truth. Ground truth refers to the information encoded in the assigned label to each instance of a dataset. In order to run the data collection tool, parameters recorded as ground truth had to be passed to the application. While a contributor wrote the characters in alphabetical order, the file's ground truth was changed accordingly, by the application. This feature facilitated the data collection and eliminated any effort that may have been required for creating ground truth formation after the data would be collected.

We developed this software using GTK² under the Unix system. Figure 3.3 shows a screen shot of this application. It records the pen movements as (x, y) coordinates of the pen on the plane. The Clear button clears what has been written, without saving the current data sample. When the New button is pressed, the current data is

²GTK (GIMP Toolkit) is a cross-platform widget toolkit for creating graphical user interfaces.

saved, the window is cleared, and the file name is automatically changed for the next character. The Quit button saves the current data and closes the application.

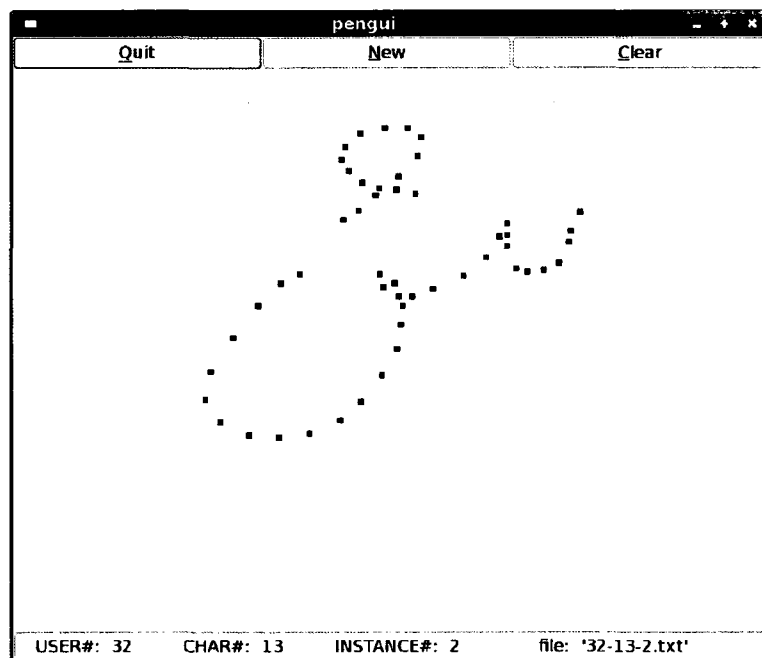


Figure 3.3: long
Screen shot of our developed software application used for our database collection.

We have imposed no restriction on the size of the written samples. In our collecting tool, the writing area is a window of size 800×600 pixels with no indication of the baseline. Therefore, the slant i.e. the rotation of symbols have been left as a degree of freedom to the writer as well. We now present some of the viewing tools we designed.

3.3.2 Viewing Tools

In order to visualize the collected data easily for verification and analysis purposes, we have developed a software application for our designed data format. This tool

draws the symbol, which each binary file encodes as an image while the interface shows the corresponding ground truth. Our viewing tool also allows the viewing of a group of files at the same time in a vertical scrolling window.

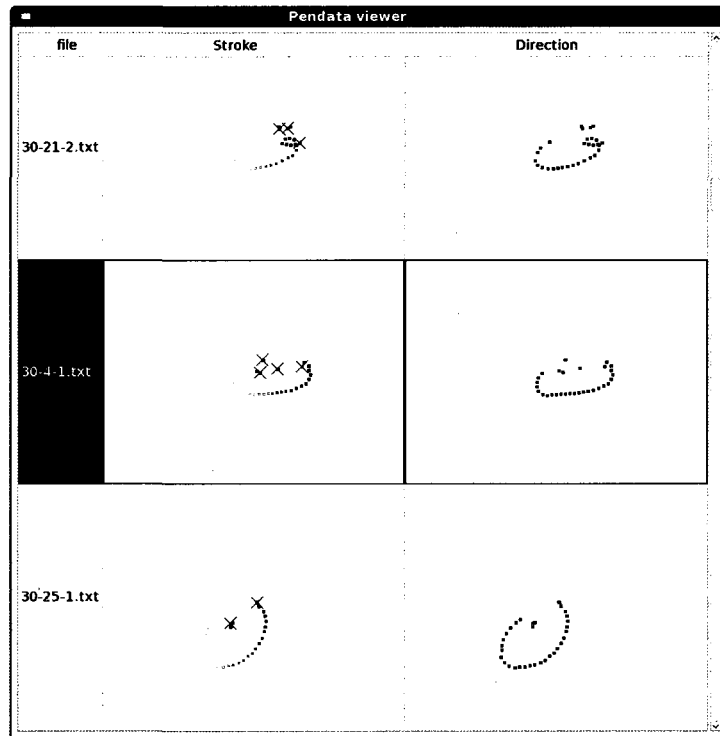


Figure 3.4: A snapshot of the software application developed for visualizing the data samples, with two modes showing Arabic letters Gaaf, Thaa, and Nuun, from top to bottom, respectively. The two columns: Stroke and Direction represent two complementary modes for visualizing each letter. The former depicts the starting point and direction of each stroke, while the latter depicts the order of strokes.

In order to facilitate the analysis of samples, two different modes have been designed for the viewer. The first mode has been designed for easy viewing of different strokes for each of the samples, while the second mode has been designed for easy viewing of the writing directions. Stroke viewing mode draws the time-ordered strokes of a symbol using a fixed sequence of seven colors (black, blue, red, green, yellow,

aqua, and fuchsia). If there are more than seven strokes then the colors will repeat. Direction viewing mode uses the effect of fading colors for showing the direction of writing, while it identifies the start point of each stroke with a cross symbol. Figure 3.4 shows this viewer with both modes. In this figure, we observe the three Arabic letters: Gaaf, Thaa, and Nuun, in the three rows of the table. Each letter is shown by the two modes of the viewing tool. We can distinguish between different strokes and the direction of each stroke in the left-side column. In the right-side column, the colors represent the degree to which a stroke is a major one, with black being the major stroke and blue, red, and green being time-ordered complementary strokes (see Section 2.8 for definitions of major and complementary strokes). In the next two sections, we review the policies used for the database collection and characteristics of the new database.

3.4 Policies used for sample collection

When the tablet was set up, and collection tools were ready, a pre-printed text was provided to each individual, to explain the purpose of collecting online handwritten character patterns. If the subject agreed with the purpose of the study and its usage for academic research on character recognition, the writing method was explained and the collection started with the subject writing a profile. During the sample collection, only missing or wrong patterns that the subject marked as incorrect, were requested to be rewritten. This preserved the writer's natural writing style as much as possible, while decreasing the noise related to missing data. We needed to collect

a considerable number of patterns from each individual. Therefore, some writers may have written patterns that were intended to be easy or difficult to recognize. Each subject was initially requested to write in his/her own normal style, without any restrictions imposed on the used allograph or on the quality of patterns.

3.5 Characteristics of the new database

In this database, all 28 classes of Arabic characters have been collected in their complete character forms. The writers have been chosen in such a way that different variations in writing styles were included. We considered writers of different origins, who were native speakers of Arabic, Persian, or Urdu with different levels of education, age categories and genders. We collected the data from 150 different individuals with left-handed and right-handed writing habits. Writers were asked to write the complete Arabic alphabet between two to five times each. We have also taken many samples from each individual continuously to account for the possible changes in their tiredness, motivation and mood. Table 3.2 demonstrates the descriptive statistics related to sample diversity. In total, there are 360 samples for each character, and over 10,000 entries in the database.

Table 3.2: The statistics of our database.

Native Language		Gender		Age		Handedness		Education	
Persian	40%	Male	41%	less than 18	11%	left-handed	3%	University	54%
Arabic	55%	Female	59%	18-65	73%	right-handed	97%	Non-university	46%
Urdu/Pashtoo	5%			more than 65	16%				

3.5.1 Data Variation

During the sample collection, there was evidence of real handwriting variations when we noticed that some beliefs about writing styles could be violated. For instance, it is commonly believed that Arabic is written from right to left (especially for the main stroke of the character/word), however, some contradictory examples were observed in our samples. Figure 3.5 shows a real sample of the letter Zaad from an Arab native individual, in our database. The cross symbols depict the starting point of each stroke. This letter has a major stroke presented in the third column on the right and a complementary stroke as a dot shown in the first two columns on the left.

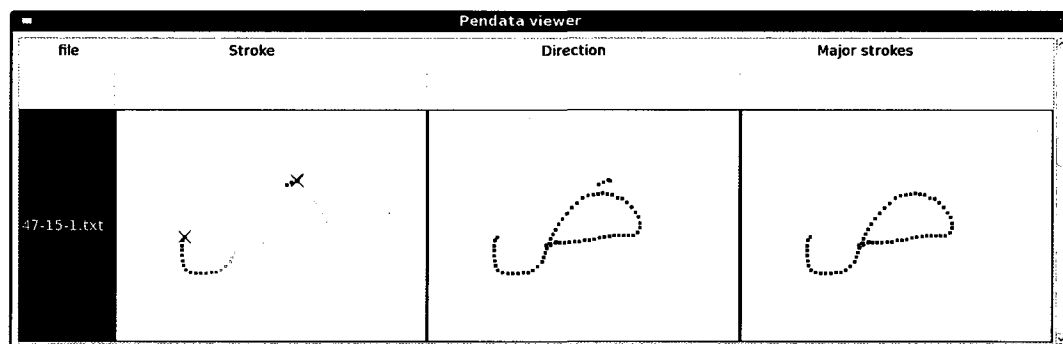


Figure 3.5: A real example of an Arabic character Zaad being written in its main stroke from the left to right. The first and second columns from left depict the full character with the two modes of our viewing tool, and the third column shows only the major character stroke.

In the far left column, the x symbols show the beginning of each stroke and the fading of colors indicate the direction of writing. As Figure 3.5 shows, the writing direction is from left to right for the main stroke of letter Zaad. This type of observation suggests that making assumptions about the direction of a stroke may

lead to errors for user-independent online recognition systems, which rely heavily on directional information.

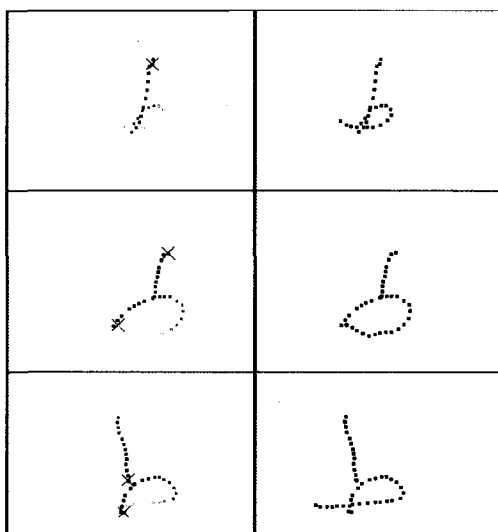


Figure 3.6: Three samples of the Arabic letter Taa in our database, written in different directions and with a different number of strokes by three different individuals. Each sample is shown in a row using two modes of our viewing tool.

Another interesting observation during data collection was the writing variation in terms of the number of strokes, the order of strokes, and the direction of strokes. As Figure 3.6 shows, a single letter Taa with the same final shape has been written in one or two strokes by three different individuals. The columns of the table in this figure present different visualizations of the same letter, and the rows show samples written by three different individuals. In the first row, the letter is written in only one stroke, and in the second and third rows it is written in two strokes but in different directions. Such observations suggest that there are variations, even among samples

Table 3.3: The statistics about the number of strokes in each class of our database (in percentages).

Class	Number of strokes					Class	Number of strokes				
	1	2	3	4	≥5		1	2	3	4	≥5
1	41.38	47.84	7.76	2.16	0.86	15	0.22	43.07	47.19	6.49	3.03
2	0.65	43.7	45.43	5.22	5	16	3.25	43.72	43.51	5.84	3.68
3	0	21.65	43.72	26.19	8.44	17	0	3.9	42.86	42.21	11.04
4	0	6.47	21.55	37.72	34.27	18	41.99	49.57	8.01	0.43	0
5	0	40.73	46.55	8.19	4.53	19	0.22	42.03	48.06	7.33	2.37
6	42.46	48.28	7.11	1.72	0.43	20	0	42.86	47.19	6.49	3.46
7	0	41.38	46.77	7.54	4.31	21	0	21	45.45	27.06	6.49
8	44.83	49.14	5.17	0.65	0.22	22	8.66	41.77	38.53	6.93	4.11
9	0.22	43.97	48.28	5.6	1.94	23	42.83	48.7	7.17	1.09	0.22
10	45.48	49.14	4.31	0.43	0.65	24	42.29	48.24	7.71	1.54	0.22
11	0.22	44.18	48.71	5.6	1.29	25	0	43.26	47.83	6.3	2.61
12	43.1	48.49	6.9	1.51	0	26	44.37	49.35	5.41	0.65	0.22
13	0	9.7	23.49	33.41	33.41	27	44.81	49.57	5.19	0.43	0
14	42.21	47.84	7.58	1.95	0.43	28	33.98	41.77	14.07	7.79	2.38

of the same allograph.

For all of the samples in our database, we measured the number of strokes written for each letter. Table 3.3 shows this information for all classes of the database. The class number corresponds to the character shapes, as presented earlier in Figure 3.1. Each character could be written in one, two, three, four, and five strokes or more. The numbers presented in each column show the percentage of the corresponding number of strokes per letter by writers in our database. In total, there were nine classes of letters which were never written by a single stroke.

3.5.1.1 Complementary Parts

Complementary parts of a letter could also be written in different shapes and with a different number of strokes. This is shown in Figure 3.7 for an Arabic letter Shiin, which has three dots. As this figure shows, besides the obvious difference in the shape of the main stroke, dots could be written with three strokes as three single dots (as

shown in the first row), with two strokes as a dot and a curvy/straight line segment (as shown in the second row), or with only one stroke as a circle/half-circle (as shown in the third row).

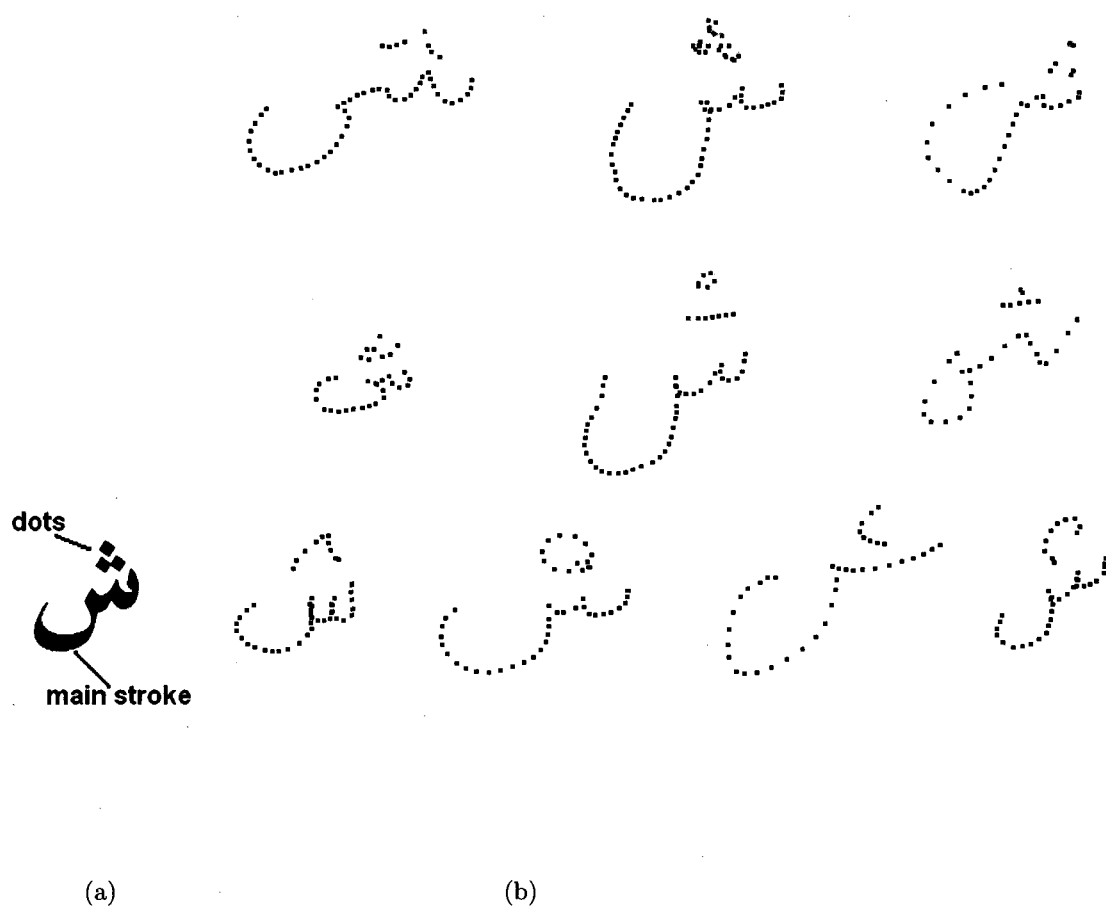


Figure 3.7: (a) Arabic letter Shiin in typewritten form. This letter has three dots that appear above the main stroke of letter, (b) ten different forms of writing the dots of letter Shiin by ten individuals. Dots are written in three, two, and one stroke(s), for the samples shown in the first, second, and third rows respectively.

Variations in forming loops, critical points, or high-curvature points with different writing styles have also been observed in our samples. This observation verifies that such global features, even for a particular allograph, can be misleading and can reduce

the robustness of recognition. Figure 3.8 illustrates this observation as variations of cusp points and loops between six individuals in writing the Arabic letter Jiim. As this figure shows, for the top two samples a loop has formed where the head part is connected to the body part. The two middle samples have a cusp point where the head and body are connected. The bottom two samples have sharp points but only in the body of the letter. No loop is formed in the main stroke of the four bottom samples.

Another observation was that two different letters which are technically supposed to be written differently could look the same when written by some individuals. Those similar classes usually create more confusion for recognition systems. We show this difficulty by presenting real samples from our database. Figure 3.9 shows how samples of the letters Thaal and Zaay can look like each other. The sample shown in Figure 3.9(a) represents Thaal, while it looks like Zaay. The sample shown in Figure 3.9(b) represents Zaay, while it looks like Thaal. The two samples shown in Figure 3.9(c) represent Thaal and Zaay, while they look very similar.

Figure 3.10 shows how samples of the letters Faa and Waaw can look like each other. Samples shown in Figure 3.10(a) represent Faa, while they look like Waaw. Samples shown in Figure 3.10(b) represent Waaw, while they look like Faa.

3.5.2 Data Format and Verification

For each sample we needed to record the pen movements as (x, y) coordinates. Besides, samples may have been written in a uni-stroke or multi-strokes. Therefore, we needed a way to specify where in the sequence one stroke has ended and the next

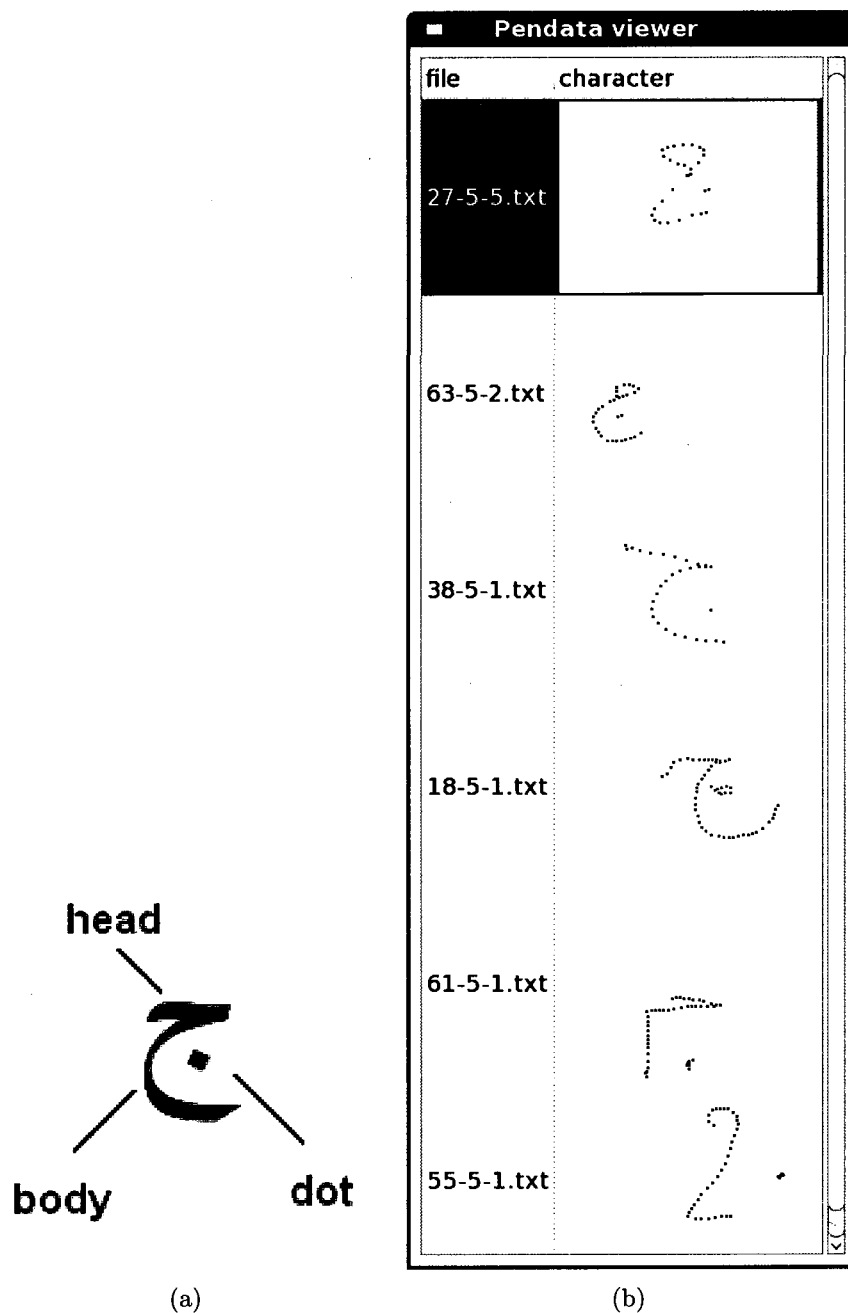


Figure 3.8: (a) Arabic letter Jjim in typewritten form. The main stroke of the letter is composed of two curved parts: head and body, (b) Variations in terms of formation of loops and cusp points in the main stroke of the letter Jjim (class 5) for six different samples written by six individuals contributing to our database.

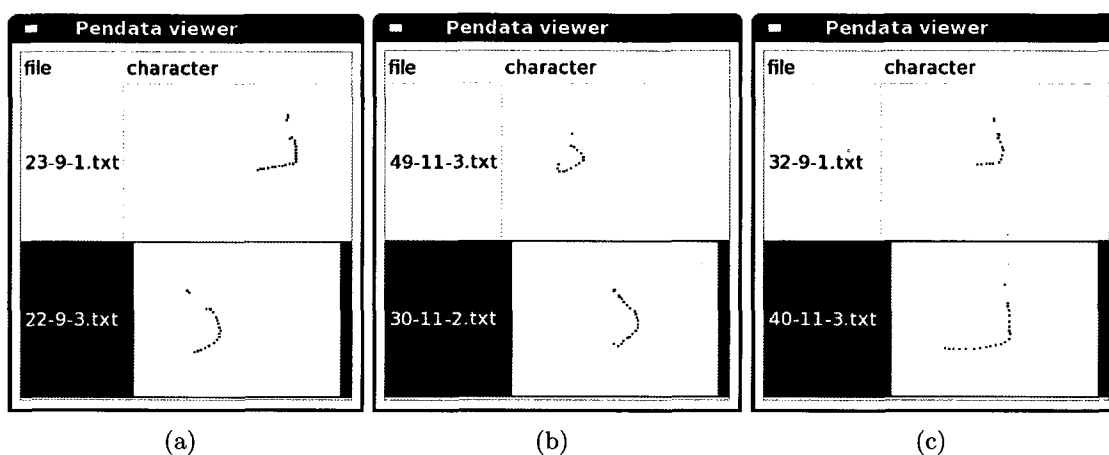


Figure 3.9: Similarly written letters: (a) Samples of the Arabic letter Thaal (class 9) look like letter Zaay, (b) Samples of the Arabic letter Zaay (class 11) look like letter Thaal , (c) Samples of letters Thaal (top) and Zaay (bottom), written in a similar shape. Letters were written by six different people.

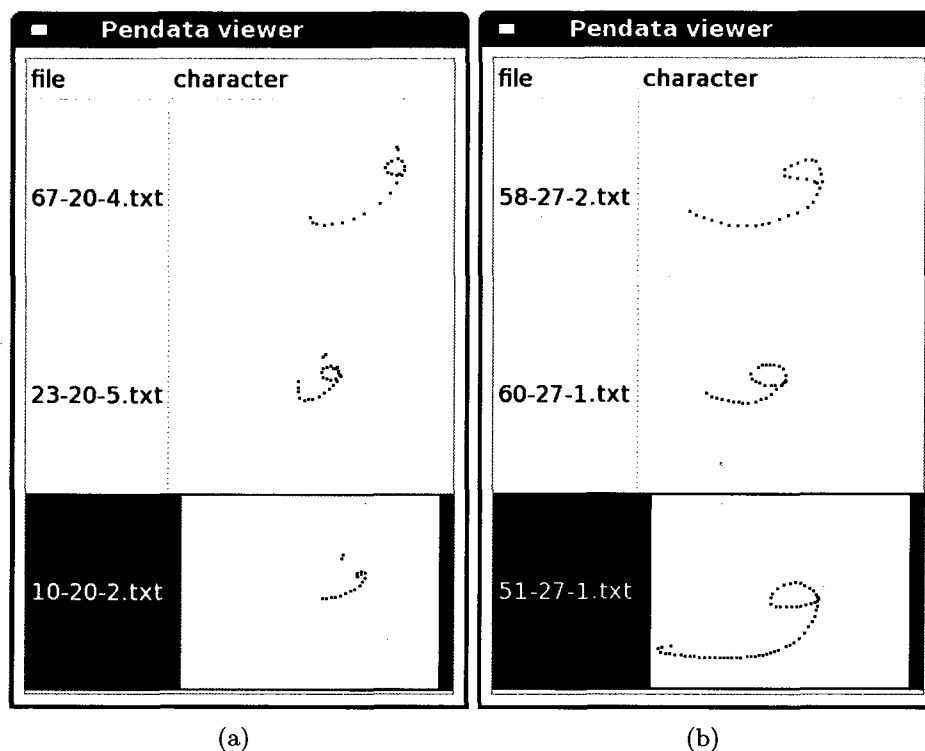


Figure 3.10: Different letters which are written similarly: (a) Three samples of the Arabic letter Faa (class 20), (b) Three samples of the Arabic letter Waaw (class 27). Letters in each row, although they belong to two different classes that are not alike in typewritten format, they look very similar in their main shapes in handwriting. Letters were written by six different people.

stroke has started. To record the event of pen-up or release, we used the values of $x = -1$ and $y = -1$, which could not be taken as valid pen coordinates during normal pen movements within the designed window. Recording pen-up in the same format as the pen coordinates allowed for easy and fast data manipulation. Each sample was recorded as a binary text file. The ground truth for each sample encoded the information about the writer's ID, the sample ID, and the repetition count for the sample. The sample ID encoded the class where sample belonged. The formation of the ground truth made it easy to extract the samples from a specific writer. This ground truth is useful for the purpose of adapting a specific writer to the system.

During the process of data collection, sometimes users made mistakes or forgot to write one of the symbols. As the ground truth progresses sequentially for the alphabet in our software, correcting such mistakes requires restarting the application with the right value for the label parameter again. A mistake anywhere in this correction process resulted in some mislabeled patterns. Therefore, we used a clustering algorithm to find the outliers. These outliers could then be verified manually for correct labeling.

3.5.3 Sections of the Database

Most of the time for multi-stroke letters, one stroke is the main one and the others are complementary. Some research studies are designed for the recognition of the main shape of characters, and some databases are built for this purpose [114]. To be able to offer a general purpose data resource which could also be used for main shape recognition, we considered two separate sections in our database consisting of

multi-stroke and uni-stroke characters, which were prepared in 28 and 17 classes, respectively. The main strokes were extracted from the multi-stroke version and were saved with the appropriate labels. Usually the main stroke is written by a writer prior to writing the complementary ones. However, as we did not impose such a condition for developing our database, for some writers such an assumption failed in practice. Therefore, our approach to main stroke extraction did not rely on stroke

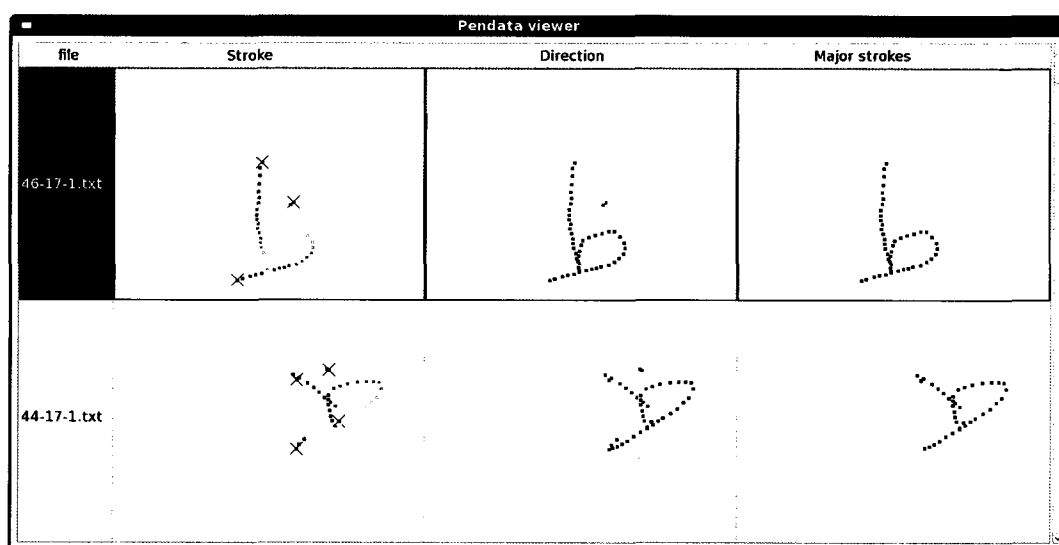


Figure 3.11: Each row shows a sample of multi-stroke Arabic letter Thaa. In the first two columns from the left, letters are shown under two modes of our viewing tools. The main shape of each letter is extracted as a single stroke, and is shown in the right-most column.

order. Instead, we used proportional stroke sizes to discriminate between the main and complementary strokes as follows:

- we computed the maximum length and width and the number of points for all strokes of a single sample;
- we normalized these values with respect to their maximums among the strokes

of all the samples to get ratios;

- a threshold value was selected based on the size statistics extracted from all samples in the database; and
- the complementary strokes were then determined by thresholding the ratios against a fixed value.

In order to examine the results of our algorithm for extracting the main stroke, we developed another version of the viewing tool, which enabled us to view the multi-stroke character, and its uni-stroke counterpart at the same time. Figure 3.11 shows two samples of the letter Thaa. In this figure, the first and second columns show two visualizations of the samples plotted by our viewing tools, and the third column shows the main shape extracted.

In this chapter, we discussed the development of a database for isolated multi-stroke online Arabic characters. We presented the designed tools for collection of the data, which allows for rewriting the written text, and automatic ground truth registration. We also described our visualization tool for online data. Using this tool for viewing a multi-stroke sample, one can see the time order of writing the strokes as well as the number of strokes. We also presented the developed tool for the extraction of the main stroke from a multi-stroke instant. Using our implementation, we created a uni-stroke section for the database, which contains only the main stroke of the samples. This section of the database is used in our experiments, which will be presented in the next two chapters.

Chapter 4

Relational Histogram Feature

Representation in a

Neural-Network-Based Recognition

System

Chapter Outline

In this chapter, we propose a new representation for online Arabic characters called relational histogram. This representation allows for the expression of knowledge regarding characters' shapes, with reasonably robust features. For this representation, we also provide the corresponding feature extraction algorithm. This representation is used with a neural network classifier in an online recognition system. Our described method does not need any segmentation of the character into sub-components and can empirically show improvements over the best reported recognition rate on the same online Arabic database.

In a recognition system, in general, the recognition performance depends on the efficiency of techniques and methods adopted for pre-processing, feature extraction,

and learning classification. Artificial Neural Networks (ANNs) have shown good capabilities in performing classification tasks in many applications. However, classifier models used for learning in pattern classification are challenged when the differences between the patterns of training data are small. This can happen partially as a result of using a poor pattern descriptor. However, selecting descriptive features to represent a character is more crucial to both learning and recognition than the choice of the classification method. Therefore, the choice of effective features is mandatory for reaching a good performance. An effective feature representation should allow the classifier to distinguish between allographs of all characters. Features must also be high-level enough so that the classifier assigns the same character to the same class, when written by different people at different speeds, different proportions (shorter or longer strokes), and different deformities (in head, tail, loop, or curve).

The research in online handwriting recognition shows that statistical and geometrical features alone are not sufficient for the recognition of handwritten characters, due to shortcomings in capturing the variations in writing styles. This type of feature extraction may result in deformations of character shapes. We try to address this problem by using a relational feature combined with a local descriptor, for training a neural network-based recognition system in a user-independent online character recognition application. Our feature extraction approach provides a rich representation of the global shape characteristics, in a considerably compact form. This new relational feature provides a higher distinctiveness compared to techniques previously suggested for online Arabic character recognition in the literature and shows robustness to character deformations. While enhancing the recognition accuracy, the

feature extraction is computationally easy. We show that the ability to recognize online Arabic handwritten characters can be increased by adopting this mechanism, which provides input to a feed forward neural network architecture.

A character recognition system includes four discrete blocks: (1) pre-processing, (2) normalization of global handwriting parameters, (3) extraction of features per stroke, and (4) a learning phase for allograph recognition. Here, we describe the design of these stages, related to the pattern recognition system proposed in this thesis.

4.1 Data Preprocessing

Prior to feature extraction, in order to reduce the noise introduced by the digitizing device, we need to preprocess the data. The preprocessing operations applied to our recognition system includes three steps: first *smoothing*, then *de-hooking*, and finally *point re-sampling*. Below, we define these operations and our approach to implementing them.

Slight shakes of the hand are the sources of the noise that a smoothing operation tries to remove. If each point on the main stroke is expressed as (x_i, y_i) in the Cartesian system, a trajectory of points is smoothed by a weighted averaging of each point and its immediate right and left neighbors:

$$(x_i, y_i) = 1/4(x_{i-1}, y_{i-1}) + 1/2(x_i, y_i) + 1/4(x_{i+1}, y_{i+1}). \quad (4.1)$$

Figure 4.1 depicts the Arabic letter “Dal” in its original recorded form and also in its refined form after applying these preprocessing operations. This smoothing effect

is illustrated in Figure 4.1(b). In online handwriting data, a hook-shaped noise often occurs at the start or at the end of a stroke. Figure 4.1(a) shows a starting hook. Hooks appear due to a digitizer device inaccuracy in the detection of a pen-down event, or due to a fast hand motion when positioning the pen on, or lifting it off the writing surface. Elimination of these noises is called de-hooking. Distinguishing a hook from the necessary hook-shape part of a character is not a straightforward task. In this thesis, for de-hooking, the head or tail of a stroke is considered as a hook if its length is short compared to the length of the entire stroke, and if the direction of the writing undergoes a sharp angular change. Figure 4.1(c) shows the result of the de-hooking process for the letter dal.

The (x, y) coordinates obtained from a digital tablet are originally equidistant in time. However, the distribution of points along a trajectory can be uneven due to variations in the writing speed (see Figure 4.1(a)). The re-sampling of the data produces points which are equidistant in space instead of time. This operation is used for either down-sampling or up-sampling, when fewer or more data points are desired, respectively. Figures 4.1(d) and 4.1(e) show examples of re-sampling. In order to satisfy the requirement of supervised classifiers for having feature vectors of equal lengths for all samples, we performed point re-sampling for all training and testing instances.

In our experiments, to achieve scale invariance (after smoothing, de-hooking, and point re-sampling), we normalized all characters to have the same heights, while their original aspect ratios remained unchanged. After preprocessing, patterns were

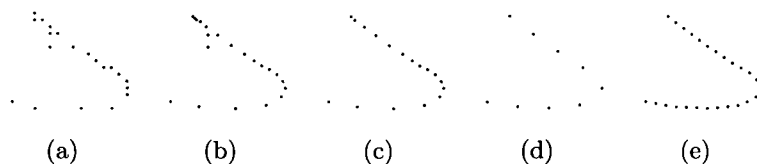


Figure 4.1: The Arabic letter *dal* in its (a) original form, and its preprocessed versions where the following operations were applied: (b)smoothing; (c) de-hooking; (d) down-sampling; and (e) up-sampling.

described for the classifier. In the next section, we present our method for the representation of online data.

4.2 Relational Histogram Representation

When dealing with online handwriting, features are mainly related to shape, length, and positions of parts of the strokes. Global features¹ in a character shape may provide more descriptive information about a character than local features. The representation proposed in this section is global and does not use local features, however, it does use a temporal ordering of observed points contained in online data.

We first demonstrate a technique that allows for spatio-temporal data representation. Then, we describe a more flexible contextual model that can be used to augment the visual realism of the shape of a character, by incorporating physical and geometric characteristics of the data. Our method for data representation treats both visual and dynamic data as contexts. We call this representation method *Relational Histogram (RH)* for quantifying the relationships between points of sequence data in the form of histogram counts. The *RH* representation builds on the shape context

¹See Section 2.6.3 for the definitions of global and local features.

idea [14]. Shape context is a descriptor intended for measuring similarities in the shape matching framework by using internal or external object contour information. Given that shapes are represented by a set of points sampled from their contours, a shape context descriptor finds the correspondence between the two objects by minimizing the computational cost of point-to-point matching between their contours. The shape context matching method is not applicable to sequence data, as the contour of the object is required in this matching method. Applying the shape context method to the sequence data which does not necessarily contain closed curves violates this contour assumption, and results in erroneous distances. Moreover, the point-to-point matching is computationally expensive. In applications like online handwriting recognition, all computations need to occur in real time. Therefore, the shape context method in its original form cannot be applied to online handwriting recognition. We try to eliminate this restriction in our *RH* definition.

To describe our technique, we introduce some notations. We denote them by:

- P , the set of online trajectory points of a character sample;
- S , the points of the normalized character's trajectory;
- Q , the set of reference points;
- r_{bin} , a set of bins for distances;
- θ_{bin} , a set of bins for angles; and
- V , the feature vector corresponding to a character.

The feature extraction related to *RH* representation is explained in Algorithm 1, which selects adequate relational features.

In the first part of this algorithm, we select an arbitrary and fixed set of points Q , which is equi-distanced from the normalized representation of the character, or from P . This set of reference points should not be located outside the minimum bounding rectangle of the symbol, in order to provide a more meaningful distribution of the symbol's characteristics. The Minimum Bounding Rectangle (MBR) is an expression of the maximum extent of a 2-dimensional object within its 2-D coordinate system. The points $q \in Q$ may capture some interrelationship structures. For instance, Q may contain symmetrical corners of the bounding box or the center of the bounding box. Let a character data be a set of points P made of x, y coordinates. The algorithm normalizes the size of the character and removes the duplicate points to obtain a set of points called S . The surrounding area of each reference point that falls inside the bounding box is divided into bins according to the distance and angle of each bin with respect to the reference point. The values of all the bins are initialized to zero for each reference point $q \in Q$. Then, all the points in S are described from the view of each reference point and are placed into the corresponding bins. This is done by computing the distance and angle between the pair of points, namely $dist(q, s)$ and angle (q, s) , and updating the corresponding bin that this pair can be mapped on. After this step, the number of points in the bins provided by all the reference points will give a compact representation of the character.

Using this system makes the descriptor more sensitive to differences in nearby points. Figure 4.2 shows the log-polar histogram bins for computing the features of

Algorithm 1 Feature Extraction of Relational Histogram Representation

INPUT: A set of normalized trajectory points S

OUTPUT: Feature vector V

Select an arbitrary set of reference points Q

Select an arbitrary set of r-bins: r_1, r_2, \dots, r_n and $\theta_{bins} : \theta_1, \theta_2, \dots, \theta_k$

for all $q \in Q$ **do**

 Initialize(r_{bins}, θ_{bins})

for all $s \in S$ **do**

 Compute $dist(q, s)$

 Compute $angle(q, s)$

 Assign($r_{bins}, \theta_{bins}, q, s$)

 Update(r_{bins}, θ_{bins})

end for

end for

$V = count(r_{bins}, \theta_{bins}, q)$

Return V

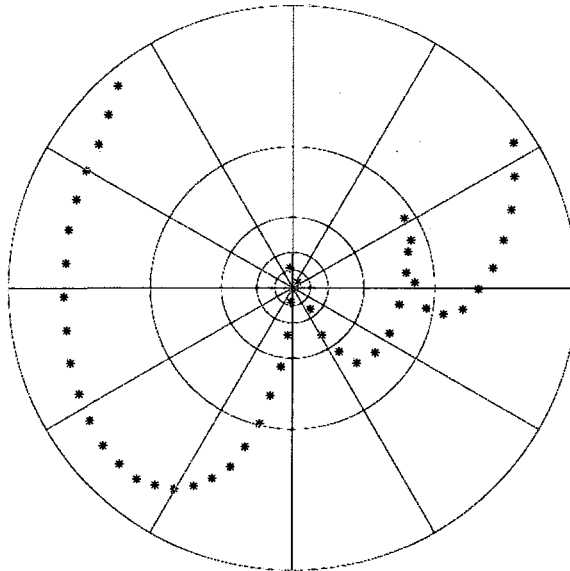


Figure 4.2: Diagram of the log-polar bins around the center of the bounding box of the Arabic character Seen. This diagram is used for relational histogram feature computation.

the Arabic character S , pronounced as “seen”. The center of the circle is located at the center of the bounding box of the normalized letter. The template for extracting the context features for the diagram has 5 bins in the tangential direction and 12 bins in the radial direction, yielding an RH feature vector of length 60. We could capture the global characteristics of the character by this feature vector. In our experiments with a variety of point sets, we noticed that the geometrical center of each character’s bounding box provides better recognition results, while it keeps the size of the feature vector more manageable. Therefore, we used this center point as a reference point in the log-polar coordinate system.

We used a directional feature as a complementary section of the RH feature to extract the local writing directions. The tangent along the character’s trajectory is calculated as follows:

$$\text{Arctan}((x_i - x_{i-1}) + j(y_i - y_{i-1})) \quad \text{for } i = 2 : N, \quad (4.2)$$

This tangent feature together with RH allows for spatio-temporal data representation and augments the visual realism of the shape of a character. Such contextual modeling has the advantage of creating a fixed size representation for a variable size of data. We used this representation in a recognition system, which is explained in the next section.

4.2.1 The Neural-Network-Based Recognition System

Artificial Neural Networks (ANNs) are, in principle, a family of nonparametric models that are intensively used to estimate probabilities (e.g., class-posterior probabilities).

Neural Networks (NNs) are able to learn complex nonlinear input-output relationships. These methods use sequential training procedures and adapt themselves to any kind of data. The standard neural network architecture for character recognition consists of adaptive forward connections and lateral inhibition, which provides competition between output neurons. Typically, a feed-forward NN with one hidden layer is mathematically expressed as:

$$n(x) = \sum_{i=1}^m c_i \sigma \left(\sum_{j=1}^s w_{ij} x_j + \theta_i \right), \quad x \in R^s, \quad s \geq 1, \quad (4.3)$$

where for any $1 \leq i \leq m$, $\theta_i \in R$ is the threshold, $w_i = (w_{i1}, w_{i2}, \dots, w_{is})^T$ is the connection weights of node i in the hidden layer with input nodes, $c_i \in R$ is the connection strength of node i with the output neuron, and σ is the activation function of the network.

A standard method of finding the weights is to use gradient descent optimization (back propagation method), where we calculate the derivatives of the cost function and find the weights in an iterative way. The classifier utilizes the feature vectors, and trains the network on these features by outputting all the resulting weights and the general network information. It then runs a full character set through the network, and outputs identification information for all the characters. A perceptron is the simplest type of NN, that takes the number of inputs and produces one output, which is a linear combination of all the inputs. If we threshold the output and interpret it as the predicted class, a perceptron can solve any linearly separable two-class classification problem. The true classification is obtained by adjusting the weights

during the training. Multi-class classification problems can be handled by a multi-output perceptron structure, which is defined as a number of parallel perceptrons equal to the number of classes, as the output. In order to cope with nonlinearly separable problems, additional layer(s) of neurons are placed between the input and output layers that lead to the MLP architecture. The MLP classifier is characterized by a number of hidden layers and output layer weights. Theoretically, every bounded continuous function can be approximated with arbitrary precision by a network with one sufficiently large hidden layer [41]. However, when the network grows, the training takes longer. Therefore, there may be an imposed constraint on the network size, in a practical application.

We exploited this classifier together with the relational features for online Arabic character recognition. We used feed forward ANNs as our learning classifier method for the RH feature representation. The relational histogram feature magnifies the differences between similar characters and improves learning in ANNs. The training problem for ANNs includes both the design of a network function and the fitting of that function to a set of input and output data points by computing a set of coefficient weights. The approach taken was to train Multi-Layer Perceptrons (MLPs), by using a conjugate gradient method for classification. This training algorithm for MLPs, which is commonly considered to be a robust and versatile method, optimizes the set of W weights and biases, w , so as to minimize an error function, E , when applied to a set of N training patterns.

Neural Networks are among the most popular methods used for classification of handwriting patterns. Feed-Forward Neural Networks and Self-Organizing Maps

(SOM/ Kohonen-Network) are the two most widely used NNs in the pattern recognition field for classification. A combination of two separately trained Kohonen Neural Networks in the voting scheme were used for recognizing online isolated Arabic characters [110]. While our recognition system uses the same learning method as in [110], our feature extraction method provides more descriptive feature vectors for learning NNs. Our experimental results in the next section empirically illustrate this strength.

4.3 Empirical Evaluations

Improvement of the recognition performance is considered as the most important quality for a feature representation. We have quantitatively analyzed the performance measures, defined as the accuracy and the timeliness of our recognition system. Before presenting the analysis, we describe the experimental setup and discuss the results.

4.3.1 Experimental Setup

Recognition experiments were conducted on the database developed in Chapter 3 and on an existing data set which is described in [113]. This data set consists of isolated Arabic letters provided by INRS.² In this database, the Arabic letters are divided into 17 classes according to their main shapes and each character has a large variety of samples. The data collection was done under no constraints, however, the number of participants was small. In the previously reported experiments with this data [110, 113], authors used 288 and 144 samples of each class for training and testing, respectively. As this database was not provided to us in a divided form of

²This database was provided as a courtesy of INRS-EMT Vision Group.

training and testing sets, we have used the following strategy to pick samples for training and testing:

In our experiment, we have used a 2 out of 3 ratio between the number of training and testing samples to make our results more comparable to those reported in previous studies on the same database. We randomly selected training and testing samples from the whole set of samples. By repeating this procedure 10 times, we created 10 pairs of training and testing sets. We conducted experiments for each pair of sets, and we reported the statistics of recognition results over ten independent runs. We aimed for more confidence in the reported recognition rate by following the experimental setup as explained. We trained an MLP classifier through a conjugate gradient method for classification using a three-layer network layout. In this study, we used the three-layer network with 100 nodes in the hidden layer.

4.3.2 Results

We present the results of using the RH feature compared to other methods for two databases: Mezghani’s database [113], and our developed database. In all of the experiments which are presented in this section, we computed cross-validation errors and reported the mean of the obtained recognition rate with a 95% confidence interval.

Results of Mezghani’s Database: Table 4.1 compares the performance of the RH feature to the previously used methods in the literature. The confusion matrix for a sample run is presented in Table 4.2. As the empirical results suggest, our method has shown a considerable improvement over the 1-NN on average, while for the previous study on the same database, the best recognition rate reported was

Table 4.1: Performance comparison of different features in the same experimental setup, which includes 33% training, 66% testing, and 6-fold cross validation. The reported figures for the recognition rate include the average over 10 independent such runs.

Method	Recognition Rate	Training Time	Test Time
1-NN Classifier (Euclidian distance)	R	0	8min and 6 sec
Kohonen Memory[113]	R - 1.19%	2 hours	26 sec
RH Feature	R +4.22%	7.5 min	23 sec

outperformed by 1-NN [113]. The recognition rate for different classes vary, however, on average, our recognition rate for different letters is about 5% higher than the method in [113] (Mezghani et al., 2005). Table 4.1 also compares the training and testing times on the same database with the same family of classifier. In addition to a faster training, the recognition time for each character sample using the NN-based system is 23s with our relational feature approach, while this was 26s for the experiments in [113].

Table 4.2: Confusion matrix for a sample run using the RH representation.

class label	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Recog rate%
1	143	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	99.31
2	0	137	0	0	0	1	1	0	0	1	0	0	0	4	0	0	0	95.14
3	0	0	144	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00
4	0	0	0	130	0	0	0	0	0	1	0	12	0	1	0	0	0	90.28
5	0	2	0	1	140	0	0	0	0	0	0	1	0	0	0	0	0	97.22
6	0	1	0	1	0	137	0	0	0	2	0	0	1	0	0	0	2	95.14
7	0	1	0	0	0	1	142	0	0	0	0	0	0	0	0	0	0	98.61
8	0	0	0	0	0	0	0	144	0	0	0	0	0	0	0	0	0	100.00
9	0	0	0	0	0	0	0	0	0	141	0	0	1	0	1	0	1	97.92
10	0	6	0	0	1	2	1	0	0	133	0	0	0	0	0	0	1	92.36
11	0	0	0	0	0	0	0	0	0	0	144	0	0	0	0	0	0	100.00
12	0	0	0	7	0	0	0	0	0	1	0	133	0	3	0	0	0	92.36
13	0	0	0	0	0	1	0	0	1	0	0	0	139	0	0	3	0	96.54
14	0	0	0	3	0	0	0	0	0	0	0	2	0	132	3	0	4	91.67
15	0	0	0	0	0	1	1	0	0	0	1	0	1	1	139	0	0	96.53
16	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	141	0	97.92
17	0	1	0	0	0	0	0	0	0	1	0	0	0	2	5	0	135	93.75
total rate																		96.16

Table 4.3: Performance of RH features using our database.

Method	Recognition Rate	Training Time	Testing Time
RH Feature	90%	6.3 min	25 sec

Results for the New Database:

We have conducted the experiment with RH feature using the second section of our database which consists of the main shape of the Arabic letters. The data has been prepared as two disjoint training and testing sets, including 4080 and 2023 samples, written by 101 and 49 users, respectively. We used the same topology for the NN classifier. The result is presented in Table 4.3. As this table shows, the recognition rate for the new database is lower. This is a result of more diverse data in this dataset.

4.4 Conclusion

In this chapter, we introduced the relational histogram feature representation for online handwriting recognition of isolated Arabic characters. We also investigated how the *RH* representation can be used by a supervised learning method such as ANNs. We showed the strength of this feature in Arabic character recognition of two different databases. Our experiments showed that this representation has gained a performance increase when compared to the best existing results in the literature. In the next Chapter, we will introduce another relational feature representation in combination with other supervised learning techniques for online handwriting recognition of Arabic characters.

Chapter 5

Relational Context Representation and an SVM-Based Recognition System

Chapter Outline

In this chapter, we propose another novel feature representation for online Arabic characters called relational context representation. This representation aims to capture the local relevance of any single point of a character's trajectory together with the neighborhood relationship between different parts of a character's shape. We present the corresponding feature extraction algorithm for this representation and incorporate this new relational feature with support vector machine classifiers. The characteristics of the two representations presented both in this chapter and in Chapter 4 are discussed in detail. Empirical results are also provided.

In defining the RH representation in Chapter 4, we mentioned that we only consider a few (or in fact only one in our experiments) reference points. These points can be independent of the character's trajectory points and only relate to the bounding

box of the character. It is not obvious to know how many reference points may provide the best descriptor of a character's shape. In this chapter, we have lifted up that dilemma and have designed *Relational Context* (RC) representation, which considers a fixed number of points selected from the character's trajectory. This new representation captures the relative pairwise distances and angles of the temporally ordered point sequence data. The idea is to use the dynamic and temporal information contained in online data to improve the recognition accuracy. The *RC* representation also encodes the contextual information of the sequence data. We use all pairs of points on the trajectory of a digital character to capture as much contextual information as possible about the shape of the character. This action preserves the local descriptors in the context of the global character shape. Each character point is observed from all the other points' views, and not from the viewpoint of independent locations from the character. Point to point analysis is an important step in many pattern recognition paradigms. For instance, in the field of Computer Vision, these analyses on the contours of objects are used to extract conceptual information. In such analyses, the curvature is a commonly used geometrical invariant to extract characteristic points. We explain how to extract the tangent-based curvature information, local, and neighborhood estimation of distances from the trajectory of data points in online characters. The resolution of the representation and the dimension of the related feature vector can be arbitrarily long. For instance, when there are some very long trajectories with many data points, then the feature vector can be potentially long.

The research in online handwriting recognition shows that appropriate preprocessing operations can increase the recognition accuracy by a few percentages. Therefore,

prior to the feature extraction phase, we need to do preprocessing operations as explained in Chapter 4. This includes removing the hooks of the strokes by using the changed-angle threshold with the length threshold, then filtering the noise by using a smoothing technique which is a moving average, and then normalization of the size to ensure that character shapes to be compared are of equal size. In the next section, we will describe the *RC* representation within the feature extraction phase.

5.1 Relational Context Representation

In this section, we describe the RC feature representation and explain why this feature may not be the best match with classifiers such as ANNs, in terms of required network complexity. However, for Support Vector Machine (SVM) classifiers, the size of the feature vector is not usually a practical constraint. Therefore, we deploy this feature representation in an SVM-based classifier with a suitable kernel function.

In this feature representation, local relevance is captured by the representation of any single component in the character's shape feature vector. Neighborhood components are considered for capturing the relationship between different parts of the character's shape. We have tried to achieve invariance with respect to different writing styles by preserving these relationships. In different writing styles, character points may have absolute information that is different from independent data points, however, the relative interrelationships on a set of points might not be different. Therefore, a small local change in the total shape of the character does not cause a large change in its representation. The global shape of the character is captured

through all pairwise relationships between any two components. To extract the RC feature, we have used Algorithm 2, described below.

Algorithm 2 Relational Context Feature Extraction

INPUT: A fixed number of sampled trajectory points S

OUTPUT: Feature vector V

$v = \{\}$

for all ordered data points $a \in S$ **do**

for all ordered data points $b \in S$ **do**

$L = \text{fit} - \text{line}(a, b, S)$

$d_{a,b} = \text{length}(L, a, b)$

$\delta_{a,b} = \text{tangent}(L, \text{x-axis})$

end for

end for

Add $d_{a,b}$ and $\delta_{a,b}$ to V

Return V

In this algorithm, in order to make the feature vectors of the same length as this is required for the application of supervised learning, we have re-sampled all characters so that they are represented by the same number of points. We have also preserved the order of the trajectory samples.

The relational property spanned by the RC makes it possible to include different levels of detail in the character's description. In other words, more details in larger features can be achieved by increasing the trajectory point density or fewer details in shorter features can be computed by decreasing the trajectory point density. The latter is particularly important in multi-stage classification schemes, in which a fast and rough classification is desired. The length of the *RC* feature for a trajectory of $|S| = N$ sample points is quadratic in N .

Figure 5.1 depicts some of these RC features for letter *Y*, pronounced as "ye" in

Arabic scripts. In this figure, the pairwise relationships are illustrated for point P_j . A complete set of these relationships for all pairs of points is considered as a feature representation for this character.

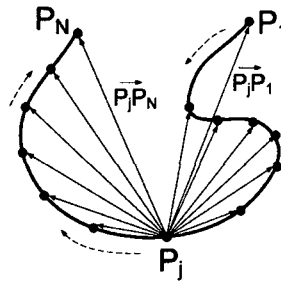


Figure 5.1: The Arabic letter “ye” in a relative context feature representation.

The recognition time is considered as a measure of time performance since the training process happens offline in a supervised classification. The separability level introduced by a feature representation can also have an impact on the training time. The time complexity is related to the properties of the feature representation and the type of classifier in general. Neural networks and SVMs are the most efficient and widely used discriminant supervised classifiers for handwriting recognition and are used in this research. Once the learning phase is completed, the length of the feature vector is not a concern for the execution time for two reasons. First, the number of data points that we need in order to describe a symbol is almost always very small. Second, the computation time for these features is a low-order polynomial. In the next section, we report on the recognition time with SVMs, which is considerably short and makes the whole system perform very fast.

5.2 The SVM-Based Recognition System

The use of Support Vector Machine (SVM) classifiers has gained increasing attention in recent years due to their excellent recognition results in various pattern recognition applications. The SVM as introduced by Vapnik [170], is a learning technique based on statistical learning theory. The idea behind SVMs for a binary classification problem is to find a separating hyperplane with the maximum margin. The maximum margin gives the smallest upper bound of the generalization error. For linearly separable classes, finding this hyperplane is a trivial task. If the data is represented as $\{(x_i, y_i)\}; i = 1 : N$ where $y_i \in \{+1, -1\}$, then the classifier can be expressed as follows:

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i < x_i, x > +b\right) \quad (5.1)$$

For non-linearly separable classification tasks, SVM maps the original space of input examples into a high-dimensional (possibly infinite-dimensional) space, where input examples become linearly or almost linearly separable. The separating hyperplane is then built into the new space. The non-linear mapping is performed by using kernel functions. Computing of the inner products of the vectors in the high dimensional feature space is avoided, and replaced by evaluation of a non-linear kernel between support vectors and vectors in the input space. In the case of non-linearly separable examples, the classification, or the optimal hyperplane solution can be found by finding the coefficients α_i that maximize the following objective function:

$$Q = \sum_{i=1}^N \alpha_i - 1/2 \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (5.2)$$

subject to the following constraints:

$$\sum_{i=1}^N \alpha_i y_i = 0, \quad (5.3)$$

$$0 \leq \alpha_i \leq C. \quad (5.4)$$

When it comes to applications, the choice of effective features is mandatory for boosting the performance of SVMs.

The RC representation potentially provides long feature vectors which might not be the best match with NNs in terms of the required network complexity for learning. However, for the case of SVMs, the size of the feature vector is not usually a practical constraint. In the next section, we demonstrate that the recognition time with SVMs is considerably short and that the whole system performs very fast. By using a decreased resolution of feature vectors in our representations, even NNs are capable of providing faster results than the ones previously reported in the literature on the same data set [82].

5.3 Qualitative Analysis for the Proposed Features

The strength of the new features presented in this thesis can be summarized qualitatively according to several important properties. Below, we describe some of these properties which include: invariance and uniqueness, flexibility, ease of implementation and memory issue.

5.3.1 Invariance and Uniqueness

If two characters have similar shapes, then it is useful to have a representation that maps them to similar feature vectors, and if they have different shapes then they should have highly distinctive feature vectors. The relational features we have proposed are able to capture all the interrelationships of a character's points. Preserving these relationships provides invariance with respect to similarity transformation, which could be decomposed into translation, scaling and rotation. A similarity transformation invariant is one kind of integral invariant. In particular, for our case we could set the integral domain to be a segment of a curve [155]. Supposing each point p_i has coordinates $(x_i, y_i) = (r_i \times \cos\Theta_i, r_i \sin\Theta_i)$, we can express the three invariant geometric primitives in the plane with polar coordinates. We keep $(\cos^2\Theta_i + \sin^2\Theta_i)$ which is always 1 in the formula for a quick transformation into xy coordinates: $R(i) = (x_i^2 + y_i^2)^{1/2} = r_i(\cos^2\Theta_i + \sin^2\Theta_i)^{1/2}$; $A(O, i, j) = 1/2(x_i y_j - x_j y_i) = 1/2 r_i r_j \sin(\Theta_j - \Theta_i)$; and $Dp(O, i, j) = (x_i, y_i)(x_j, y_j) = r_i r_j \cos(\Theta_j - \Theta_i)$. This similarity transformation can be interpreted as variations in writing styles.

In different writing styles, character points may have different absolute information of independent data points, however, the relative interrelationships on a set of points might not be different. Therefore, a small local change in the total shape of the character does not cause a large change in its representation. The global shape of the character is captured through all pairwise relationships between any two components. In our experience with two different data sets, our proposed representations, in particular the RC feature, provides very well separated clusters with high margins

in the feature space. In other words, the RC feature has the ability to incorporate a fair distribution of weights among all local and global features to be used in discriminating a character. Therefore, the total discriminability of the representation is robust against minor local deformations. We conjecture that the feature vectors combined by linear kernel SVMs are tolerant with respect to Gaussian noise.

5.3.2 Flexibility

The relational property spanned by all the features makes it possible to include different levels of detail in the character's description. In other words, more details in larger features can be achieved by increasing the trajectory point density or number of reference points, if desired. This also means that fewer details in shorter features can be computed by decreasing the trajectory point density. Such flexibility is particularly important in multi-stage classification schemes in which a fast and rough classification is desired.

5.3.3 Ease of Implementation

It is advantageous to choose a representation that requires easy programming and an implementation that needs a small amount of time. The algorithms we have proposed for feature extraction are easy to code and provide a low-order polynomial complexity in time and space.

5.3.4 Memory Issue

The memory issue is a side effect of feature representations. This issue has an impact on the classifiers due to the fact that the parameters of a trained classifier have to be stored in order to perform the recognition task in the execution time. These parameters include the weights and biases in the case of NNs, and they include support vectors and kernel parameters in the case of SVMs. The length of the feature vector in NNs can linearly change the size of parameters, and therefore the amount of memory needed. Whereas, in SVMs there is an additional effect on the number of support vectors induced by the separability property of the feature representation. This impact can change the size of SVM parameters in order of magnitude. The proposed features in this thesis are very attractive in this matter. More precisely, their high separability characteristic decreases the number of support vectors required and the amount of memory needed. As a consequence, the ultimate system will be very practical with respect to memory constraints.

5.4 Empirical Evaluation

In this section, quantitative analysis of our RC feature representation is presented with respect to recognition accuracy and time. In order to compare the relative performances of the proposed system to the state-of-the-art system performances, the SVM classifier was trained using the RC feature on the databases that were described in Chapter 3.

5.4.1 Experimental Setup

In each experiment, the training set was twice as large as the testing set, without any cleaning applied to both sets. The training and testing samples were disjoint while one writer may have contributed to both training and testing sets. In our multi-class character recognition experiments, we used *LibSVM* [35]. In order to extend SVM to be applied to a multi-class classification task, different extensions were used in the literature such as one-against-one [98], one-against-all [73], DAGSVM [136], and Divided-By-2 (DB2). We have used the “one-against-one” approach to deal with multi-class classification, in which $k(k - 1)/2$ classifiers are constructed, and each one trains data from two different classes. Also, some generalization errors are minimized by using k-fold cross-validation estimates. It was observed that C (the penalty parameter of the error term in the primal support vector machine optimization problem [170]) values between 60 and 100, for the linear SVM kernel, yielded a good character recognition rate. We have chosen different values of C for our training on different datasets.

5.4.2 Results

In Table 5.1, our RC and RH features in an SVM-based classifier for isolated Arabic characters are compared with previous research, in terms of accuracy. These results are compared with the best previously reported results in the literature, on the same database, by Mezghani et. al. [113].

Although our experimental conditions are more general, our method shows a superior performance compared to [113]. For the used database, our results show a very

low error rate, while the best recognition result in the literature has an error rate of 5.4% [113]. Although SVM allows training with a large number of features in a man-

Table 5.1: Comparative experiments on the database in [113] of Arabic letters.

Approach	Accuracy(%)
Kohonen Map [113]	94.6
RH-SVM (our method)	96.8
RC-SVM (our method)	99.0

ageable time, it has the drawback of memory size¹. SVM models require a relatively large memory in order to store support vectors which characterize the classification boundaries. The memory needed for each support vector depends on the dimension of the feature vectors. The flexibility of the RC representation in terms of size of the feature vector makes it adaptable to different memory constraints.

The recognition time in an SVM-based system also depends directly on the number of required support vectors. Therefore, the recognition time of our system (composed of an SVM classifier and the introduced RC feature) is fast, due to the reasonable size of the support vectors. While the recognition time for a character has been reported in [113] as 26s, it is remarkably decreased to 15ms using the SVM classifier and the relational context feature.

The empirical results confirm a considerable improvement in our method for speeding up the recognition process, and therefore proves the suitability of our proposed representations for real time applications.

¹With the advance of solid state electronics, memory is starting to be less of a limitation.

5.5 RC in Comparison with Existing Features for Online Handwriting Recognition

The experimental settings with respect to the dataset, the number of samples used in the training and testing sets, and the data preprocessing methods may not all be exactly the same among different researchers. Therefore, we decided to use identical settings to conduct some experiments in order to have a fair comparison of the expressional power of the proposed features with the previously existing features. We explain the details below.

Measuring the representational power of a feature by using a classifier as the evaluating function is a standard approach of feature selection methods used for different studies in the field of online handwriting recognition [108], [77], [171]. Here, we chose SVMs as the identical classifier. One of the reasons for this choice is the low expected probability of generalization error in SVMs. The other reason is the scalability of SVMs, meaning that the classification complexity does not depend on the dimension of the feature space in SVMs [88], [118]. Our kernel choice is a linear kernel. The SVM classifier with a linear kernel can provide us with measures that can be interpreted in two meaningful ways, described below.

The first interpretation is the direct relationship between the accuracy of recognition and the discriminational power of the features. More representational features can provide a better accuracy. The second interpretation, however, has to do with interclass distances. In geometry, when two sets of points in a two-dimensional plane can be completely separated by a single line, they are said to be *linearly separable*.

In general, two groups are linearly separable in an n -dimensional space, if they can be separated by an $n - 1$ dimensional hyperplane. A linear kernel partitions the Euclidian feature space by hyperplanes, thus it can identify the extent to which the data distribution in the feature space is linearly separable. In other words, more recognition error of hyperplane decision boundaries can be translated as less linear separability. Linear separability, on the other hand, is closely related to the concept of discriminative features. The whole purpose of extracting features and mapping the data to the feature space is to achieve easy class discrimination by increasing the interclass distance. Easy class discrimination is only provided by easily separable data, and separability is measured well by linear kernel SVMs. Therefore, through this framework, we can indirectly measure the class separability or interclass distances in a feature space by measuring the extent to which the recognition using the feature is erroneous.

Table 5.2: Comparison of different features tested on an Arabic dataset [113] and on our database, shown in the second and third columns, respectively.

feature	error%	
	Mezghani et al. database	our database
directional	2.7	9.9
positional	3.7	10.3
directional+positional	2.2	9.6
zone (3x3)	34.5	44.2
zone (10x10)	17.2	26.3
RC	1.0	8.0

In Table 5.2, we present the results of a feature comparison for Arabic characters.

We have chosen some of the most widely used features (noted in our presented survey in Section 2.6 on the employed features for online handwriting recognition) for this comparison. As Table 5.2 shows, the combination of the directional and positional features shows the best performance on all datasets. However, the error rate by using our RC feature is significantly lower than other features, given the same data, the same preprocessing methods, and the same classifier.

The results also show that the worst performances belong to zone features for all datasets. Such low performances might be explained by the fact that a small number of points represent each symbol. Even the more detailed zone features (10×10) did not result in a much higher performance, as we can see in Table 5.2.

Some of the misclassified samples when *RC* representation is used, are depicted in the Figures 5.2 and 5.3. Figures 5.2 shows some samples, their real labels, and the result of the classification. As shown in this figure, these samples are confusing, even for humans. The second figure shows samples of the Arabic letters *Waaw* misclassified as the joint class of *Faa-Qaaf*. These errors are mainly due to the high similarity of some characters. As we include more styles of writings in the data, the overlap between classes of characters increases. This is the reason that the recognition error is higher in our experiments on our database, regardless of the used feature. For some of these errors, using the complete shape of the character could help the recognition.

We have presented a novel feature representation (RC) for online Arabic character recognition. The feature extraction is easy to implement and provides a low-order polynomial in time and space. In SVMs, there is an additional effect on the number of support vectors induced by the separability property of a feature representation.







Arabic Letter main shape						
class label	Laam	Laam	Nuun	Nuun	Nuun	Nuun
classification result	Nuun	Nuun	Laam	Ha	Baa-Taa-Thaa	Baa-Taa-Thaa

Figure 5.2: Some misclassified Arabic letters from our database. The real labels are listed in the top row, and the results of classification are shown in the bottom row.

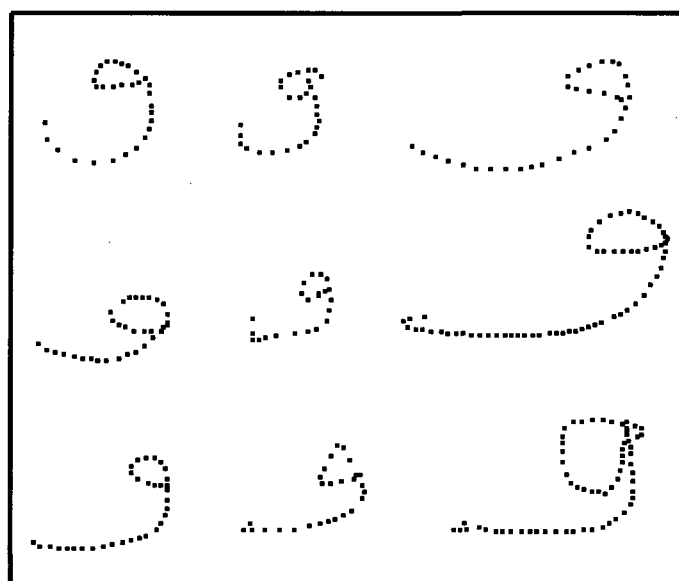


Figure 5.3: Nine instances of the Arabic letter waaw misclassified as faa-qaaf.

In our experience with RC, the number of support vectors required is small. As a consequence, the ultimate system will be practical with respect to memory constraints. We investigated the effectiveness of this feature empirically in our experiments. In all the experiments, the results have shown that at the character level, an SVM trained with the proposed feature performs significantly better when compared with the state-of-the-art features.

Chapter 6

Separability Analysis and Evaluation of Feature Representations

Chapter Outline

In Chapter 5, we have represented the effectiveness of our *RC* and *RH* features, measured by the performance of supervised learning classifiers. In order to further investigate the ability of these feature representations to provide the best description of the data, in this chapter, we use techniques from statistical analysis of the data in feature space and we use approaches normally used in cluster analysis. We use separability analysis as a means of quantitative evaluation of feature representations, independent of the choice of a supervised classifier. In the particular context of character data, we evaluate a feature representation by the level of separation that it introduces in the feature space for different character class instances.

The performance of a pattern representation is usually shown by the error of the

evaluating classifier. Our *RC* and *RH* features have demonstrated a stronger discriminatory ability in recognition systems when using SVM and ANN classifiers. In this chapter, we utilize another type of performance measure for feature representations, which is not tied to a particular classifier. Our goal is to investigate the extent to which the data we have at our disposal can be separated only by our choice of feature representation and by the choice of a metric for measuring the distance between data samples. A distance measure can be computed between the Probability Density Functions (PDFs) of samples from the same class or from different classes. In such an analysis, the intrinsic data characteristics are used in order to make a comparison between the separation ability of different features while using the same distance metric. These statistical measurements are directly applied to the feature space and include metrics such as Euclidian distance, Bhattacharyya coefficient, and Fisher's discriminant functions. Separability analysis may also provide some information about how easily the data may be correctly classified by using a feature representation.

In our data, we had 17 character classes and we tried to compare different types of feature representations in terms of their capabilities in separating the data samples in each character class. Ideally, we wanted a feature representation which would result in data points from one class to be close to each other and data samples from different classes to be placed widely apart. It should be noted that separability is not an absolute measure, and here we only used a relative measure of separability in order to make a comparison between different representations of feature descriptors. This is because there is no standard level of separation, and the data samples can be hard to separate, even by humans.

6.1 Separability Measure

In this section, we provide the required definitions in order to describe a single measure of separability performance for a feature representation. This measure must summarize the information about the distance of samples within each class, called *intra-class* distance, as well as information about the distances between samples from different classes, called *interclass* distance.

For the following equation, let us consider a feature space φ containing a sample set $S \subseteq \varphi$ that has been separated into N classes $\{C_1, \dots, C_N\}$, where class C_i contains N_i samples.

Definition: The intra-class distance $SW^\varphi(C_i)$ of class C_i is the average over all disjoint pairs of samples in C_i (i.e., within class distances) for each class C_i :

$$SW^\varphi(C_i) = \frac{1}{N_i(N_i - 1)} \sum_{\text{disj. } x_k, x_l \in C_i} D_m^\varphi(x_k, x_l), \quad (6.1)$$

where $D_m^\varphi(x_k, x_l)$ refers to the distance between two data samples x_k and x_l according to a metric m in φ .

Definition: For two classes C_i and C_j , the interclass distance $SB^\varphi(C_i, C_j)$ is defined as:

$$SB^\varphi(C_i, C_j) = \frac{1}{N_i N_j} \sum_{x_k \in C_i, x_l \in C_j} D(x_k, x_l), \quad (6.2)$$

Definition: The interclass distance $SB^\varphi(C_i)$ of class C_i is defined as:

$$SB^\varphi(C_i) = \frac{1}{N - 1} \sum_{j \neq i} SB^\varphi(C_i, C_j), \quad (6.3)$$

Definition: The separability measure of a class C_i in φ is defined as:

$$CS^\varphi(C_i) = \frac{SB^\varphi(C_i)}{SW^\varphi(C_i)}, \quad (6.4)$$

According to these definitions, the smaller the intra-class and/or the larger the interclass distances are, the larger the separability measure will be for a class C_i in feature space φ . Therefore, in order to compare two representations related to feature spaces φ and φ' in the separability that they provide for a class C_i , we say that the representation related to φ is better than the corresponding representation related to φ' if $CS^\varphi(C_i) > CS^{\varphi'}(C_i)$. Therefore, the feature representation which provides a better separability in most classes will provide an overall better performance. It should be noted that the metric m in these definitions should be the same for the two feature spaces that are being compared. This metric can be Euclidean, Bhattacharyya or any other alternative. The choice of metric can influence the separability distances. We discuss the details about different metrics and their pros and cons later on in the next section.

Another alternative statistic to the average measures used in the separability definition in the above equations is the maximum intra-class and minimum interclass distances, however, such measures will be affected considerably by the outliers' effect. Therefore, we have applied the above definitions in our experiments, which will be presented in Section 6.2.2.

6.2 Distance Metrics

There exists a variety of distance metrics which have been defined for measuring the distance between feature vectors. Some of these distances are only valid under the assumption that the data are presented by a feature representation that has a normal

or multivariate normal distribution. The use of the assumption that the data follows the multivariate normal distribution is common in pattern recognition and clustering, in general. However, it is required to determine if such an assumption is statistically valid, especially for data residing in high dimensions such as handwriting recognition data. Several statistical tests are available for multivariate normal distributions. We outline these three tests in the following subsections. If a distance proves to be a useful measure for class separability, then it may also be incorporated into the design of classification methods.

6.2.1 Test of Normality

Statistically, the normality testing is applied to test the null hypothesis that data samples X_1, X_2, \dots, X_n came from a normally distributed population. The concept of the covariance matrix is vital to understanding multivariate normal distributions, i.e., Gaussian distributions. For a pair of random variables X and Y , their covariance is defined as:

$$\Sigma[X, Y] = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y] \quad (6.5)$$

When working with multiple variables, the covariance matrix summarizes the covariances of all pairs of variables. In particular, the covariance matrix Σ , is the $n \times n$ matrix whose (i, j) th entry is $Cov[X_i, X_j]$. Under the assumption that the covariance matrix Σ is non-singular, the probability density function can be written as:

$$N_X(\mu, \Sigma) = \frac{1}{\sqrt{2\pi^d |\Sigma|}} \exp\left(\frac{-(x - \mu)^T \Sigma^{-1} (x - \mu)}{2}\right) \quad (6.6)$$

Thus μ is the mean value, Σ is the covariance matrix and $|\cdot|$ denotes the determinant function. Multivariate normality tests include the Cox-Small test [40], Smith and Jain's adaptation of the Friedman-Rafsky test [156], the Quiroz and Dudley test [139], and the Shapiro-Wilk test [153].

We have considered the widely used Shapiro-Wilk test as a typical normality test which was implemented in the statistical package *R*. We have also used this test in the experiments illustrated in this chapter, in order to test the normality of the data in the online Arabic character database that was presented in Chapter 3. To test a vector x against the normal distributions, the statistic for the Shapiro-Wilk test is defined as:

$$W = \frac{(\sum_{i=1}^n \alpha_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \mu)^2} \quad (6.7)$$

where μ is the sample mean, $x_{(i)}$ is the i -th smallest (order) statistic and the constants α_i 's are given by $(\alpha_1, \dots, \alpha_n) = \frac{m^T V^{-1}}{\sqrt{m^T V^{-1} V^{-1} m}}$. In this equation, $m = (m_1, \dots, m_n)^T$ and m_i 's are the expected values for the order statistics of independent and identically distributed (i.i.d.) random variables, sampled from the standard normal distribution, and V is the covariance matrix for those order statistics (the definition of V has been taken from Wikipedia). The null hypothesis for this test is that the data are normally distributed. It is possible to reject the null hypothesis if W is too small. If the chosen alpha level is 0.05 and the p-value is less than 0.05, then the null hypothesis is rejected. If the p-value is greater than 0.05, then the null hypothesis cannot be rejected. Quantile-Quantile plots (QQ-plots), which create graphical displays to test the distribution of data, are used to interpret the Shapiro-Wilk test results. In

performing QQ-plots, the data is ordered as $x(1) < \dots < x(n)$. Then, for $i = 1, \dots, n$ the ordered data $x(i)$ is plotted against i th equally spaced quantiles of i.i.d. sample points from a standard normal distribution. If the data set appears to be a sample from an approximate normal distribution, then the points will fall roughly along a diagonal line.

In this section, we have reviewed possible metrics for separability analysis, some of which require a normality test prior to implementation. This is due to the fact that these metrics are based on the computation of a covariance matrix, which is not valid for non-normal distributions.

6.2.2 Geometric-based measures

The most commonly chosen type of geometric-based measures is the Euclidean distance. It simply is the geometric distance in the multidimensional space. In order to place progressively greater weights on dissimilarities, the squared Euclidean distance is used alternatively, computed as: $d(x, y) = \sum_i (x_i - y_i)^2$. In some cases, the average difference across dimensions is used to quantify the similarity and dissimilarity between data samples in a feature space. This measure is called the Manhattan distance and is computed as: $d(x, y) = \sum_i |x_i - y_i|$.

The Minkowski distance as a general form of geometric-based metrics is defined by:

$$g_{ij}(q) = \left(\sum_{h=1}^N |x_i(h) - x_j(h)|^q \right)^{-q} \quad (6.8)$$

Note that when $q = 1$, it is the Absolute distance; when $q = 2$, it is the Euclidean distance; when $q \rightarrow \infty$, it is the Chebyshev distance. When the dissimilarity between

each feature vector has a great disparity, it is not reasonable to employ the Minkowski distance. In addition, the Minkowski distance has two shortcomings: it is relative to the fundamental dimension, and the correlativity between feature vectors is not considered. Since the mean and covariance of each class contribute to the recognition, the within-class information should be fully taken into account. However, the Minkowski distance has limitations for considering the within-class information, so other types of metrics which utilize the distribution characteristics of the samples in each class are more desirable.

We have considered the Euclidian distance from the group of geometric-based measures for comparing the between-class separability level provided by each feature representation in the experimental results. Our selection is based on certain advantages of this metric. For instance, the effect of outliers can be reduced since the distance between any two samples is not affected by the addition of a new sample to the analysis.

It should be noted that Euclidian distances can be greatly affected by differences in scale among the dimensions of the feature vector from which the distances are computed. For example, if one of the dimensions measures a characteristic which is an order of magnitude larger than the measures in other dimensions, then the resulting Euclidean distances computed from multiple dimensions can be biased by those dimensions which have a larger scale. However, this is not the case in our analysis with respect to features designed in this thesis and the ones designed previously by others, as the feature vectors have similar scales in all dimensions.

6.2.3 Information-Based Measures

It is possible to define distances based on information-theoretic measures, as the (asymptotic) KullbackLeibler (KL) divergence [100], Chernoff divergence [37] or Bhattacharyya divergence [19]. Mutual information is also an information-theoretic measure. However, it is not applicable in the same sense as the above measures since the mutual information of two random variables does not measure the similarity or dissimilarity of their probability densities. Instead, mutual information is a measure for the dependence of two random variables.

It has been shown that the KL divergence makes a poor estimate of the true divergence, known as Bayes error [101]. The most popular metric in this group, which provides an upper bound of the true error in special cases, is the Bhattacharyya distance. The Bhattacharyya distance is a measure of divergence from a known multivariate distribution. In statistics, the Bhattacharyya distance measures the similarity of two discrete probability distributions. In classification, this distance quantifies the between-class separability as a measure for the quality of features, regardless of the type of classifier. The reason is that Bhattacharyya coefficients are closely related to the average probability of classification error or the Bayes risk for the special case of equal priors [91]. The Bhattacharyya coefficients are used for estimation of upper and lower bounds on the probability of classification error for two [167], or more [57] classes. The Bhattacharyya distance between the two probability density functions' of classes i and j is formulated as follows, where class distributions are multivariate Gaussians $P_i = N(M_i, \Sigma_i)$:

$$D_{Bh} = \frac{1}{8}(M_i - M_j)^T \left[\frac{\Sigma_i + \Sigma_j}{2} \right]^{-1} (M_i - M_j) + \frac{1}{2} \ln \frac{|\Sigma_i + \Sigma_j|}{\sqrt{|\Sigma_i| |\Sigma_j|}} \quad (6.9)$$

The first and second terms in equation 6.9 are equal to zero when both classes have the same mean or covariance matrix, respectively. Therefore, depending on which term is dominant, classes can be separable by their means or their covariances. In case the first term dominates, a linear classifier is sufficient, and a more complex classifier is needed otherwise. The Bhattacharyya distance eliminates the disturbance of the correlativity between feature vectors and is not affected by the fundamental dimension. The Bhattacharyya distance utilizes the distribution characteristics of the samples in each class when the distribution of the set is a Gaussian distribution. A geometric interpretation of the Bhattacharyya distance, its relation to the Fisher measure of information, the statistical properties of the sample estimates, and explicit forms for various distributions are given in [48, 92]. Recent research in designing SVM kernels that take advantage of the Bhattacharyya distance has had successful results in different disciplines, for instance, for a language recognition system reported in [31]. Such kernels provide the potential possibility to exploit the information from the mean and also from the covariance.

6.3 Related Application Areas

Separability analysis has been widely studied and successfully applied in a large variety of application domains and many techniques based on this concept have been proposed in the literature. These techniques share the property that, given a fixed

metric, an automatic categorization of patterns can be accomplished by using a discriminant to partition the feature space. Separability can often benefit from efficient feature representation. Traditional classifiers such as Maximum Likelihood (ML) and Minimum Distance to Means (MDM) are examples of techniques which directly use separability analysis [99, 137]. SVMs with roots in statistical learning theory can be considered as one of these techniques. More precisely, SVMs operate by nonlinearly projecting the training data in the input space to a feature space of a higher (infinite) dimension by use of a kernel function. This results in linearly separable data that can be separated by a linear classifier. This process enables the classification of the data, which are usually nonlinearly separable in the original input space¹. In general, separability analysis has been used in two major areas of applications: feature selection and clustering, discussed in the following subsections.

6.3.1 Feature Selection

The feature selection problem, which is central to many data mining and pattern recognition applications, is the problem of mapping the original high-dimensional feature space into an optimum low-dimensional space based on certain criteria for error reduction of the mapping [149]. Linear and nonlinear functions have been introduced in previous research for the feature selection, however, there are two issues in the core of the feature selection problem: (1) to find an appropriate metric for

¹Classification in high dimension feature spaces may result in overfitting in the original input space, however, in the case of SVMs, overfitting is controlled through the principle of structural risk minimization, as explained in Chapter 5.

measuring the separability of data and (2) to design an appropriate function for mapping. Principal and canonical component transformations are two feature selection (or reduction) techniques for removing the redundancy in a feature representation. They are similar in that they both form a new n -dimensional set of data from a linear combination of the original n features [134]. These transformations are linear and can be expressed as an $n \times n$ transformation matrix. The principal component analysis is a special case of this transformation equation, which is optimal in the sense that the transformation matrix is chosen to be the one that diagonalizes the covariance matrix of X . The principal component features are therefore uncorrelated.

While the principal components transformation does not utilize any information about the class signatures, the canonical transformation maximizes the separability of the defined classes. Each class mean and covariance matrix must be specified for the transformation; the average within-class covariance matrix is calculated from the individual class covariance matrices and the between-class covariance matrix is calculated from the class mean vectors. A transformation matrix is then determined by simultaneously diagonalizing the between-class covariance matrix and transforming the average within-class covariance matrix into the identity matrix [177]. The goal is to maximize the separability between any two classes and to minimize the variance within the classes.

Discriminant analysis, which is used to determine which variables or features discriminate between two or more naturally occurring groups, can be also viewed as a feature selection problem. A specific type of discriminant analysis is called Linear Discriminant Analysis (LDA), which is used as a dimensionality reduction technique for

many classification problems. LDA tries to maximize the class separability criterion based on separability analysis.

6.3.2 Clustering

Data clustering, which is sometimes defined as finding partitions in data, is a popular area of data analysis and pattern recognition [121]. The difference between grouping provided by classification and clustering is that in clustering, only the structure of the data dictates the grouping (as in unsupervised learning), therefore, there is no labeling of the data (i.e., no predefined class labels). There are also no examples (i.e., of training data) that would show the kind of desirable relations among the data points [16]. Thus, the main concern in the clustering process is to reveal the organization of patterns into *sensible* groups. Since clustering algorithms define clusters that are not known a priori, the final partition of data into groups should allow the algorithm to find how many clusters exist in the data set. A clustering algorithm usually results in different partitionings of a data set, depending on the criterion used for defining similarity and dissimilarity between data samples. Therefore, the distance measure between data points is an important criterion used to establish the clustering rule or algorithm.

Cluster validity assessment is the process of evaluating the performance of a clustering algorithm. The general terms of “cluster validity methods” or “cluster validity measures” aim at quantitative evaluation of the results for the clustering algorithms in terms of two criteria: compactness and separability. To empirically assess this performance by comparing its output to a given correct clustering, one needs to define a

distance of the space for partitions of the data. The metrics defined in the previous sections can be used in this context.

Separability analysis in this thesis has been used in a special way, which is similar to and yet different from both feature selection and clustering. Our objective is to use the separability analysis to evaluate the relative performance of different representations when we already know the class label of each data sample and the total number of classes. The dataset we have worked with for this purpose is a set of instances from all classes, mapped into different feature spaces of different dimensions (depending on the corresponding feature representation). In this sense, the separability measure is similar to measures of cluster validation. The fundamental difference is that the true number of clusters is known and this number is the same for both representations.

6.4 Experimental Results

In this section, we will investigate how *RH* and *RC* feature representations perform with respect to the separability they provide in their feature spaces, when applied to the online Arabic character data. Our *RC* representation in the previous chapter has proven to be more useful than other existing representations when used with SVM classifiers. In the classification results, we have observed that the best performance of features after ours were Directional combined with Positional (DP). Here, we conduct experiment on the separability performance of these features.

In these experiments, we used the same number of data samples for all classes for the case of all representations. This is because the number of data samples in a

Table 6.1: Separability measures for three features on a dataset that includes 17 classes.

Class number	RH	DP	RC
C1	0.499	0.712	1
C2	0.136	0.143	0.244
C3	0.247	0.321	0.301
C4	0.153	0.156	0.268
C5	0.186	0.271	0.261
C6	0.152	0.128	0.226
C7	0.145	0.135	0.205
C8	0.157	0.160	0.231
C9	0.161	0.227	0.350
C10	0.138	0.144	0.204
C11	0.157	0.229	0.309
C12	0.172	0.159	0.305
C13	0.144	0.152	0.194
C14	0.224	0.196	0.277
C16	0.167	0.167	0.183
C17	0.155	0.216	0.248

class has an effect on the separability measures for that class in comparison to other classes. In particular, if a class has a large data population compared with other classes, it is usually better represented compared to the rest of the groups.

The separability has been measured in these experiments when operating on the Euclidian distance metric for parameter m in Equations 6.1-6.4. The results are demonstrated in Table 6.1 for RH , RC , and DP features. According to these results, RC introduces the highest separability measure in the majority of classes. Only the cases of class $C3$ and class $C5$, show that the DP feature provides the best separability. However, the results of DP for these two classes are not significantly different from the results of RC . Therefore, the best feature with respect to separability performance according to our definition is RC . The high recognition performance of the RC feature can be explained by the separability it provides between the data

samples in its feature space. However, in Figure 6.1, we clearly observe that there is no significant difference in the overall separability of the two features RH and DP for all classes.

The comparison of separability measures for different classes shows that class $C1$ has the best separability of all classes within every feature representation. This is consistent with our conclusion for recognition results among classes. In other words, the letter $Alif$ has the least similarity to all other letters, regardless of the feature used. It should be noted that the separability provided by the feature can only partially indicate the recognition performance, since decision boundaries for the recognition of each class is determined by the classifier at the end.

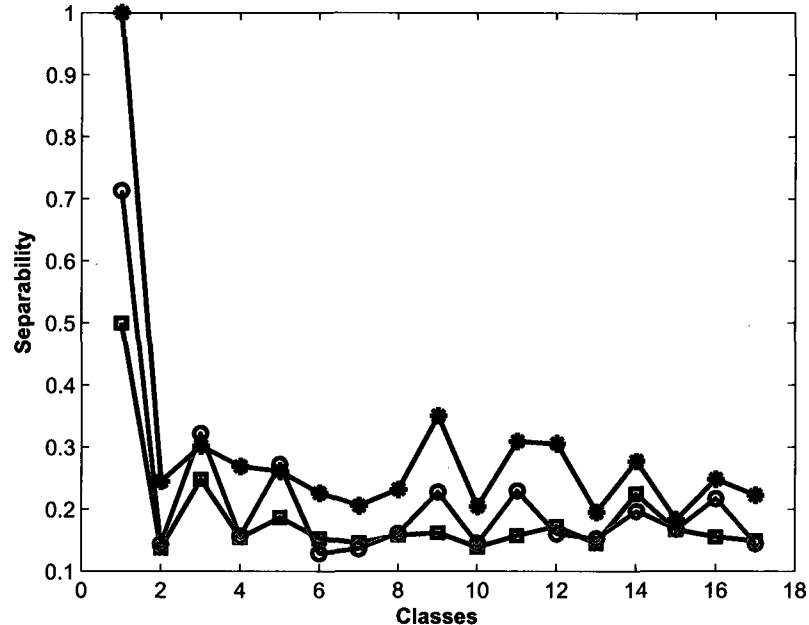


Figure 6.1: Comparison of the separability measure for classes $C1$ to $C17$ in three different feature spaces of RC , RH , and DP .

Showing multi-class data in a high dimensional feature space is not easy. In order to better visualize the point distribution of samples in the feature space, we tried to illustrate the point distribution of some samples from classes $C1$ and $C2$ for RC . For the ease of demonstration, only 30 random samples of each class were selected. The results are presented in Figure 6.2 and Figure 6.3 for $C1$ and $C2$, respectively. According to these results, there is a minor overlap between $C1$ and $C2$ for a few dimensions in the RC feature space. It should be noted that even on the overlapping axis, the scale is not the same.

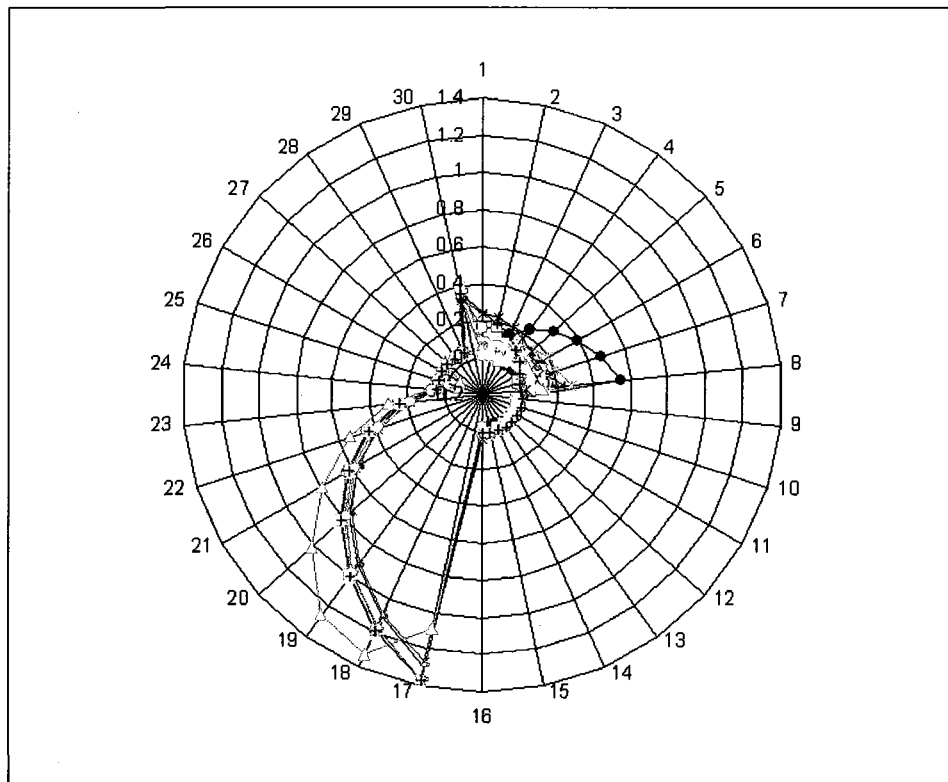


Figure 6.2: The scatter plot for some samples of class $C1$ in the RC feature space.

We also tried to analyze the separability of features with respect to metrics chosen

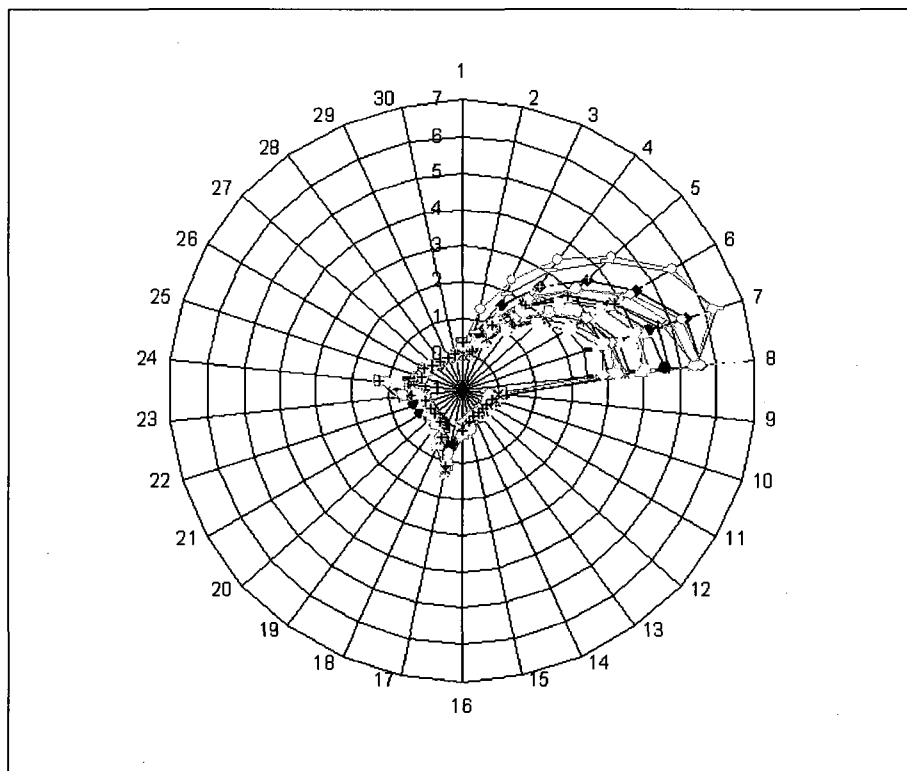


Figure 6.3: The scatter plot for some samples of class $C2$ in the RC feature space.

from the information-based measures, however, one requires the assumption that the data points present a multivariate normal distribution in order to be able to use these metrics. In order to perform the normality test for a multivariate distribution of samples, described by a feature representation in our database for each class, we used QQ-plots [85], which uses the Shapiro-Wilk test that was described in this chapter, to check on the similarity of the data to the multivariate normal distribution. The test data was compared to points following the multivariate normal distribution, generated with a mean vector, and compared to covariance matrix, estimated from the test data. Figure 6.4 shows the results of these experiments with QQ-plots for the *RC* feature. This figure shows that the distribution of data is far from normal in all classes of the database. In fact, neither of the feature representations previously designed nor designed in this thesis have resulted in multivariate normal distributions of data samples in their feature spaces. Therefore, none of the information-based measures could be used for separability analysis. The rest of the QQ-plots are shown in Appendix.

6.5 Conclusion

In this chapter, we have performed additional experiments using class separability analysis to evaluate a feature representation, independent of a classification method. We have discussed two groups of metrics for distance measures, to be used in separability analysis, namely geometric and information-based metrics. As our first criteria,

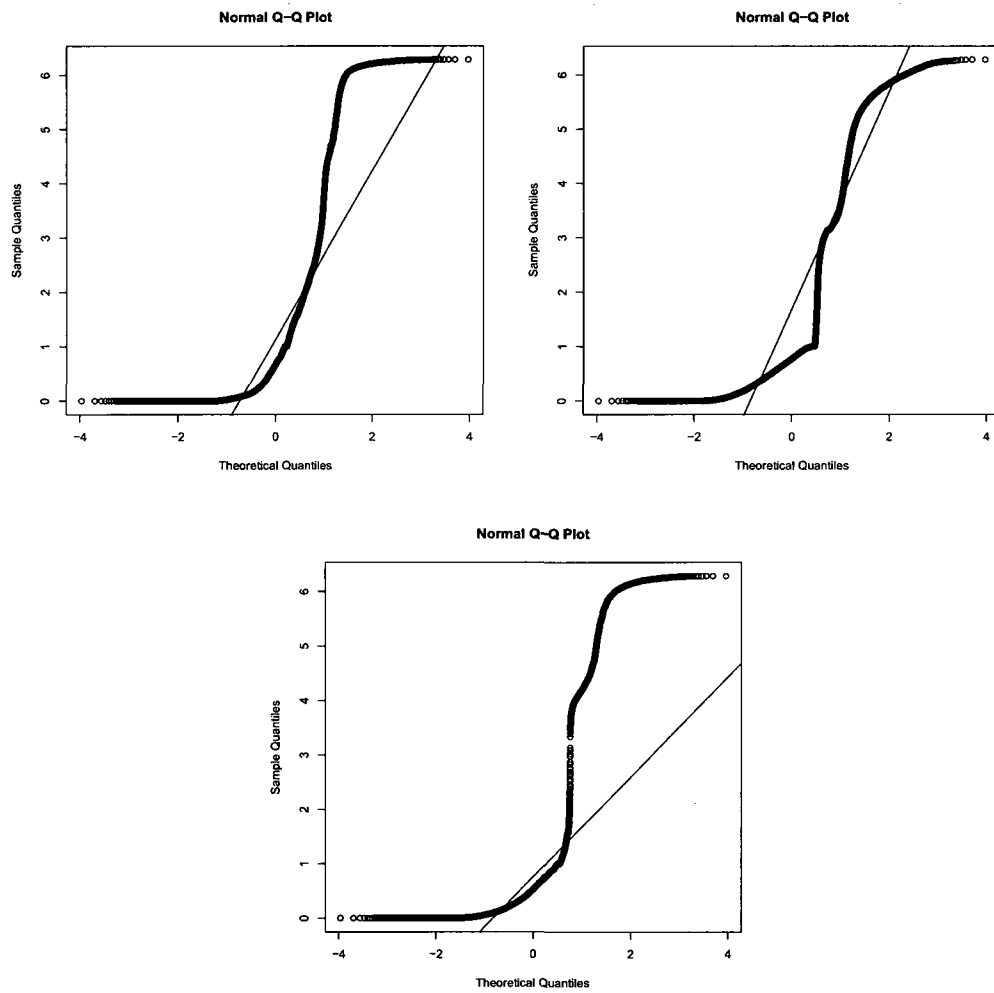


Figure 6.4: QQ-plots of some of the 17 classes of Arabic character shapes using directional feature representations.

the Euclidian distance was applied to a number of feature representations. This distance shows the amount of class separability that each feature set provides. Naturally, classes which are closer together in shape are more likely to have more confusion at the time of classification, and should therefore deserve more attention. Our findings and results suggest that there is a general agreement between separability and the classification accuracy. The feature representation with the highest values of separability according to the definitions provided in this chapter, is the one responsible for the highest classification accuracy. In addition, one of the best existing features with a lower value of separability also yields a good classification accuracy. In our experiments, we used the same size of sample sets for all classes. An interesting area for future research would be to explore how the sample size could affect the separability performance.

Chapter 7

A Hybrid System for Recognition of the Complete Shape in Arabic Letters

Chapter Outline

So far, we have demonstrated that *RC* features are powerful representations for the recognition of the main shapes in Arabic characters. However, due to the difficulties as reviewed in Chapter 2 for the recognition of complete Arabic characters, there exists only one approach in the literature that attempts to learn and recognize the complete shape in Arabic characters. In this chapter, we develop an innovative approach for classification of the complete character shape using *RC* features. We define three types of strokes at the complete character level based on the temporal data from a character's trajectory. We propose a new segmentation algorithm for extracting these strokes. A novel hybrid approach based on three SVM classifiers at the stroke level and a Bayesian network classification model at the whole character level is proposed. Our hybrid classifier is presented and evaluated using the database that was developed in Chapter 3. Experimental results of this evaluation are also provided in this chapter.

In Chapter 2, we discussed the main issues with the recognition of handwritten scripts such as Arabic/Persian, in detail. One of these issues, which brings a lot of confusion for recognition, is the fact that some Arabic characters are formed by more than one connected component. Such characters contain one main component and one or more secondary components. Due to this particular difficulty, except for one approach [160], all researchers in the literature on online handwriting recognition of Arabic characters have considered only the main shape of the characters. In [160], the complete form of Arabic letters are considered in both the constrained and non-constrained format, in a template-based recognition system. This work has been only conducted on a proprietary set of isolated Arabic single characters owned by the company Zi Decuma and the evaluation results are somewhat fragmentary. The author makes a note that the system cannot handle degraded shapes in characters and the recognition is restricted to the shape matching information made available by manually-produced templates. Although this task may be feasible when representing only a few symbols, representation and recognition of a large number of characters magnifies the limitations of this approach. Another, and more serious drawback with this approach, is that the accuracy in unconstrained character recognition is not robust and can vary from 40.9% to 85.4%, depending on the tuning of a large number of free parameters.

A number of methods in the literature on handwriting recognition have been explored for the recognition of multi-stroke gestures, symbols and characters in other languages [178, 78, 58]. These methods work either by joining the strokes to make a single stroke, by relying on the number of strokes to be an important distinguishing

parameter between various characters, or by assuming that the strokes are usually near-straight line segments (e.g., in Chinese language or in some characters). However, none of these methods can be efficiently applied in the case of Arabic letters. First, it is difficult to merge several strokes into a single distinguishing stroke because the Arabic character is not known in advance, and the relative position of strokes to each other can be highly variable. Second, the number of strokes in each single character is different due to handwriting variations and it highly depends on the variations in the stroke extraction process. The collected statistics on the variation in the number of strokes show this point, as presented in Chapter 3. Third, strokes in Arabic characters are cursive and different from line segments.

In this thesis, for the first time, we use a hybrid classifier for the recognition of complete shapes in online Arabic characters. We previously discussed hybrid classifiers in Chapter 2 and categorized them into abstract-level and measurement-level groups. The proposed approach fits into the measurement-level group, since it combines the class labels of different classifiers at the stroke level to make inferences about the posterior probability related to the final class of labels at the character level.

The most usual secondary components that form the complete shape of characters include one dot, two dots, three dots, and the hamzah, as shown in Figure 7.1. A dot is not necessarily a single point in the character's trajectory. In fact, most dots contain a number of points in the form of a stroke. The count and the position of dots relative to the main component form different characters. In different styles of Arabic writing, the appearance of the hamzah at a specific position may also form different characters. The hamzah may appear above the characters Waw and Ya,

above or below the character *Alif*, and inside the characters *Kaf* and *Ya*, or it may appear as a single isolated character. It is important to note that we only focus on the most popular style for writing the *hamzah*, which is consistent with characters in Persian handwriting, and that is to consider the *hamzah* only possibly inside of the character *Kaf*. This is the standard way of representing the 28-character set defined in the Arabic language. This standard was also respected for collecting data samples in our database in Chapter 3.

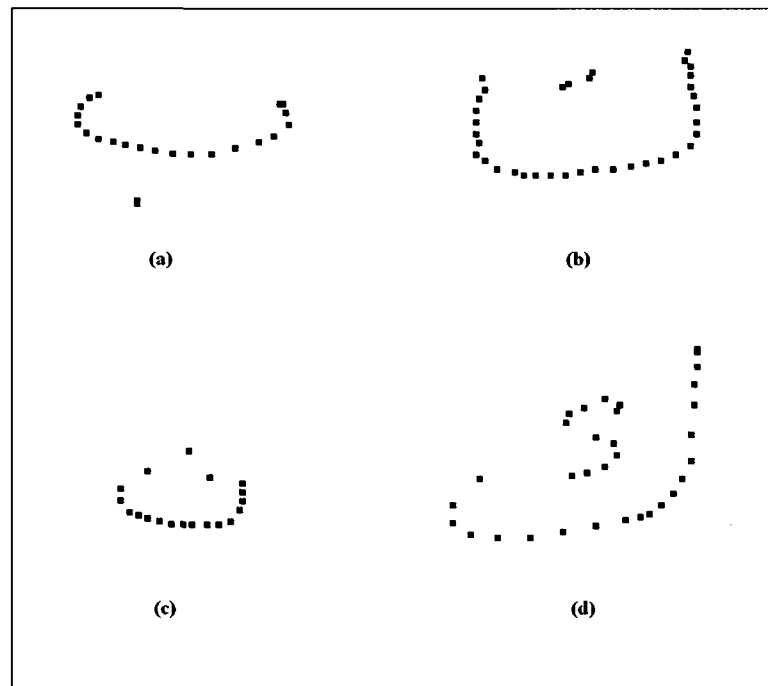


Figure 7.1: Different forms of secondary components: (a) one dot, (b) two dots, (c) three dots, and (d) *hamzah*.

In this chapter, we try to use the *RC* feature in the recognition of complete Arabic

character forms in a segmentation-based style. We will present the results of this approach in the experimental analysis of this chapter. The first step in the segmentation-based character recognition system is to extract features from the strokes that form the complete character shape in a space-time manner. This requires segmenting the characters into strokes. Segmentation is a crucial step of a recognition system as it extracts meaningful regions for analysis. In general, segmentation can potentially introduce serious problems in the development of cursive script systems by adding extra levels of complexity to the recognition problem. This is the reason why in those scripts, even in the case of word recognition, segmenting them into characters has often been avoided or made implicit (e.g., through a sliding window of a time-delay neural network or through a sequence of frames with Hidden Markov Models) [30, 86, 150] and [104, 154]. We present our segmentation method to extract strokes in the next section and then we present our hybrid approach for the learning and recognition of natural class labels.

7.1 Segmentation Method

There are a variety of segmentation methods that exist for online cursive scripts [6, 28, 32, 133, 129, 158]. However, no segmentation method is flawless. This is because producing reliable segmentation is a difficult problem. Our segmentation method is different from the methods previously proposed in the literature, in that it tries to separate the character's main shape from the complementary parts. Our approach is based on the definition of three different categories of pen strokes: (1)

main component character strokes (M), (2) secondary (or extra) component character strokes (E), and (3) hidden strokes (H). Figure 7.2 shows these three types of strokes for the Arabic letter Zay.

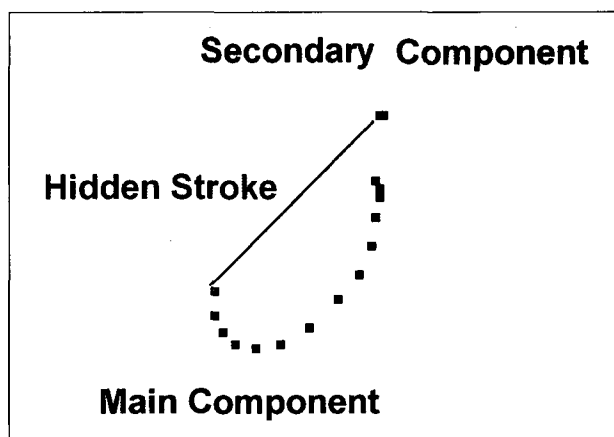


Figure 7.2: Segmentation of the Arabic letter Zay.

Main strokes play the leading role in recognition as they define the main shape or the main component of the character. The secondary and hidden strokes can provide extra support for the refinement in the classification of characters which share the same main shape but are different. Therefore, considering the *E* and *H* strokes may enhance the recognition performance to a certain degree. On the other hand, the variation in the styles of writing secondary components, particularly in the case of dots, increases the variation in patterns. The difficulty of recognition is in direct relationship with variation. The precise quantification of such a trade-off between support of the extra information for better recognition and increased difficulty of the recognition through added variation is an open problem. In the experiments reported

in this chapter, we try to empirically investigate this problem. Algorithm 3 presents our proposed character segmentation for the extraction of three types of strokes, M , E , and H .

Algorithm 3 MEH Character Segmentation in Online Arabic Data

INPUT: A set of sampled trajectory points of complete character data S

OUTPUT: Main, Hidden, and Extra strokes

$M = H = E = \{\}$

let $P_d = \{p_{d_1}, p_{d_2}, \dots, p_{d_k}\}$ represent the pen down points of S

let $P_u = \{p_{u_1}, p_{u_2}, \dots, p_{u_k}\}$ represent the pen up points of S

$M =$ longest p_{d_i} to p_{u_i} for $i \in 1, 2, \dots, k$

For all other points p_{d_j} and p_{u_j} in P_d and P_u :

if the distance between p_{d_j} to p_{u_j} is greater than ϵ_{max}

or less than ϵ_{min} connect them to M

$E =$ the rest of the trajectory after M

$H =$ from p_u^M to p_d^E

Return M, E, H

In this algorithm, by using all the pen-down (p_d) and pen-up (p_u) data, the longest stroke, which is the main stroke is determined. This is a reasonable thing to do since in natural writing, the main shape of the character is the largest part of it and the extra parts (*dots* and *hamzah*) come as considerably smaller portions of the character's shape. Noise in the input data can also appear in the form of extra strokes. It is also possible that the main shape of the character consists of more than one stroke. Therefore, connections between strokes need to be adjusted with respect to noise and the noise strokes should be separated from the character's base shape. Some artifacts, usually at the beginning and end of strokes and caused by immature hardware, have previously been the focus of preprocessing techniques [61].

Handling noise in the segmentation context is typically done by two main approaches: a fixed/dynamic template model [54, 150], and a fixed distance value for a noise segment [32]. We use a fixed distance between every pen-down and pen-up event in order to distinguish between three components: noise in the data (e.g., as a result of slight shakes); a stroke which is part of the main character shape; and a secondary stroke which is part of the secondary component of a character. This distance is described in our algorithm by using the threshold bounds ϵ_{max} and ϵ_{min} .

The secondary component of a character is determined after the main component is extracted. The rest of the strokes, if they exist (e.g., for more than one dot), are taken to form the stroke type E . We consider the distance from the pen-up information which defines the end of M , to the pen-down information which defines the beginning of E as the hidden stroke H . Unlike M and E , this stroke has no data point between its beginning and its end. Therefore, the recognition algorithm also incorporates directional features of the hidden stroke in its feature representation. It should be noted that the sets E and H can still remain empty at the end of the algorithm, depending on each character case. Ideally, this emptiness should correspond to the cases of characters which do not have secondary components including alif, haa, daal, raa, siin, saad, thaa, ayn, laam, miim, ha, waaw, yaa. The segmentation method presented here is simple and does not require lengthy numerical computations.

7.2 A Hybrid System for Recognition of Complete Online Arabic Characters

Our hybrid approach is based on multistage classification and the combination of two powerful classifiers. The idea is to first classify the main strokes into 17 classes representing the main shape of a character; the extra strokes are classified into five classes representing three different dot classes plus the *hamzah* and Null; and the hidden strokes are classified into eight classes representing the main possible geographical directions on a plane plus a Null class. These classification results are then combined in another stage by a Bayesian network classifier to learn and recognize the complete shapes of characters. The main idea of this approach is to learn the relationships between stroke types M , E , and H . Typically, this idea can be captured through a large number of rules to reason about each possible combination of different classes in M , E , and H . This is how rule-based systems, that are at the intersection of statistics and artificial intelligence, handle probabilistic reasoning. There are many differences and similarities between rule-based systems and Bayesian networks for reasoning under uncertainty [131]. However, the problem with rule-based methods is that it is impossible in most cases to design an exhaustive set of rules that model all possible ways of forming a complete set of relationships between different effective features. This results in the less efficient development of a rule-based system than its corresponding Bayesian network. For instance, a rule-based system would be less efficient in a case like our hybrid system in which we have to span all possible relationships between 17 classes of M and 5 classes of E plus 9 classes of H . On the other hand,

the designer of a rule-based system has more control over the ultimate classification behaviour of the system, whereas in Bayesian networks, the behaviour of the system is entirely based on the data used to learn and test the model. When used in conjunction with statistical techniques, Bayesian network methodology is more advantageous than rule-based techniques for data analysis. For these reasons, we preferred using a Bayesian network model to a corresponding set of rules in the case of our hybrid system (Figure 7.3) for probabilistic reasoning on the joint classification results of M , E , and H .

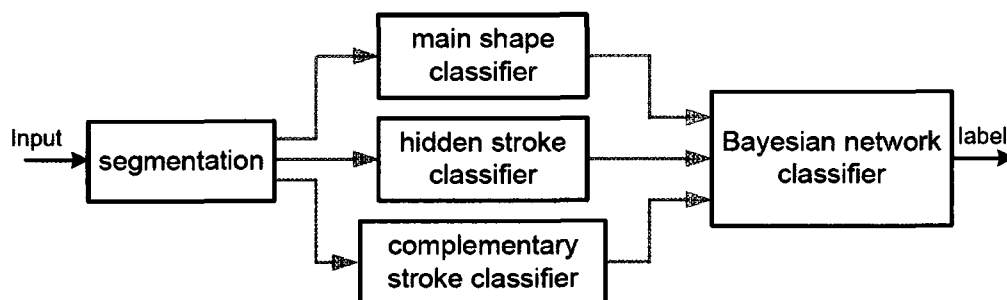


Figure 7.3: The block diagram of our BN hybrid classification approach

As illustrated in Figure 7.3, our proposed hybrid system for recognition of complete shapes in online Arabic characters uses the segmentation method described in the previous section to divide the character trajectory into three stroke types, M , E , and H . In the next step, we use three SVM classifiers to recognize the strokes in each of their corresponding classes. We use RC features to describe the strokes of types M and E , and directional features to describe the hidden stroke H . A Bayesian network classifier uses Bayesian inferential analysis to combine the recognition results of these

three SVMs in order to determine the appropriate Arabic character in a 28-class classification. In the next section, we provide a brief overview of Bayesian networks and their inference methodologies.

7.3 Bayesian Networks

Bayesian Networks (BNs) have emerged in the last decade as one of the most important technologies in the field of machine learning and artificial intelligence. They have been used for reasoning under uncertainty and have also been considered as a classification tool. A Bayesian network is a graphical representation of relationships between random variables X_1, X_2, \dots, X_n based on probability theory and graph theory. A Bayesian network consists of a Directed Acyclic Graph (DAG) in which every node V_i represents a variable X_i , and every edge represents a conditional dependency between variables. The edges are directed and define a parent-child relationship over the network. The nodes are parameterized with probability distributions that quantify the conditional probability relationships among connected nodes. The probability of an arbitrary event $X = (x_1, \dots, x_n)$ can be computed as $P(X) = \prod_i P(x_i | \text{parent}(x_i))$. In general, encoding the joint distribution of a set of n discrete variables requires a space exponential in n . Bayesian networks reduce this encoding to a space exponential in $\max_{i=1, \dots, n} |\text{parent}(x_i)|$. Because of the graph-based nature of BNs and their ability to describe uncertainty and complex relationships between all variables in a compact manner, they provide an efficient general framework for reasoning with almost any type of data. A joint probability table has an exponential size in the number of

discrete random variables; however, BNs represent the joint probabilities compactly, based on the conditional independence relationships between variables presented by the edges of the graph. In addition, BNs can simultaneously provide more accurate and flexible predictions on more than one node. A BN encodes a unique joint probability distribution over all the nodes, which can be computed by using the chain rule. The posterior probabilities are defined by the Bayes rule: $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$, where $P(X|Y)$ is the conditional probability of X given Y (i.e. likelihood), and $P(X)$ and $P(Y)$ are the probabilities of X and Y , respectively (i.e. priors).

Building a BN consists of determining its network structure and its set of parameters. The structure encodes a local Markov assumption: a variable is conditionally independent of its non-descendants in the network, given the value of its parents. The parameters describe how each variable relates probabilistically to its parents through Conditional Probability Distributions (CPDs). Automatically, learning the structure of a Bayesian network from data is a well-researched but computationally difficult problem [42, 135, 127, 144]. A function is used to score a network with respect to the training data, and a search method is used to look for the network with the best score. Different scoring metrics and search methods have been proposed in the literature [144]. The scoring functions used to select models are based on the likelihood function of a model given the data or the logarithm of this function. Since the associated search space is exponentially large, local search-based approaches are usually used to find the best network. These approaches iteratively consider local changes (adding, deleting, and reversing an edge) to the network structure. This type of search is very useful when dealing with large data sets like ours because of its

computational efficiency.

There are different algorithms for learning the parameters of BNs [135, 87]. The Expectation Maximization (EM) algorithm [44] is usually used to find a locally optimal maximum likelihood estimate of the parameters. The basic idea behind the EM algorithm is that if we knew the conditional probability tables of all the nodes, learning (the M step) would be easy. So in the E step, we compute the expected values of all the CPTs by using an inference algorithm, and then treat those expected values as though they were observed (distributions). Next, we maximize the parameters, and then re-compute the expected values, and the procedure is repeated. This iterative procedure is guaranteed to converge to a local maximum of the likelihood surface and it produces the CPTs for all the nodes. In a BN, the conditional or posterior probability of a variable, given information about the values of other variables, can be computed by a technique called message-passing. The intuition is that while all nodes of the network are correlated, distant nodes are less correlated. Therefore, the network is decomposed into highly connected parts (cliques), and then inference is performed in each part to a high level of accuracy, where only aggregate information is exchanged between parts. If there are no missing values in the data, a gradient descent algorithm [135, 87] is used as an alternative to the EM algorithm. The gradient descent learning algorithm searches the space of BN parameters by using the negative log likelihood as the objective function it is trying to minimize. Given a BN, it can find a better one by using BN inference to calculate the direction of the steepest gradient to know how to change the parameters (i.e. CPTs) to go in the steepest direction of the gradient (i.e. maximum improvement) [97, 122]. Another

alternative for parameter learning in BNs is counting, which is by far the simplest method. However, it can be used only whenever there are no latent variables, and there is not much missing data or uncertain findings for the learning nodes or their parents.

Probabilistic inference is one of the most common tasks to be solved by using BNs. After a BN has been constructed and learned, inference is performed by entering findings (also known as evidence). These findings are the values for the known nodes and they use a belief updating procedure such as belief propagation [135, 87] to determine new probabilities for the states of all the other nodes. Probabilistic inference results in a set of stochastic beliefs at each node. The inference problem in BNs is generally a computationally challenging problem. However, different inference algorithms are known for BNs such as variable elimination, likelihood weighting, and Gibbs sampling. These algorithms are explained in detail in [135, 87, 144] and they can be applied efficiently to many practical problems. In the next section, we will describe how we have constructed a Bayesian network in the context of online Arabic character recognition and in combination with other classifiers.

7.4 The BN Hybrid Classification

This is the first time that Bayesian networks are used for the recognition problem in online handwritten Arabic data. In the hybrid system proposed in this chapter, we will use a BN to enable inferences about the combination of classifications across different classifiers with different types of strokes. In practical terms, the network will

help answer questions about the overall classification into complete character shapes, given the partial knowledge about the classification results from each stroke.

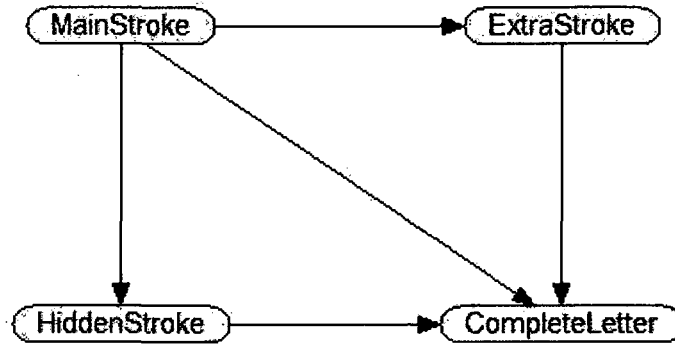


Figure 7.4: Bayesian network used in our hybrid system for recognition of complete Arabic letters.

In a classification task with our hybrid approach, we use a Bayesian network B , as presented in Figure 7.4. The network encodes three nodes for three types of strokes, namely, M , E , and H . At the core of each node there is an SVM classifier that provides the classification results for M , E , and H corresponding to each data sample. The main task of the BN is figuring out how to generalize the relationships between results of these SVMs as exhibited in the training data, for finding the final classification of the test data. The structure of the network is based on the fact that both E and H are probabilistically dependent on the shape M . However, E and H are conditionally independent given M . The final class label of a character is dependent on all other three nodes. With this structure used as the DAG for the network, we will use the gradient descent method to learn the distribution $P_B(M, E, H, C)$ and the parameters of the network (i.e. CPTs) from a given training set. We can then use

the resulting model for making inferences. We use the fastest known algorithm for making the exact general probabilistic inference in BNs, known as message passing [87]. The intuition is that while all nodes of the network are correlated, distant nodes are less correlated. Therefore, the network is decomposed into highly connected parts (cliques), then the inference is performed in each part to a high accuracy, and only the aggregate information is exchanged between parts. To use this method, we need to represent the network as a graph by using a complex set of data structures that are connected to the network. This graph is called a junction tree of cliques.

The Bayesian network will enable inferences about the distribution of the final class labels related to the complete shape of the character, given the distributions over classes of the main shape, classes of the hidden stroke type, and classes of the secondary stroke type. In a high-level description, this means that given a set of attributes (m, e, h) , the classifier, based on B , returns the label c that maximizes the posterior probability $P_B(c|m, e, h)$. Quantification of the relationships between these three distributions would require an exponential set of rules if we were to use a rule-based system. Therefore, in this context, BNs are efficient tools for learning and generalizing possible rules between these three distributions, resulting in an appropriate class label. We empirically evaluate the hybrid system through our experiments presented in the next section.

Table 7.1: The recognition results for multi-stroke Arabic letters in 28 classes by using a hybrid classification system.

Class	Recog. (%)	Class	Recog. (%)
C1	98.96	C15	96.72
C2	95.78	C16	90.02
C3	78.55	C17	93.91
C4	77.39	C18	92.27
C5	86.96	C19	91.20
C6	91.94	C20	75.65
C7	95.65	C21	76.52
C8	86.16	C22	86.66
C9	86.09	C23	94.81
C10	87.96	C24	94.64
C11	88.13	C25	92.17
C12	94.78	C26	92.36
C13	96.52	C27	93.91
C14	91.30	C28	94.15
Average		90.04	

7.5 Experimental Results

To evaluate our system, we performed several tests on the database that was described in Chapter 3. These tests are unlike previous efforts, in which experiments were carried over small sets of data collected from a limited number of writers and implemented and evaluated by a recognizer. Our goal was to evaluate how the new hybrid technique deals with a large database of unrestricted online handwriting patterns. For evaluating the recognition rate, we performed multiple runs of 3-fold cross validation. In each run, the database was divided into two distinct sets : a training set of 6720 samples and a testing set of 3360 samples (which correspond to 240 samples of each character for training and 120 for testing). Table 7.1 summarizes these results.

The average of overall recognition of all classes by using the hybrid approach, as presented in Table 7.1, was 90.04%. Unfortunately, recognition of the complete shape in online Arabic characters is an unexplored area of research. For this reason, we were not able to compare our results with available results in the literature. As mentioned earlier, the only study related to this topic [160] used a private database and therefore the reported results were not clear. For example, it is not clear if the author performed the experiments on the full set of the Arabic alphabet or only on some part of it. Nonetheless, the accuracy in unconstrained character recognition varied from 40.9% to 85.4%, depending on the tuning of a large number of free parameters. Furthermore, the author did not mention if the reported rates were based on the best possible rates, best character rates, or averages over all the characters. Therefore, our results can be considered as an advance to the field.

We observed a range of recognition rates that fell between 75.64% and 98.96%. Low recognition rates belonged to classes *taa*, *thaa*, *faa*, and *qaaf*. This is not so surprising, as the recognition between the two letters *taa* and *thaa* and also *faa*, and *qaaf* is very difficult, even for humans, due to the natural similarity between these characters. The use of our hybrid approach at the final classification level increased the recognition ability for the class of characters which were similar to each other in the main shape but which possessed different hidden strokes or secondary strokes. However, we observed that there were cases for which this approach decreased the recognition power for the complete character shape compared to recognizing only the main shape. In demonstrating the results, we categorized these cases of errors and they will be discussed in more details in the following section.

To illustrate how the hybrid system is used to find the final class label of a complete character, we will present inferential results on some of the character samples from our database in Figures 7.5, 7.6, 7.7, and 7.8.

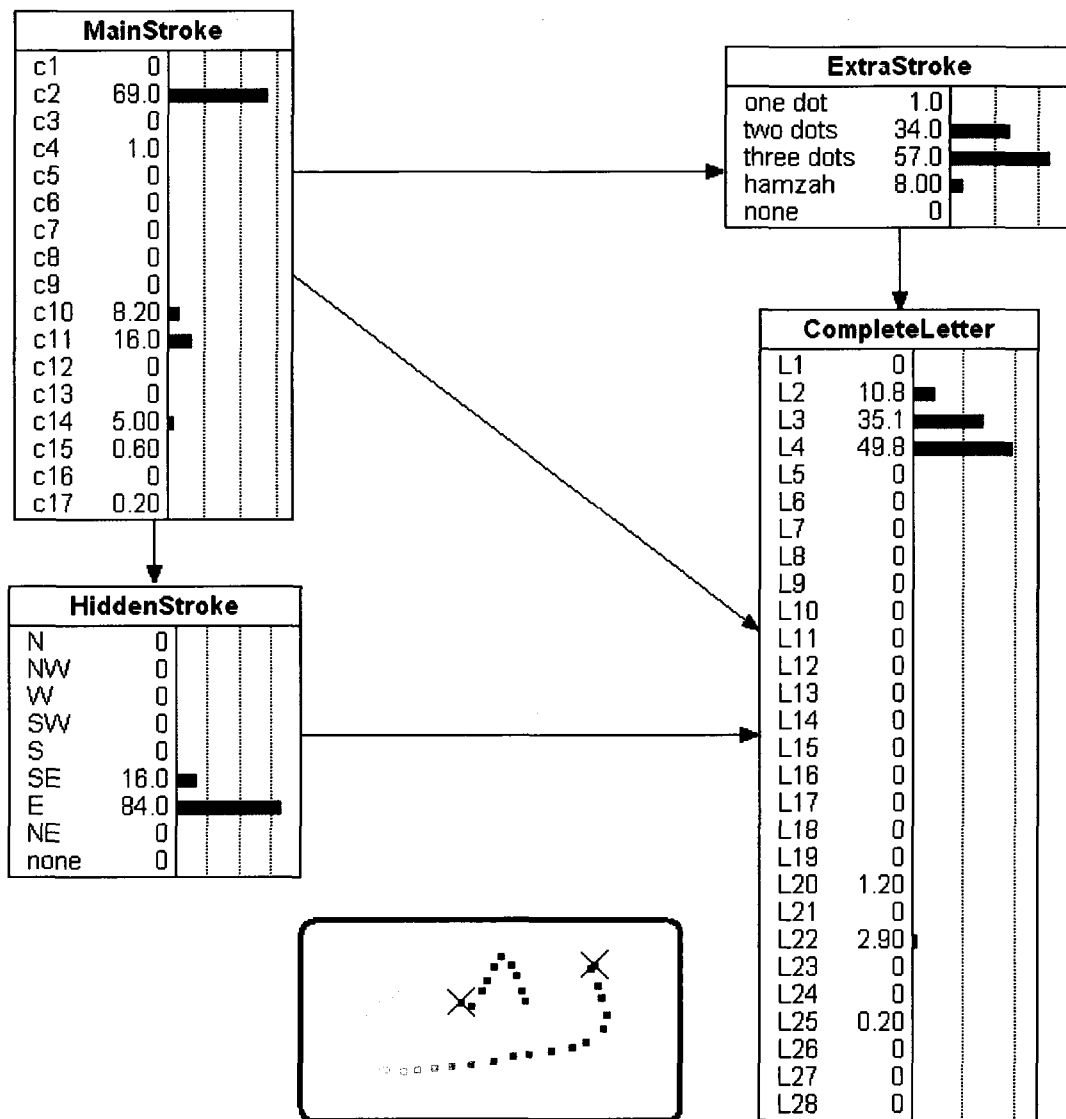


Figure 7.5: The probabilities for a sample of the Arabic letter taa in our BN hybrid system.

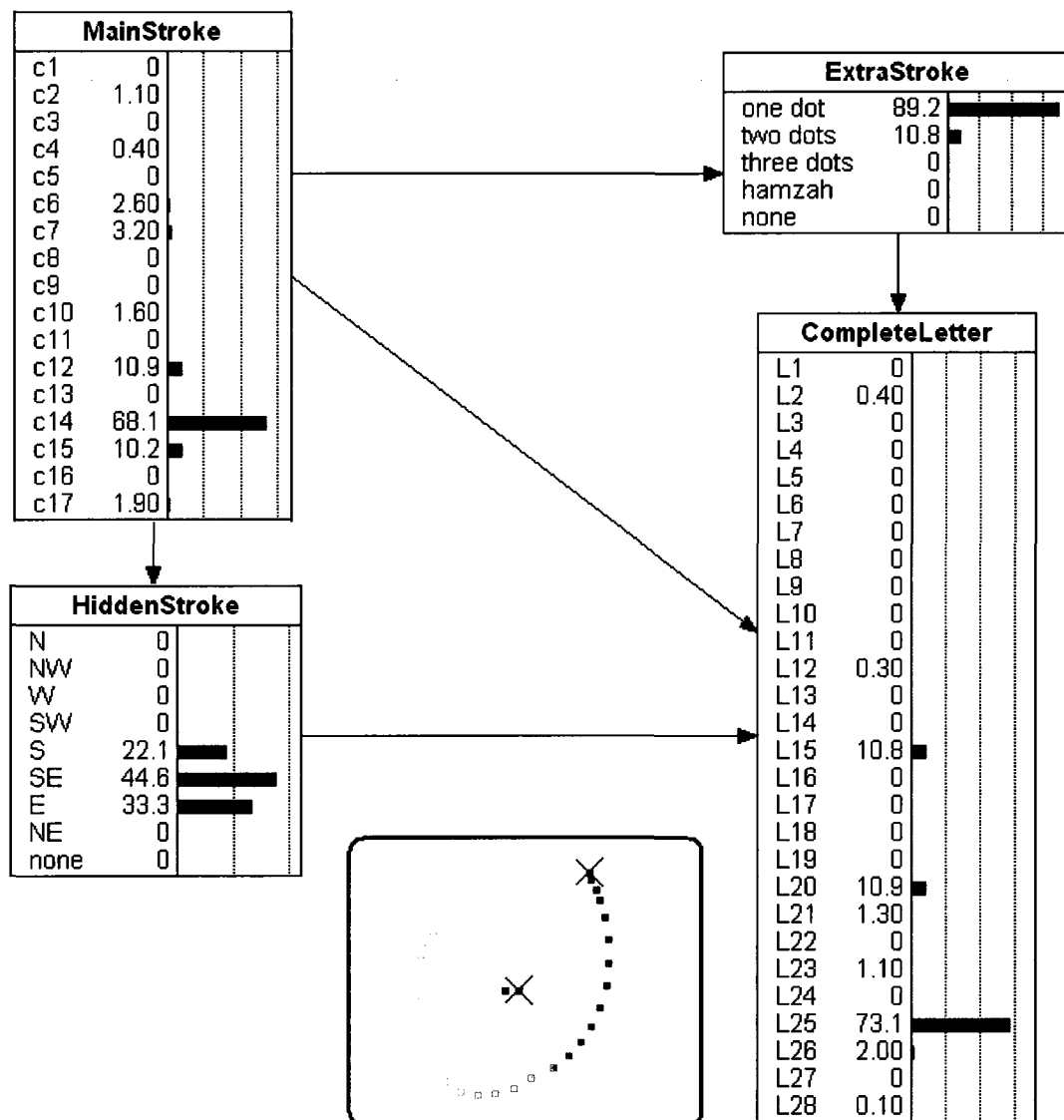


Figure 7.6: The probabilities for a sample of the Arabic letter nuun in our BN hybrid system.

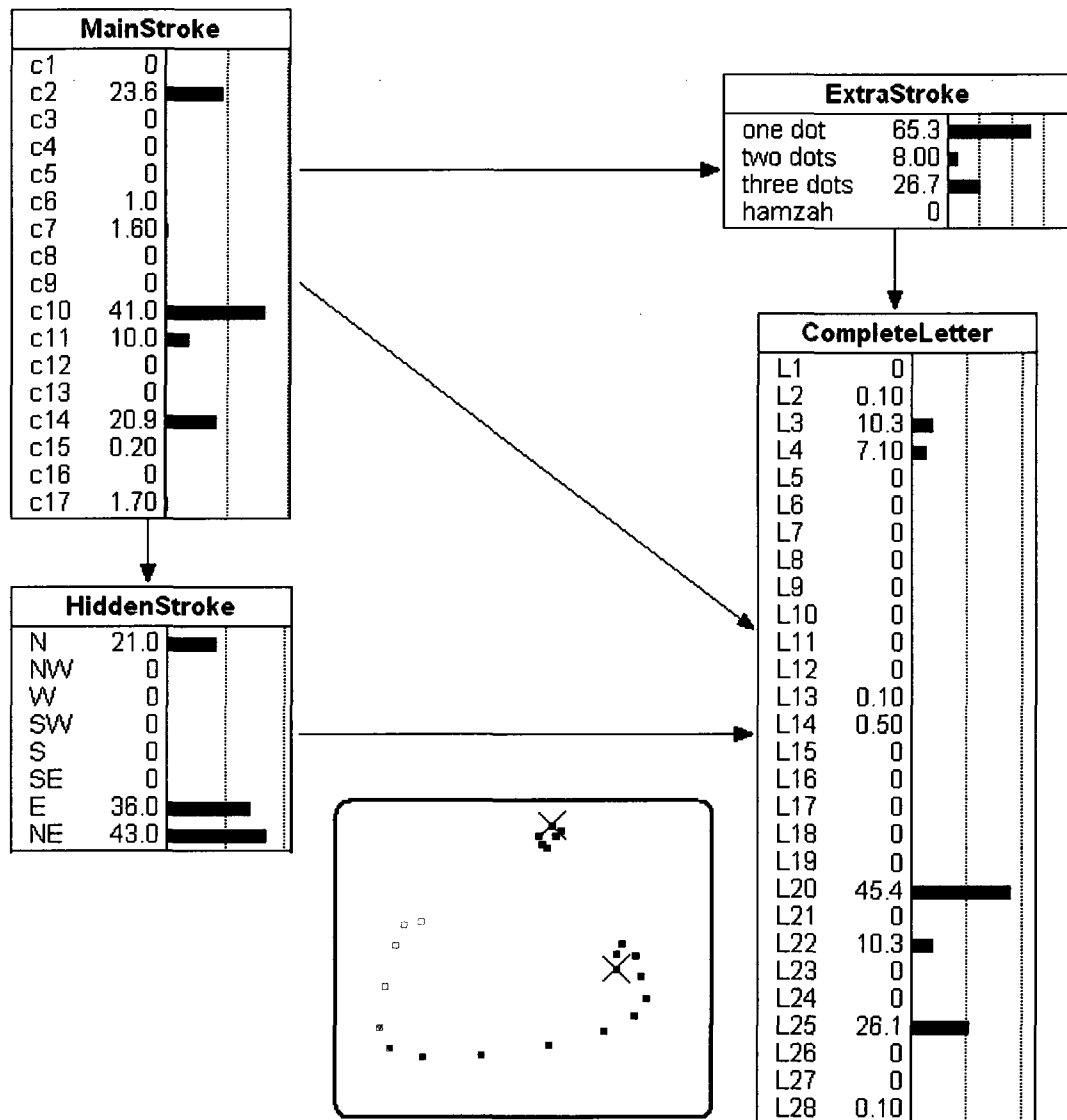


Figure 7.7: The probabilities for a sample of the Arabic letter nuun in our BN hybrid system.

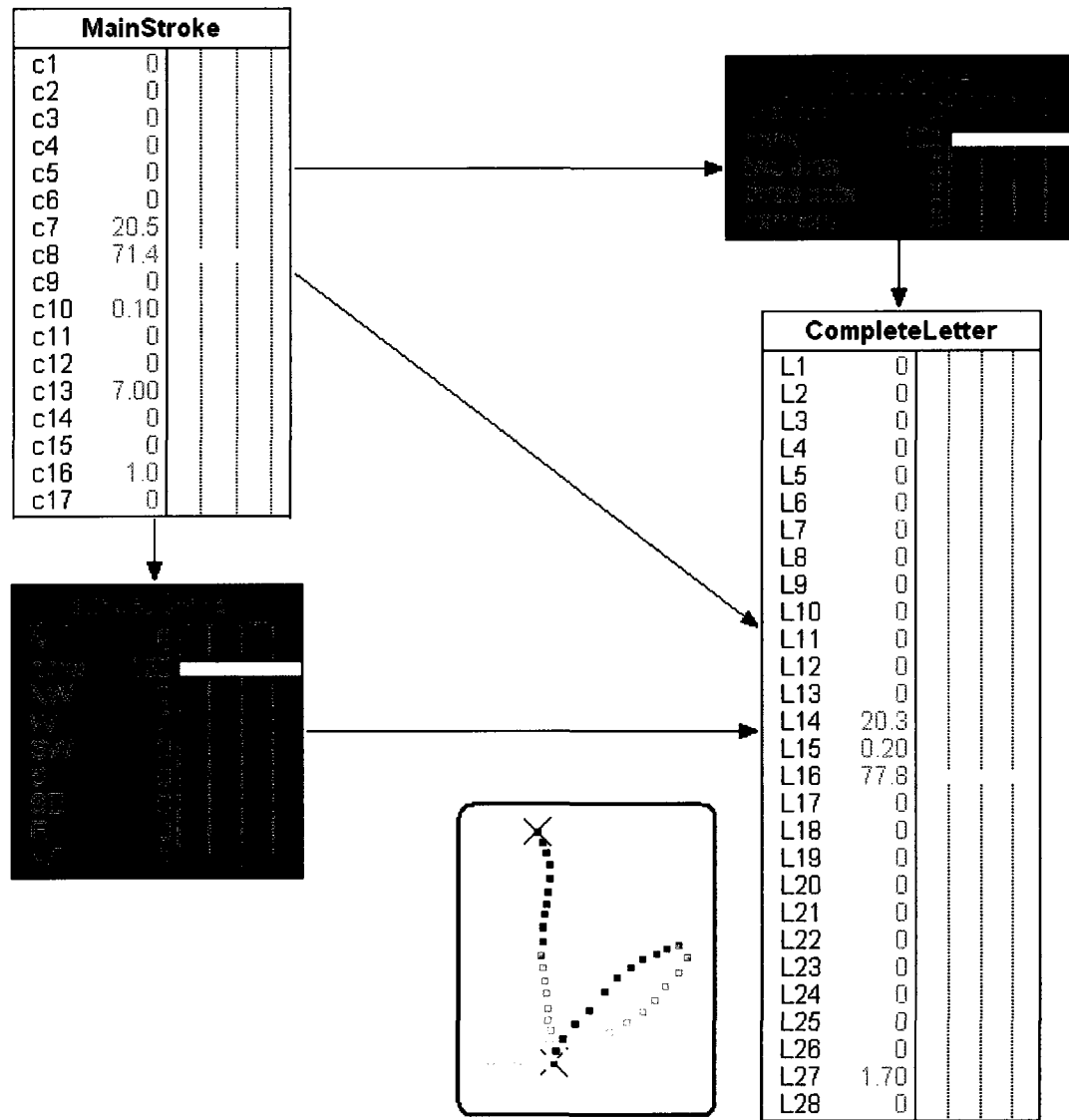


Figure 7.8: The probabilities for a sample of the Arabic letter taa in our BN hybrid system.

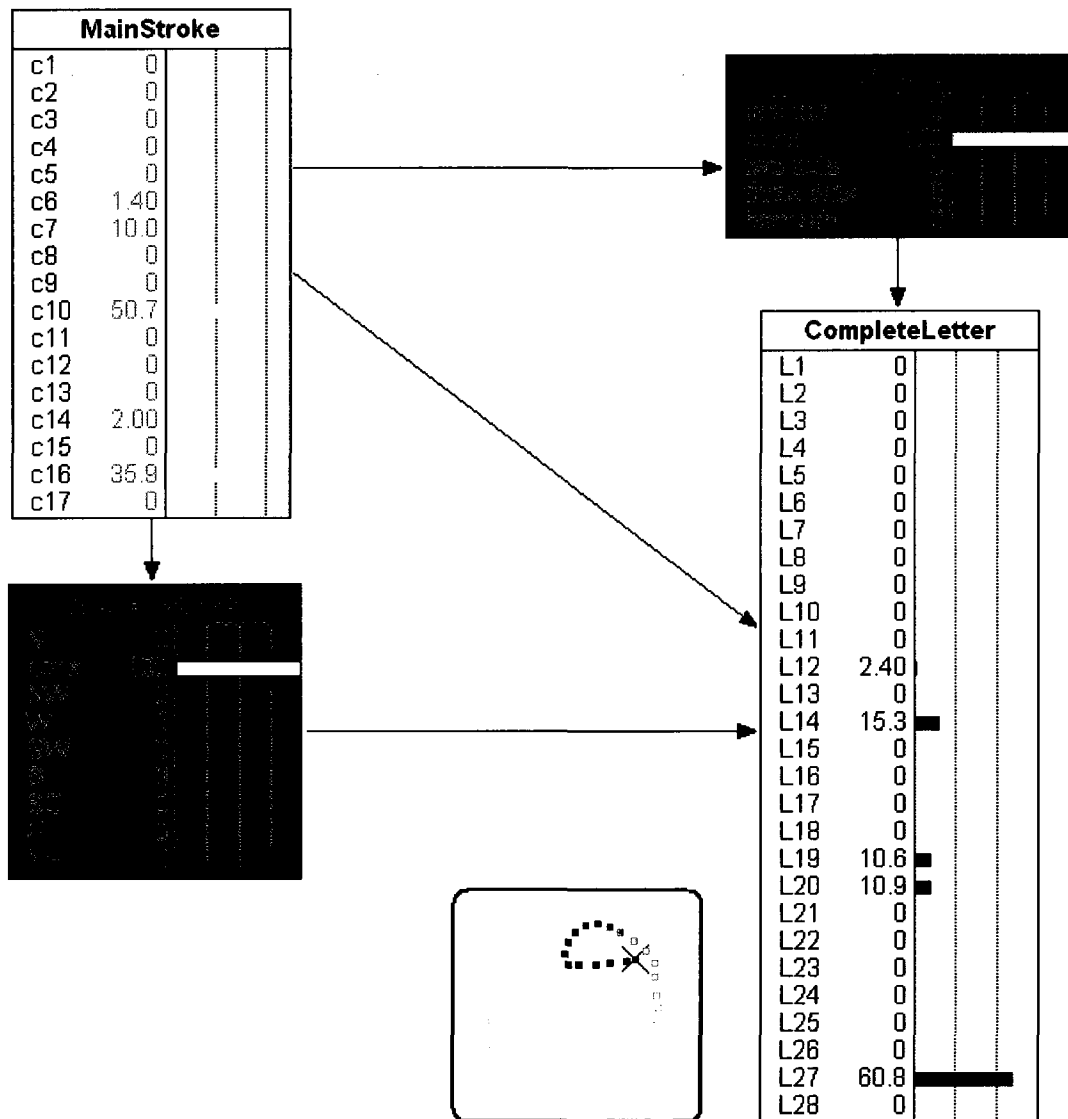


Figure 7.9: The probabilities for a sample of the Arabic letter waaw in our BN hybrid system.

7.5.1 Erroneous Cases

In this section, we will discuss sources of the errors for the hybrid recognition system that was introduced in Section 7.2. We will also present some of the misclassified samples along with practical suggestions for improving the system's performance.

Errors in our hybrid system could have been caused by the segmentation procedure, errors in recognizing the main shape, hidden strokes, and complementary strokes (even in the case of correct segmentation) or by ambiguous cases for which even humans would possibly have made the same mistakes. The three main categories of errors included segmentation error, classification error, and natural ambiguity error. These categories are discussed below.

Segmentation Error The false segmentation either wrongly locates or it misses the segmentation point. Wrongly locating the segmentation point causes the wrong shapes for both the main and complementary shapes. If those changes are severe, the main shape is recognized as another character with a high confidence value. In Figure 7.10, the wrongly located segmentation points are shown with red arrows, and the correct location of segmentation points are identified by green arrows. For the first three samples from the left, this figure shows segmentation points that are wrongly located. In the case of a missed segmentation point, pairs of letters, which only differ in having a complementary part such as *jiim* and *ha*, can easily confuse the system in classifying one letter as the other. This is shown in the last sample from the left. Missing the correct segmentation point could also increase the chance of classifying the main shape wrongly. The fourth sample from the left shows such

an error, where the letter *taa* is recognized as *ha*.

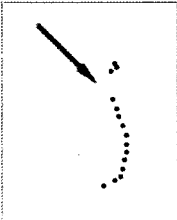
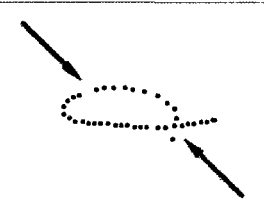
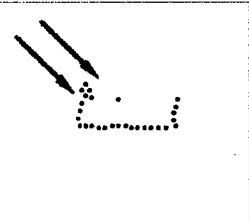
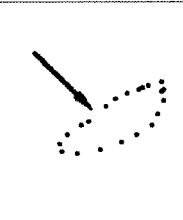
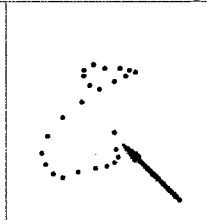
				
Alif	Baa	Taa	Taa	Jiim
Zayy	Taa	Nuun	Ha	Haa

Figure 7.10: Some misclassified samples for which the source of error is the segmentation failures. The top row indicates the real character label, while the bottom row shows the result of classification. Green arrows show the true segmentation points if any, while red arrows show the falsely detected location of the segmentation point.

Classification Error. In some cases, the misrecognized main or complementary shapes are the source of the classification error, while the segmentation may be error-free. The samples in Figure 7.11 depict this situation. For instance, as shown in this figure, the main shape of the letter *haa* has been mistaken three times by the main shape recognizer as *waaw*, *miim*, and *raa*.

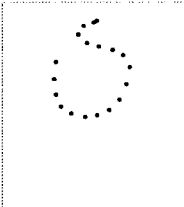
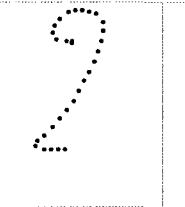
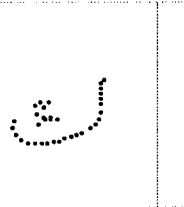
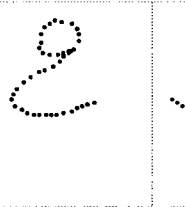
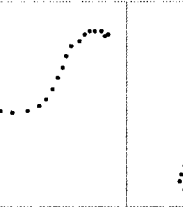

					
Yaa	Haa	Kaaf	Haa	Raa	Haa
Ha	Waaw	Thaa	Miim	Yaa	Raa

Figure 7.11: Some misclassified samples for which the source of error is the mistake in the recognition of main or complementary strokes. The top row indicates the real character label, while the bottom row shows the result of classification.

Natural Ambiguity. Some misclassified samples are due to the ambiguity of the samples. These kinds of samples are not usually given the same label when humans identify them. Figure 7.12 shows some examples in this category. The letters *dall* and *raa* are one example of such similar letters. Figure 7.12 shows that the letters *faa* and *qaaf* also can be written very similarly. Another ambiguity is that the recognition of two dots versus three dots is not straightforward. Such mistakes happen for some pairs of letters such as *taa* with two dots and *thaa* with three dots.









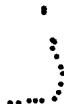

				
Daal	Raa	Raa	Haa	Faa
Laam	Daal	Daal	Miim	Qaaf
				
Faa	Taa	Taa	Zaay	Taa
Qaaf	Thaa	Thaa	Thaal	Thaa

Figure 7.12: Some misclassified samples for which the source of error is the intrinsic ambiguity of the sample.

7.6 Conclusion

We presented a hybrid multi-level recognizer for online Arabic letters in their complete natural shapes. This system requires segmentation of the complete shape, and uses a combination of multiple SVMs and a Bayesian network model. We proposed

a novel segmentation method for extracting three constructive components of online characters. The Bayesian network model allows for three-way reasoning and the quantization of the relationships concerning the recognition results of these components.

The presented recognition system in this chapter is unique in several ways. This is the first time that a full set of Arabic characters are considered in an online handwriting recognition system and are applied to a large database. The hybrid approach is also unique as this is the first time that a Bayesian network was used in such systems for online handwriting recognition. Therefore, we believe our approach has made a significant contribution to the field of online handwriting recognition in Arabic. Our results are encouraging in that the overall recognition rate with a complete shape is not much different from the recognition result of the main shape as presented in Chapter 5, on the same database. While in some studies such as [114], expanding the number of classes from 17 to 18 classes caused an accuracy reduction of about 5%, our results did not change significantly when increasing the number of classes from 17 to 28.

There are a number of exciting ways that this research could be further expanded, investigated, and improved. More sophisticated segmentation methods could probably prevent the error cases caused solely by segmentation and could potentially improve the recognition rates. While our database may be reasonably good in covering the variation and variability exhibited in natural handwritings, experimenting with more comprehensive databases will show how robust our approach is in overcoming these issues of variation and variability.

Chapter 8

Conclusions and Future Work

The main focus of this thesis was the recognition of online unconstrained handwritten Arabic characters, where our efforts were concentrated on the problems of improving the performance of such recognition systems. We emphasized that this is not a trivial problem, due to the inherent between-writer variation and within-writer variability of the data patterns and, more importantly, due to the considerable lack of a public database for developing and testing online Arabic handwriting recognition systems. In this thesis, we identified and designed approaches which are likely to help with these two key issues. The major contributions of this dissertation can be identified as follows: designing efficient feature representations to attempt resolving the issue of variation of the data patterns, and developing a more representative and a more comprehensive database of online Arabic characters compared to the existing databases. The design and dissemination of comprehensive databases, data representations, recognition algorithms and frameworks will ultimately contribute to improvements for a new generation of character recognition technology.

In this chapter, we revisit the main contributions of this thesis and highlight

avenues for future work.

8.1 Summary

We provided a detailed description of the database development and the specifications of the data collected from participants in Chapter 3. This is the first public online database of Arabic characters which contains the complete character shapes. The database was developed with a particular attention to the effects of handwriting variation and with constraints imposed, and to the human factors involved. The data annotation is also unique as it includes writer information. This information includes: native language, gender, age, writing habit (left-handed or right-handed), and education level of the writer. In total, there are over 10000 entries in the database with 360 samples for each character. The diversity of the participants contributed significantly to the variations and variability of the collected data. We demonstrated some interesting observations with respect to writing variation in terms of the number of strokes, the order of strokes, and the direction of strokes in Chapter 3.

Throughout Chapters 4 and 5, we showed that a character is classified by the means of a descriptor. This description is usually made up of parameters that were extracted or built from various parts of character's data. In this thesis, we proposed two representations for the online trajectory data. We called these representations relational context, *RC*, and relational histogram, *RH*, representations. We investigated the efficiency of these features when integrated with supervised learning methods for

classification. We used artificial neural networks with the *RH* representation and support vector machines with the *RC* representation. Both of these methods are known as the most efficient classifiers in the field of machine learning and pattern recognition. However, in the selection of classifiers, we had to consider the characteristics of feature representations as well. This was due to limitations of classifiers in handling arbitrary feature vectors. For instance, we pointed out that a good resolution feature vector in the *RC* representation will result in high dimensional feature vectors, and that makes the learning and recognition of the data computationally difficult.

Our results in Chapters 4 and 5 have showed that *RH* and *RC* features combined with the choice of classifiers are encouraging and efficient for recognizing Arabic isolated letters. We also conducted some experiments under equal conditions in order to compare the representational power of some of the most widely used features to our introduced representations. Our results showed improvements in the recognition rate in both *RC* and *RH* cases. Our *RC* feature did not only outperform the other methods in terms of recognition rate, but also in terms of a considerably smaller recognition time. The measured recognition speed was reasonable for the real-time applications. The robustness of the proposed features was also shown by our experiments on the large noisy data sets such as the one developed in this thesis, which contains broad within-class deviations.

In Chapter 6, we attempted to quantify the performance of a character feature representation with respect to separation of data samples in the corresponding feature space. A very flexible and widely used technique for such problems is to choose a metric for measuring the distance between samples of one class in comparison to the

their distance from the samples of another class. We pointed out the advantages of this type of separability analysis. This analysis, in particular, has beneficial visualization capabilities. We made use of statistical descriptors for the geometric shapes of feature vectors in the feature space, to help us perform the quantification of data separability. For this purpose, we examined a number of statistical distance metrics in the literature, however, not all of these metrics were applicable to our data under any representations. A suitable and analytically computable statistical distance measure that we were able to use was the Euclidean distance. Building on this promising metric, we performed quantitative performance validation of our feature representation in order to show the effectiveness of this proposed solution in comparison to other feature representations frequently used for online data. We witnessed the best performance in terms of inter-class and intra-class distances with the *RC* feature, and that can explain the considerable improvement in the recognition of an SVM-based system equipped with this representation.

In Chapter 7, we proposed a hybrid classifier for the recognition of online Arabic complete-shape characters. Almost all research on online handwriting recognition of Arabic characters in the literature has considered the main shape of the characters for the ease of classification. However, the main shape is not directly usable in practical applications and the complete-shape characters should be considered in order to classify the data. Extending the recognition systems based on the main shape to cover complete-shape data is not straightforward. Building on *RC* features, we developed an innovative approach for classification of the complete-shape characters

using multi-level recognitions with several SVMs and a higher-level Bayesian network for combining the recognition results of those SVMs to infer a probability-based complete-shape class label. We were able to evaluate our proposed methodology on the database collected in Chapter 3, since a unique feature of this database is that the data is stored in a complete-shape form. Given the high diversity of the samples in our database, our recognition results are promising.

8.2 Future Extensions

The success of our proposed feature representations for online handwriting of Arabic characters and the development of a unique database opens a vast scope of application opportunities. The following directions may be particularly promising:

An immediate extension of this work would be to use the motivation behind the suggested feature representations and examine the ability of the proposed systems for Arabic sub-word recognition from a limited dictionary. The suggested solutions for online feature representations is also transferable to many existing handwriting recognition systems, and can be useful for a number of recognition tasks such as the recognition of signatures, mathematical symbols, or shapes and drawings. Recent research on handwriting recognition has tried to address the lack of representative databases for handwritten data with the solution of designing generative models for simulating synthetic data ([72, 79]. It would be interesting to compare the recognition accuracy of the features proposed on the database developed in this thesis with the ones generated from these synthetic models.

The encouraging results given in this thesis for the application of SVMs with the *RC* feature could motivate the development of other variations of kernels for SVMs, which might improve the robustness of this classification technique. The choice of kernels is important in the success of SVM classification. The number of support vectors could also be reduced by optimizing model parameters through the introduction of a more sophisticated kernel. Consequently, this could improve the recognition speed even more.

In light of the difficulty of handling the variations and variability across databases (as explained in Chapter 5), the combination of multiple recognizers is a research direction that was partially explored in Chapter 7. Some classification errors in our hybrid system resulted from a faulty segmentation. It would be interesting to try a more sophisticated segmentation method in the hybrid system for avoiding this type of error.

There are several approaches in the literature for user adaptation, as reviewed in Chapter 2. It would also be interesting to build a user-dependent system based on the current user-free system, in order to obtain the maximum potential recognition rate of the proposed methodology.

In many sample collections, sometimes there is criticism that the database collection strategies are selective rather than comprehensive (i.e., the data was collected from individuals with a high or low education level, consisting of students or adults). However, little is known about how these factors really determine the unbiased variability selection. In data collection for the database that was presented in Chapter 3, we only followed the objective of designing a general and comprehensive data set.

However, the human factors annotated in the data makes it possible to analyze the impact of a writer's features on the recognition accuracy and to gain some insights on the writer's characteristics (e.g. native language), which could be important in determining how hard or easy the recognition task will be. It is also possible to investigate if we could predict the recognition rate for some people by knowing the recognition rate for others with a similar profile. We noticed that the participants were comfortable using the system to differing degrees and the recognition errors varied from one participant to another. The primary objective of new handwriting recognition systems must be to lower the percentage of error for those whose handwritings would not easily adapt. Therefore, if writer information has an impact on the recognition rate, then in designing a handwritten recognition application, human factors must be taken into account. In this case, considering a more general class of features regarding the demographics of the writer could make the system more acceptable to the users.

Bibliography

- [1] A. R. Ahmad, M. Khalia, C. Viard-Gaudin, and E. Poisson. Online handwriting recognition using support vector machine. In *TENCON '04: IEEE Region 10 International Technical Conference*, volume 1, pages 311–314, Nov. 2004.
- [2] S. Al-Emami and M. Usher. On-line recognition of handwritten Arabic characters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(7):704–710, 1990.
- [3] K. Alahari, S. L. Putrevu, and C. V. Jawahar. Learning mixtures of offline and online features for handwritten stroke recognition. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 379–382, Hong Kong, 2006.
- [4] A. M. Alimi and O. A. Ghorbel. The analysis of error in an on-line recognition system of Arabic handwritten characters. In *ICDAR '95: Proceedings of the 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 890–893, Montreal, Canada, 1995.
- [5] É. Anquetil and H. Bouchereau. Integration of an on-line handwriting recognition system in a smart phone device. In *ICPR '02: International Conference on Pattern Recognition*, volume 3, pages 192–194, Quebec City, Canada, 2002.
- [6] E. Anquetil and G. Lorette. Perceptual model of handwriting drawing application to the handwriting segmentation problem. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, page 112, Ulm, Germany, 1997.

- [7] H. Safadi B. Alsallakh. Arapen: An Arabic online handwriting recognition system. In *ICTTA '06: The 2nd International Conference on Information and Communication Technologies*, volume 1, pages 1844–1849, Damascus, Syria, April 2006.
- [8] V. Babu, L. Prasanth, R. Sharma, G. V. Rao, and A. Bharath. HMM-based online handwriting recognition system for Telugu symbols. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 63–67, Curitiba, Brazil, 2007.
- [9] M. S. Baghshah, S. B. Shouraki, and S. Kasaei. A novel fuzzy approach to recognition of online Persian handwriting. In *ISDA '05: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, pages 268–273, Wroclaw, Poland, 2005.
- [10] M. S. Baghshah, S. B. Shouraki, and S. Kasaei. A novel fuzzy classifier using fuzzy LVQ to recognize online Persian handwriting. In *ICTTA '06: The 2nd Information and Communication Technologies From Theory to Applications*, volume 1, pages 1878–1883, Damascus, Syria, April 2006.
- [11] C. Bahlmann and H. Burkhardt. The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(3):299–310, 2004.
- [12] C. Bahlmann, B. Haasdonk, and H. Burkhardt. On-line handwriting recognition with support vector machines- a kernel approach. In *IWFHR '02: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, pages 49–54, Niagara-on-the Lake, Ontario, Canada, 2002.
- [13] Z.-L. Bai and Q. Huo. A study on the use of 8-directional features for on-line handwritten Chinese character recognition. In *ICDAR '05: Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 262–266, Seoul, Korea, 2005.

- [14] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522, 2002.
- [15] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges. LeRec: a NN/HMM hybrid for on-line handwriting recognition. *Neural Computation*, 7(6):1289–1303, 1995.
- [16] M. J. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, Inc., New York, NY, USA, 1997.
- [17] A. Bharath and S. Madhvanath. Hidden Markov models for online handwritten Tamil word recognition. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 506–510, Curitiba, Brazil, 2007.
- [18] U. Bhattacharya, B. K. Gupta, and S. Parui. Direction code based features for recognition of online handwritten characters of Bangla. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 58–62, Curitiba, Brazil, 2007.
- [19] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.
- [20] F. Biadsy, J. El-Sana, and N. Habash. Online Arabic handwriting recognition using Hidden Markov Models. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 85–90, La Baule, France, Oct. 2006.
- [21] A. Biem. Minimum classification error training for online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(7):1041–1051, 2006.

- [22] K. Bounnady, B. Kruatrachue, and S. Wangsiripitak. On-line lao handwritten recognition with proportional invariant feature. In *WEC '05: The Third World Enformatika Conference*, pages 41–44, Istanbul, Turkey, 2005.
- [23] F. Bouslama. Structural and fuzzy techniques in the recognition of online Arabic characters. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 13(7):1027–1040, 1999.
- [24] F. Bouslama and A. Amin. Pen-based recognition system of Arabic character utilizing structural and fuzzy techniques. In *KES '98: The 2nd International Conference on Knowledge-Based Intelligent Electronic Systems*, volume 3, pages 76–85, Adelaide, South Australia, April 1998.
- [25] A. Brakensiek, A. Kosmala, and G. Rigoll. Comparing adaptation techniques for on-line handwriting recognition. In *ICDAR '01: Proceedings of the 6th International Conference on Document Analysis and Recognition*, pages 486–490, Seattle, USA, 2001.
- [26] A. Brakensiek, A. Kosmala, and G. Rigoll. Comparing normalization and adaptation techniques for on-line handwriting recognition. In *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 73–76, Quebec City, Canada, 2002.
- [27] A. Brakensiek, A. Kosmala, and G. Rigoll. Evaluation of confidence measures for on-line handwriting recognition. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 507–514, London, UK, 2002.
- [28] J. J. Brault and R. Plamondon. Segmenting handwritten signatures at their perceptually important points. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(9):953–957, 1993.
- [29] H. Bunke and A. Sanfeliu. *Syntactic and Structural Pattern Recognition: Theory and Applications*. World Scientific Pub. Co. Inc, 1990.

- [30] E. Caillault and C. Viard-Gaudin. Using segmentation constraints in an implicit segmentation scheme for on-line word recognition. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 607–612, La Baule, France, October 2006.
- [31] W. M. Campbell. A covariance kernel for SVM language recognition. In *ICASSP '08 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4141–4144, Nevada, USA, 2008.
- [32] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(7):690–706, 1996.
- [33] K.-F. Chan and D.-Y. Yeung. A simple yet robust structural approach for recognizing on-line handwritten alphanumerical characters. In *IWFHR '98: 6th International Workshop on Frontiers in Handwriting Recognition*, pages 229–238, Taejon, Korea, August 1998.
- [34] K.-F. Chan and D.-Y. Yeung. PenCalc: A novel application of on-line mathematical expression recognition technology. In *ICDAR '01: Proceedings of the 6th International Conference on Document Analysis and Recognition*, pages 774–778, 2001.
- [35] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [36] B. W. Char and S. M. Watt. Representing and characterizing handwritten mathematical symbols through succinct functional approximation. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, volume 2, pages 1198–1202, Curitiba, Brazil, 2007.
- [37] H. Chernoff. A note on an inequality involving the normal distribution. *Annals of Probability*, 9(3):533–535, 1981.

- [38] S. D. Connell and A. K. Jain. Writer adaptation for online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(3):329–346, 2002.
- [39] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [40] D. R. Cox and N. J. H. Small. Testing multivariate normality. *Biometrika*, 65(2):263–272, August 1978.
- [41] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals and Systems (MCSS)*, 2(4):303–314, Oct. 1989.
- [42] L. M. de Campos. Independency relationships and learning algorithms for singly connected networks. *Experimental and Theoretical Artificial Intelligence*, 10(4):511–549, 1998.
- [43] J. P. Marques de Sa. *Pattern recognition, concepts, methods and applications*. Springer, 1st edition, Aug. 2001.
- [44] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39(1):1–38, 1977.
- [45] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, London, 1982.
- [46] D. Hammarskjöld Library (DHL). INDEXES : United Nations Documentation, The Department of Public Information (DPI), 2007. Retrieved on 2008-1-15 at <http://www.un.org/Depts/dhl/resguide/itp.htm>.
- [47] M. Dinesh and M. K. Sridhar. A feature based on encoding the relative position of a point in the character for online handwritten character recognition. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 1014–1017, 2007.

- [48] A. Djouadi, O. Snorrason, and F. D. Garber. The quality of training-sample estimates of the Bhattacharyya coefficient. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):92–97, 1990.
- [49] M. Eden. Handwriting and pattern recognition. *IRE Transactions on Information Theory*, 8(2):160–166, February 1962.
- [50] T. S. El-Sheikh and S. G. El-Taweel. Real-time Arabic handwritten character recognition. *Pattern Recognition*, 23(12):1323–1332, 1990.
- [51] T.S. El-Sheikh and S.G. El-Taweel. Real-time Arabic handwritten character recognition. In *Third International Conference on Image Processing and its Applications*, pages 212–216, Warwick, UK, 1989.
- [52] R. I. Elanwar, M. A. Rashwan, and S. A. Mashali. Simultaneous segmentation and recognition of Arabic characters in an unconstrained on-line cursive handwritten document. *International Journal of Computer and Information Science and Engineering (IJCISE)*, 1(4):203–206, 2007.
- [53] R. I. Elanwar, M. A. Rashwan, and S. A. Mashali. Simultaneous segmentation and recognition of Arabic characters in an unconstrained on-line cursive handwritten document. In *WASET '07: Proceedings of the World Academy of Science, Engineering and Technology*, volume 29, pages 288–291, Berlin, Germany, May 2007.
- [54] A. Elgammal and M. A. Ismail. A graph-based segmentation and feature-extraction framework for arabic text recognition. In *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, page 622, Seattle, USA, 2001.
- [55] I. M. M. El Emary and M. M. Hammad. On-line structural approach for recognizing hand-written Indian numbers. *International Journal on Graphics, Vision and Image Processing (ICGST)*, 05(7):21–26, July 2005.

- [56] M. Friedman and A. Kandel. *Introduction to Pattern Recognition: Statistical, Structural, Neural and Fuzzy Logic Approaches (Series in Machine Perception and Artificial Intelligence)*. World Scientific Publishing Company, 1999.
- [57] F. D. Garber and A. Djouadi. Bounds on the Bayes classification error based on pairwise risk functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 10(2):281–288, 1988.
- [58] O. Golubitsky and S. M. Watt. Online recognition of multi-stroke symbols with orthogonal series. In *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 1265–1269, Barcelona, Spain, 2009.
- [59] E. Grinspun. Didactic walkthrough. A SIGGRAPH 2005 Course on: Discrete Differential Geometry, 2005.
- [60] Miniwatts Marketing Group. Internet world stats, internet world users by language, top 10 languages updated for december 31, 2008. Retrieved on 2007-9-10 at <http://www.internetworldstats.com/stats7.htm>.
- [61] W. Guerfali and R. Plamondon. Normalizing and restoring on-line handwriting. *Pattern Recognition*, 26(3):419–431, March 1993.
- [62] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN project of on-line data exchange and benchmarks. In *ICPR' 94: Proceedings of the 12th International Conference on Pattern Recognition*, pages 29–33, Jerusalem, Israel, Sep 1994.
- [63] R. Halavati, M. Jamzad, and M. Soleymani. A novel approach to Persian online handwriting recognition. In *WASET '05: Proceedings of World Academy of Science, Engineering and Technology*, volume 11, pages 81–85, Tokyo, Japan, November 2005.

- [64] R. Halavati and S. B. Shouraki. Recognition of Persian online handwriting using elastic fuzzy pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 21(3):491–513, 2007.
- [65] R. Halavati, S. B. Shouraki, and S. Hassanpour. Evolution of multiple states machines for recognition of online cursive handwriting. In *WAC '06: World Automation Congress*, pages 1–6, Budapest, July 2006.
- [66] R. Halavati, S. B. Souraki, and M. Soleymani. Persian online handwriting recognition using fuzzy modeling. In *IFSA '05: International Fuzzy Systems Association World Congress*, pages 268–273, Beijing, China, 2005.
- [67] P. Haluptzok, M. Revow, and A. Abdulkader. Personalization of an online handwriting recognition system. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 79–83, La Baule, France, Oct. 2006.
- [68] J. Han and M. Kamber. *Data Mining. Concepts and Techniques*. Morgan Kaufmann, 2nd edition, 2006.
- [69] S. Han, M. Chang, Y. Zou, X. Chen, and D. Zhang. Systematic multi-path HMM topology design for online handwriting recognition of East Asian characters. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 604–608, Curitiba, Brazil, 2007.
- [70] J.-F. Hebert, M. Parizeau, and N. Ghazzali. A new fuzzy geometric representation for on-line isolated character recognition. In *ICPR '98: Proceedings of the 14th International Conference of Pattern Recognition*, volume 2, pages 1121–1123, Brisbane, Australia, 1998.
- [71] A. Hennig, N. Sherkat, and R. J. Whitrow. Recognizing letters in on-line handwriting using hierarchical fuzzy inference. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, volume 2, pages 936–940, Ulm, Germany, 1997.

- [72] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.
- [73] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.
- [74] J. Hu, M. K. Brown, and W. Turin. HMM based on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(10):1039–1045, 1996.
- [75] J. Hu, A. S. Rosenthal, and M. K. Brown. Combining high-level features with sequential local features for on-line handwriting recognition. In *ICIAP '97: Proceedings of the 9th International Conference on Image Analysis and Processing*, volume 2, pages 647–654, London, UK, 1997.
- [76] B. Q. Huang, C. J. Du, Y. B. Zhang, and M-T. Kechadi. A hybrid HMM-SVM method for online handwriting symbol recognition. In *ISDA '06: Proceedings of the 6th International Conference on Intelligent Systems Design and Applications*, pages 887–891, Jinan, China, 2006.
- [77] B. Q. Huang and M-T. Kechadi. A fast feature selection model for online handwriting symbol recognition. In *ICMLA '06: Proceedings of the 5th International Conference on Machine Learning and Applications*, pages 251–257, Orlando, Florida, USA, 2006.
- [78] H. H. Hwawen and A. R. Newton. Recognition and beautification of multi-stroke symbols in digital ink. *Computers and Graphics*, 29(4):533–546, 2005.
- [79] S. Izadi, M. Haji, and C. Y. Suen. A new segmentation algorithm for online handwritten word recognition in Persian script. In *ICFHR '08: International Conference on Frontiers in Handwriting Recognition*, pages 598–603, Montreal, Canada, Aug. 2008.

- [80] S. Izadi, J. Sadri, F. Solimanpour, and C. Y. Suen. A review on Persian script and recognition techniques. *Arabic and Chinese Handwriting Recognition, Lecture Notes in Computer Science (LNCS)*, 4768:22–35, March 2008.
- [81] S. Izadi and C. Y. Suen. Incorporating a new relational feature in online handwritten character recognition. In *WIML '07: International Workshop on Machine Learning: Theory, Application and Experiences*, Florida, USA, Oct. 2007.
- [82] S. Izadi and C. Y. Suen. Incorporating a new relational feature in Arabic online handwritten character recognition. In *VISAPP '08: Proceedings of the Third International Conference on Computer Vision Theory and Applications*, volume 1, pages 559–562, PortugalMadeira, Portugal, January 2008.
- [83] S. Izadi and C. Y. Suen. Online writer-independent character recognition using a novel relational context representation. In *ICMLA '08: The 4th International Conference on Machine Learning and Applications*, pages 867–870, San Diego, CA, USA, 2008.
- [84] S. Izadi and C. Y. Suen. Integration of contextual information in online handwriting representation. In *ICIAP '09: International Conference on Image Analysis and Processing*, pages 132–142, Italy, 2009.
- [85] J. M. Chambers and W. S. Cleveland and B. Kleiner and P. A. Tukey. *Graphical Methods for Data Analysis*. Chapman and Hall, New York, 1983.
- [86] S. Jaeger, S. Manke, and A. Waibel. NPEN++: An on-line handwriting recognition system. In *IWFHR '00: In Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition*, pages 249–260, Amsterdam, September 2000.
- [87] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [88] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

- [89] N. Joshi, G. Sita, A. G. Ramakrishnan, V. Deepu, and S. Madhvanath. Machine recognition of online handwritten Devanagari characters. In *ICDAR '05: Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 1156–1160, Seoul, Korea, 2005.
- [90] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath. Comparison of elastic matching algorithms for online Tamil handwritten character recognition. In *IWFHR '04: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 444–449, Tokyo, Japan, 2004.
- [91] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, February 1967.
- [92] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.
- [93] D.-W. Kim, J. B. Park, and Y. H. Joo. Design of fuzzy rule-based classifier: Pruning and learning. In *FSKD '05: The 2nd International Conference on Fuzzy Systems and Knowledge Discovery*, pages 416–425, 2005.
- [94] S.-W. Kim and B. J. Oommen. On using prototype reduction schemes to optimize dissimilarity-based classification. *Pattern Recognition*, 40(11):2946–2957, 2007.
- [95] T. J. Klassen and M. I. Heywood. Towards the on-line recognition of Arabic characters. In *IJCNN '02: Proceedings of the 2002 International Joint Conference on Neural Networks*, volume 2, pages 1900–1905, Honolulu, Hawaii, May 2002.
- [96] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences, Springer, Berlin, Heidelberg, New York, 3rd extended edition edition, 2001.
- [97] K. B. Korb and A. E. Nicholson. *Bayesian artificial intelligence*. Computer Science and Data Analysis. Chapman andHall/Crc, 2004.

- [98] U. H.-G. Kressel. Pairwise classification and support vector machines. *Advances in kernel methods: support vector learning*, pages 255–268, 1999.
- [99] K. Kryszczuk and A. Drygajlo. Impact of feature correlations on separation between bivariate normal distributions. In *booktitle = ICPR '08: Proceedings of the 19th International Conference on Pattern Recognition,,* pages 1–4, Tampa, Florida, USA, 2008.
- [100] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [101] E. L. Lehmann and G. Casella. *Theory of Point Estimation (Springer Texts in Statistics)*. Springer, September 2003.
- [102] C.-L. Liu, S. Jaeger, and M. Nakagawa. Online recognition of Chinese characters: The state-of-the-art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(2):198–213, 2004.
- [103] Y. J. Liu and J. W. Tai. A structural approach to online Chinese character recognition. In *ICPR '88: Proceedings of the 9th International Conference on Pattern Recognition*, volume 2, pages 808–810, Rome, Italy, 1988.
- [104] M. Liwicki and H. Bunke. HMM-based on-line recognition of handwritten whiteboard notes. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 595–599, La Baule, France, 2006.
- [105] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 367–371, Curitiba, Brazil, September 2007.
- [106] T. Long and L.-W. Jin. Hybrid recognition for one stroke style cursive handwriting characters. In *ICDAR '05: Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 232–236, Seoul, Korea, 2005.

- [107] T. Long, L.-W. Jin, L.-X. Zhen, and J.-C. Huang. One stroke cursive character recognition using combination of directional and positional features. In *ICASSP '05: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 449–452, Philadelphia, PA, USA, March 2005.
- [108] M. M. Liwicki and H. Bunke. Feature selection for on-line handwriting recognition of whiteboard notes. In *IGS' 07: The 13th Conference of the International Graphonomics Society*, pages 101–105, Melbourne, Australia, 2007.
- [109] S. Manke and U. Bodenhausen. A connectionist recognizer for on-line cursive handwriting recognition. In *ICASSP' 94: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 596–598, Adelaide, 1994.
- [110] N. Mezghani, M. Cheriet, and A. Mitiche. Combination of pruned kohonen maps for on-line Arabic characters recognition. In *ICDAR '03: Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 900–904, Edinburgh, Scotland, 2003.
- [111] N. Mezghani, A. Mitiche, , and M. Cheriet. A new representation of character shape and its use in on-line character recognition by a self organizing map. In *ICIP '04: IEEE International Conference on Image Processing*, volume 3, pages 2123–2126, Singapore, 2004.
- [112] N. Mezghani, A. Mitiche, and M. Cheriet. On-line recognition of handwritten Arabic characters using a kohonen neural network. In *IWFHR '02: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, pages 490–495, Niagara-on-the Lake, Ontario, Canada, 2002.
- [113] N. Mezghani, A. Mitiche, and M. Cheriet. A new representation of shape and its use for high performance in online Arabic character recognition by an associative memory. *International Journal on Document Analysis and Recognition (IJDAR)*, 7(4):201–210, 2005.

- [114] N. Mezghani, A. Mitiche, and M. Cheriet. Bayes classification of online Arabic characters by Gibbs modeling of class conditional densities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(7):1121–1131, July 2008.
- [115] H. Mitoma, S. Uchida, and H. Sakoe. Online character recognition using eigen-deformations. In *IWFHR '04: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 3–8, Tokyo, Japan, 2004.
- [116] H. Mitoma, S. Uchida, and H. Sakoe. Online character recognition based on elastic matching and quadratic discrimination. In *ICDAR '05: Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 36–40, Seoul, Korea, 2005.
- [117] H. Mouchere, E. Anquetil, and N. Ragot. On-line writer adaptation for handwriting recognition using fuzzy inference systems. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, volume 2, pages 1075–1079, Seoul, Korea, 2005.
- [118] S. Mukkamala and A. H. Sung. Feature selection for intrusion detection using neural networks and support vector machines. *Journal of the Transportation Research Board of the National Academies*, pages 33–39, 2003.
- [119] M. Nakai, T. Sudo, H. Shimodaira, and S. Sagayama. Pen pressure features for writer-independent on-line handwriting recognition based on substroke HMM. In *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 220–223, Quebec City, Canada, 2002.
- [120] A. Nakamura. A method to accelerate writer adaptation for on-line handwriting recognition of a large character set. In *IWFHR '04: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 426–431, Tokyo, Japan, 2004.

- [121] L. Nanni. Cluster-based pattern discrimination: a novel technique for feature selection. *Pattern Recognition Letters*, 27(6):682–687, April 2006.
- [122] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.
- [123] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *NIPS '01: Neural Information Processing Systems Conference*, pages 841–848, Vancouver, British Columbia, Canada, 2001.
- [124] R. Niels and L. Vuurpijl. Dynamic timewarping applied to Tamil character recognition. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 730–734, Seoul, Korea, 2005.
- [125] R. Niels and L. Vuurpijl. Generating copybooks from consistent handwriting styles. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 1009–1013, Curitiba, Brazil, 2007.
- [126] R. Niels, L. Vuurpijl, and L. Schomaker. Automatic allograph matching in forensic writer identification. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 21(1):61–81, 2007.
- [127] D. Nikovski. Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):509–516, 2000.
- [128] H. Oda, B. Zhu, J. Tokuno, M. Onuma, A. Kitadai, and M. Nakagawa. A compact on-line and off-line combined recognizer. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 133–138, La Baule, France, Oct. 2006.
- [129] J. Oh. *An On-Line Handwriting Recognizer with Fisher Matching, Hypotheses Propagation Network and Context Constraint Models* Generalized feature extraction for structural pattern recognition in time-series data. PhD thesis, New York University, New York, USA, 2001.

- [130] R. T. Olszewski. *Generalized feature extraction for structural pattern recognition in time-series data*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2001.
- [131] A. Onisko, P. Lucas, and M. J. Druzdzel. Comparison of rule-based and bayesian network approaches in medical diagnostic systems. In *AIME '01: Proceedings of the 8th Conference on AI in Medicine in Europe*, pages 283–292, London, UK, 2001. Springer-Verlag.
- [132] Encyclopedia Britannica online. Arabic alphabet. Retrieved on 2007-11-23 at <http://www.britannica.com/eb/article-9008156/Arabic-alphabet>.
- [133] M. Parizeau and R. Plamondon. Machine vs humans in a cursive script reading experiment without linguistic knowledge. In *ICPR' 94: Proceedings of the 12th International Conference on Pattern Recognition*, pages 93–98, Jerusalem, Israel, 1994.
- [134] B. Paskalevaz, M. M. Hayatz, M. M. Moyay, and R. J. Foglery. Multispectral rock type separation and classification. In *49th Annual Meeting of the SPIE: Infrared Spaceborne Remote Sensing XII*, volume 5543, pages 152–163, Denver, August 2004.
- [135] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [136] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In *NIPS '00: Advances in Neural Information Processing Systems Conference*, pages 547–553, Denver, Colorado, USA, 1999.
- [137] E. Pranckeviciene, T. Ho, and R. Somorjai. Class separability in spaces reduced by feature selection. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 254–257, Hong Kong, 2006.
- [138] Y. Qiao and M. Yasuhara. Affine invariant dynamic time warping and its application to online rotated handwriting recognition. In *ICPR '06: Proceedings*

of the 18th International Conference on Pattern Recognition, pages 905–908, Hong Kong, 2006.

- [139] A. J. Quiroz and R. M. Dudley. Some new tests for multivariate normality. *Probability Theory and Related Fields*, 87(4):521–546, December 1991.
- [140] B. S. Raghavendra, C. K. Narayanan, G. Sita, A. G. Ramakrishnan, and M. Sri-ganesh. Prototype learning methods for online handwriting recognition. In *IC-DAR '05: Proceedings of the 8th International Conference on Document Anal-ysis and Recognition*, pages 287–291, Seoul, Korea, 2005.
- [141] N. Ragot and E. Anquetil. A generic hybrid classifier based on hierarchical fuzzy modeling: experiments on on-line handwritten character recognition. In *ICDAR'03: Proceedings of the 7th International Conference on Document Anal-ysis and Recognition*, pages 963–967, Edinburgh, Scotland, 2003.
- [142] S. M. Razavi and E. Kabir. A database for online Persian handwritten recog-nition. In *6th Iranian Conference on Intelligent Systems, (in Persian)*, pages 859–863, Kerman, Iran, 2004.
- [143] S. M. Razavi and E. Kabir. Online Persian isolated character recognition. In *MVIP '05: The Third Iranian Conference on Machine Vision, Image Process-ing and Applications, (in Persian)*, pages 83–89, Tehran, Iran, Feb. 2005.
- [144] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 2003.
- [145] J. Sadri, S. Izadi, F. Solimanpour, C. Y. Suen, , and T. D. Bui. State-of-the-art in Farsi script recognition. In *ISSPA '07: International Symposium on Signal Processing and its Applications*, pages 859–863, Sharjah, UAE, Feb. 2007.
- [146] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb 1978.

- [147] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.
- [148] E.G. Sanchez, J.A.G. Gonzalez, Y.A. Dimitriadis, J.M.C. Izquierdo, and J.L. Coronado. Experimental study of a novel neuro fuzzy system for online handwritten unipen digit recognition. *Pattern Recognition Letters*, 19(3-4):357–364, March 1998.
- [149] G. Saon and M. Padmanabhan. Minimum Bayes error feature selection for continuous speech recognition. In *NIPS' 2000: Advances in Neural Information Processing Systems*, volume 13, pages 800–806, Denver, CO, USA, 2000.
- [150] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time delay neural networks and hidden markov models. *Machine Vision and Applications*, 8(4):215–223, 1995.
- [151] L. Schomaker. From handwriting analysis to pen-computer applications. *Electronics and Communication Engineering Journal*, 10(3):93–102, Jun 1998.
- [152] G. Seni, R. K. Srihari, and N. Nasrabadi. Large vocabulary recognition of on-line handwritten cursive words. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(7):757–762, 1996.
- [153] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 3(52):591–611, 1965.
- [154] B.-K. Sin and J. H. Kim. Ligature modeling for online cursive script recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(6):623–633, 1997.
- [155] A. Sluzek. Using moment invariants to recognize and locate partially occluded 2d objects. *Pattern Recognition Letters*, 7:253–257, 1988.
- [156] S. P. Smith and A. K. Jain. A test to determine the multivariate normality of a data set. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 10(5):757–761, 1988.

- [157] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP' 94: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 125–128, Adelaide, Australia, April 1994.
- [158] C. D. Stefano, M. Garruto, and A. Marcelli. A saliency-based multiscale method for on-line cursive handwriting shape description. In *IWFHR '04: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 124–129, Tokyo, Japan, 2004.
- [159] J. Sternby. Structurally based template matching of on-line handwritten characters. In *BMVC '05: Proceedings of the 16th British Machine Vision Conference*, volume 1, pages 250–259, Oxford, UK, 2005.
- [160] J. Sternby, J. Morwing, J. Andersson, and C. Friberg. On-line arabic handwriting recognition with templates. *Pattern Recognition*, 42(12):3278–3286, 2009.
- [161] C. Y. Suen, S. Izadi, J. Sadri, and F. Solimanpour. Farsi script recognition-A survey. In *SACH '06: Proceedings of International Summit on Arabic and Chinese Handwriting*, pages 101–110, University of Maryland, USA, Sept. 2006.
- [162] S. Sundaram and A. Ramakrishnan. A novel hierarchical classification scheme for online Tamil character recognition. In *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 1218–1222, Curitiba, Brazil, 2007.
- [163] M. Szummer and C. M. Bishop. Discriminative writer adaptation. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 595–599, La Baule, France, October 2006.
- [164] C. C. Tappert, C. Y. Suen, and T. Wakahara. The state of the art in online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(8):787–808, 1990.

- [165] A. Teredesai, E. H. Ratzlaff, J. Subrahmonia, and V. Govindaraju. On-line digit recognition using off-line features. In *ICVGIP '02: The 3rd Indian Conference on Computer Vision, Graphics and Image Processing*, Ahmadabad, India, 2002.
- [166] J. Tokuno, M. Nakai, H. Shimodaira, S. Sagayama, and M. Nakagawa. On-line handwritten character recognition selectively employing hierarchical spatial relationships among subpatterns. In *IWFHR '06: The 10th International Workshop on Frontiers of Handwriting Recognition*, pages 139–144, La Baule, France, Oct. 2006.
- [167] G.T. Toussaint. An upper bound on the probability of misclassification in terms of the affinity. In *IEEE Proceedings*, volume 65, pages 275–276, Feb. 1977.
- [168] O. Trier, A. Jain, and T. Taxt. Feature extraction methods for character recognition - A survey. *Pattern Recognition*, 29(4):641–662, April 1996.
- [169] S. Uchida and H. Sakoe. A survey of elastic matching techniques for handwritten character recognition. *IEICE Transactions on Information and Systems*, E88-D(8):1781–1790, 2005.
- [170] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, USA, 1998.
- [171] B. Verma and M. Ghosh. A neural-evolutionary approach for feature and architecture selection in online handwriting recognition. In *ICDAR '03: Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 1038–1042, Edinburgh, Scotland, 2003.
- [172] A. Vinciarelli and M. Perrone. Combining online and offline handwriting recognition. In *ICDAR '03: Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 844–848, Edinburgh, Scotland, 2003.

- [173] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Speeding up on-line recognition of handwritten characters by pruning the prototype set. In *ICDAR '01: Proceedings of the 6th International Conference on Document Analysis and Recognition*, pages 501–505, Seattle, USA, 2001.
- [174] L. Vuurpijl, R. Niels, M. V. Erp, and L. Schomaker. Verifying the unipen devset. In *IWFHR '04: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 586–591, Tokyo, Japan, 2004.
- [175] L. Vuurpijl and L. Schomaker. Finding structure in diversity: a hierarchical clustering method for the categorization of allographs in handwriting. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, volume 1, pages 387–393, Ulm, Germany, Aug. 1997.
- [176] S. M. Watt and X. Xie. Recognition for large sets of handwritten mathematical symbols. In *ICDAR '05: Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 740–744, 2005.
- [177] A. R. Webb. *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, October 2002.
- [178] D. Willems, R. Nielsa, M. van Gerven, and L. Vuurpijl. Iconic and multi-stroke gesture recognition. *Pattern Recognition*, 42(12):3303–3312, 2009.
- [179] D. Willems, S. Rossignol, and L. Vuurpijl. Mode detection in on-line pen drawing and handwriting recognition. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 31–35, Seoul, Korea, 2005.
- [180] M. F. Zafar, D. Mohamad, and R. Othman. Neural nets for on-line isolated handwritten character recognition: A comparative study. In *IEEE International Conference on Engineering of Intelligent Systems*, pages 1–6, Islamabad, Pakistan, 2006.

APPENDIX 1

The results of the experiments with QQ-plots for RC feature for the 17 classes of our Arabic dataset is shown in the following figures.

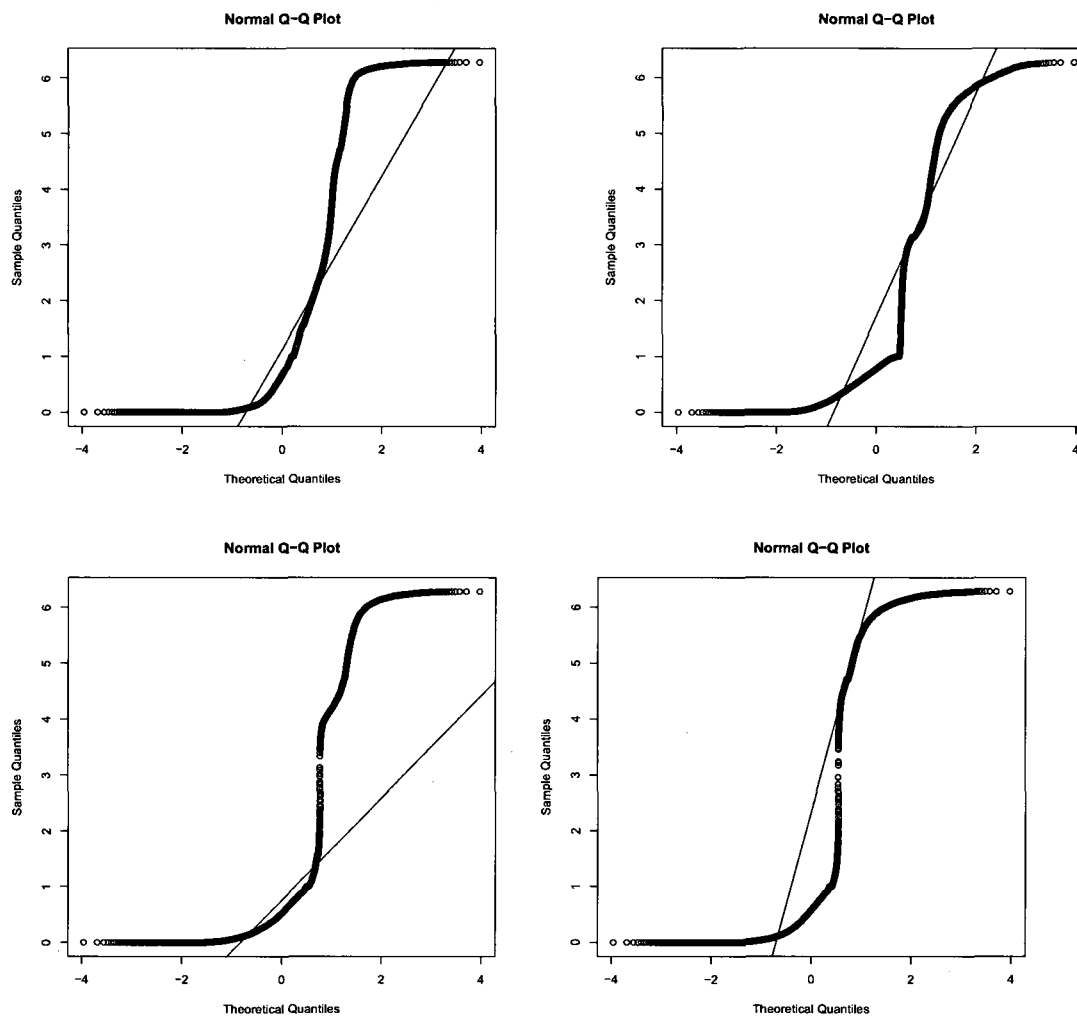


Figure 1: QQ-plots of all 17 classes of Arabic character shapes using RC feature representations; part 1 of 4.

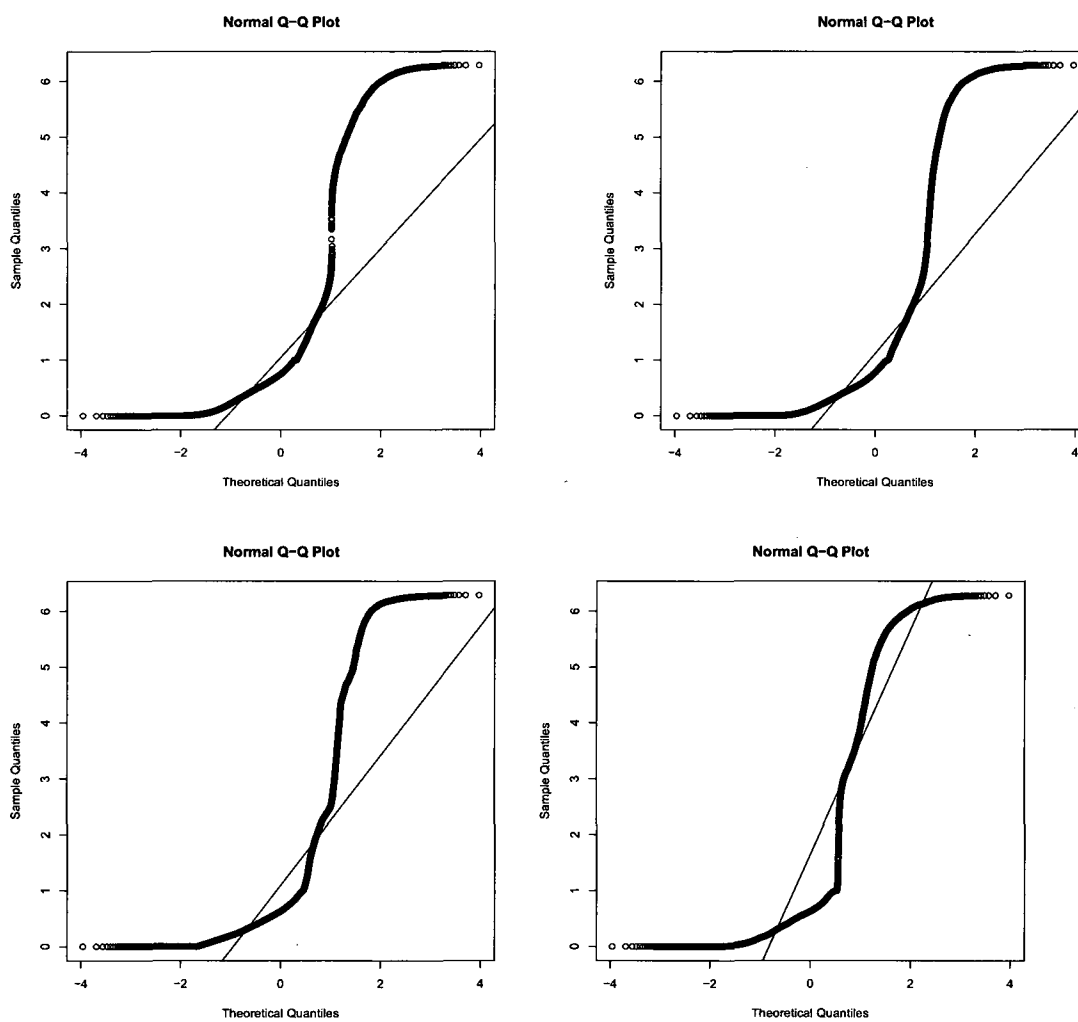


Figure 2: QQ-plots of all 17 classes of Arabic character shapes using *RC* feature representations; part 2 of 4.

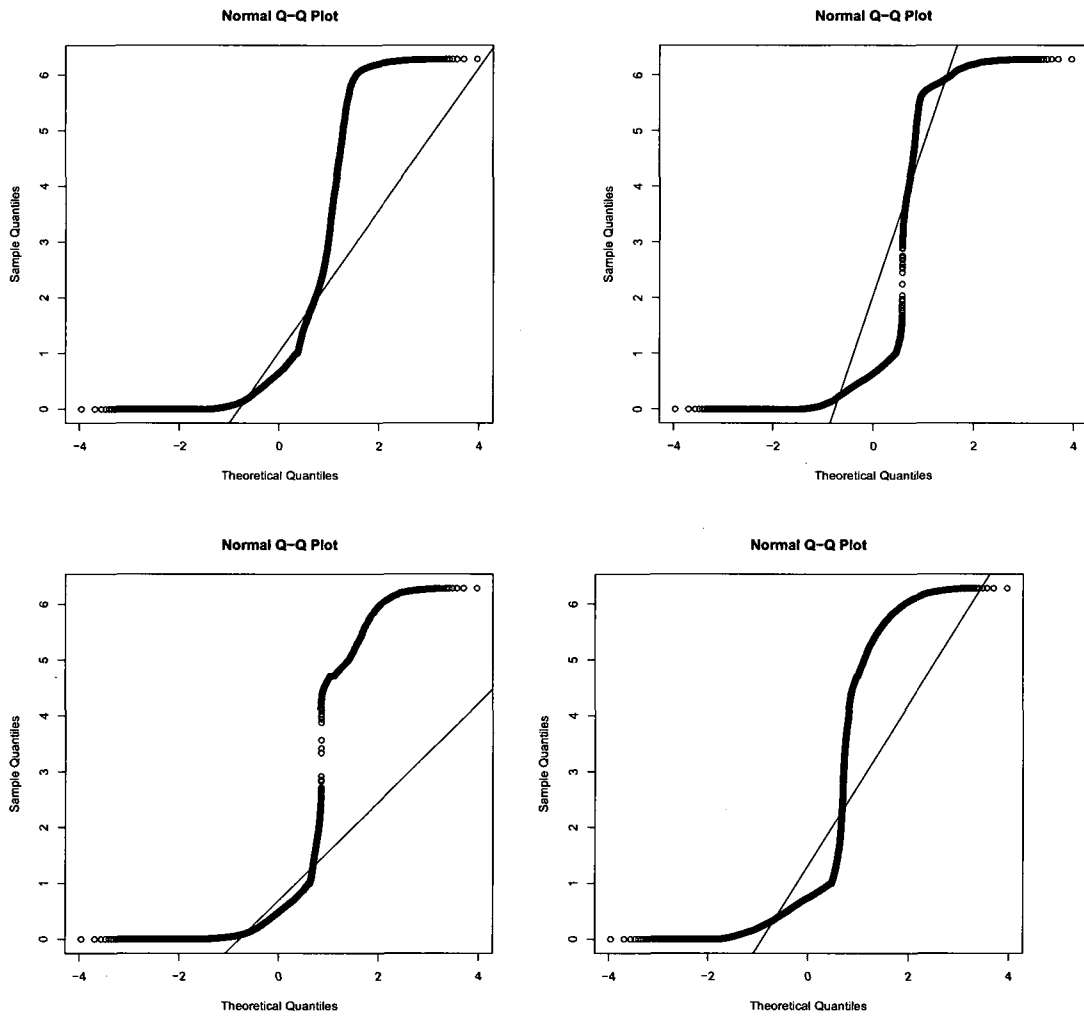


Figure 3: QQ-plots of all 17 classes of Arabic character shapes using *RC* feature representations; part 3 of 4.

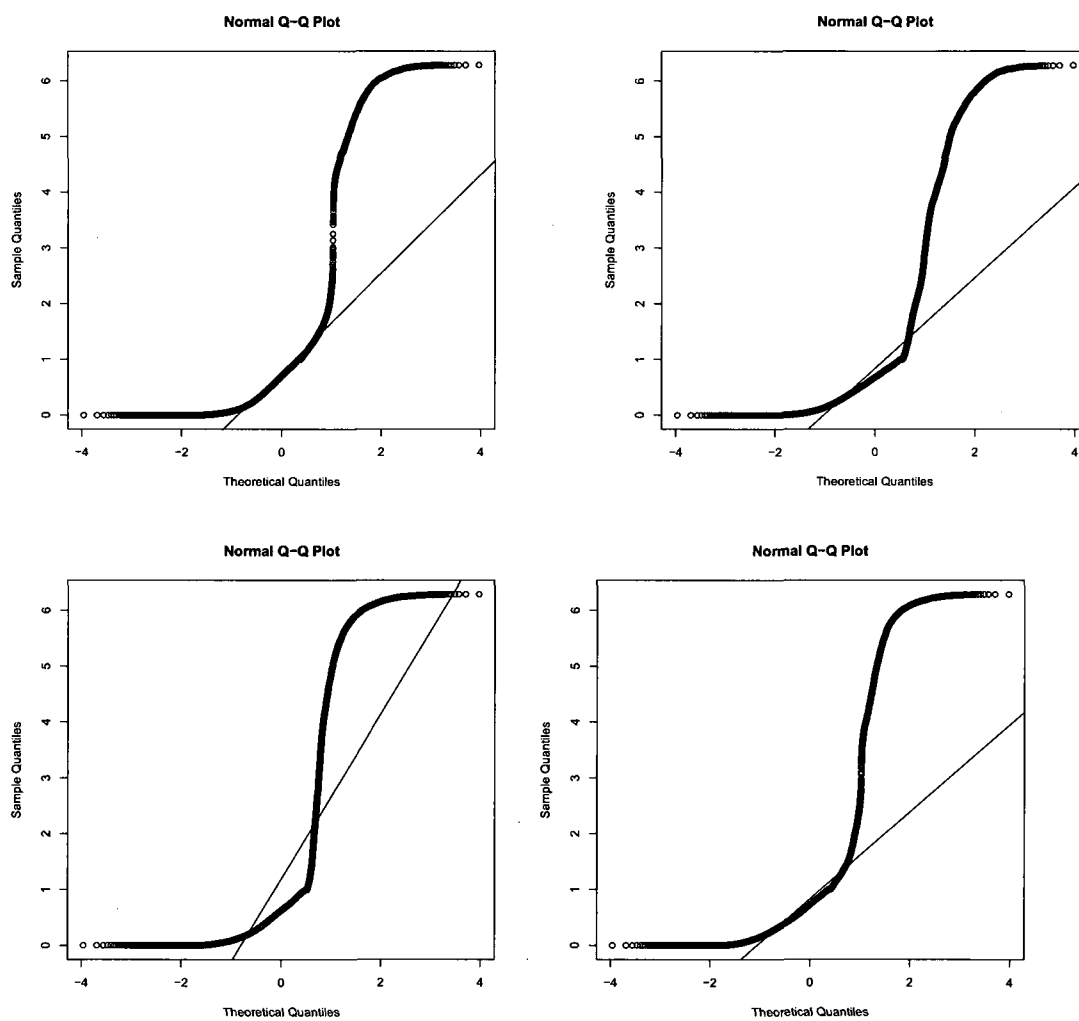


Figure 4: QQ-plots of all 17 classes of Arabic character shapes using *RC* feature representations; part 4 of 4.