Sudan University of Science and Technology
College of Graduate Studies

Online Arabic Handwriting Recognition

التعرف الآني على الكتابة العربية

A thesis submitted for the degree of Doctor of Philosophy

in Computer Science

By

Hozeifa Adam Abd Alshafy Omer

Supervisor

Prof. Izzeldin M. Osman

Co-Supervisor

Dr. Mohamed E. M. Musa

Feb 2014

Sudan University of Science & Technology

College of Graduate Studies

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

Approval Page

Name of Candidate: Hozeifa Adam Abd Alshafy Omer
حذيفة آدم عبد الشافع عمر

Thesis title: Online Arabic Handwriting Recognition
التعرف الآني على الكتابة العربية

Approved by:

External Examiner

Name: Ismail El-Azhary

Signature: ..... Date: 27.2.2014

2. Internal Examiner

Name: Eltahir Mohamed Hussein

Signature: ..... Date: 27/2/2014

Co-

1. Supervisor

Name: Mahmoud E. M. Musa

Signature: ..... Date: 5/3/2014

cgs @ sustech edu. البريد الالكتروني 83 769363 /فاكس ص.ب 407

ABSTRACT

This thesis presents a novel system approach of online Arabic handwriting recognition as well as datasets of online Arabic handwriting.

To fill the gap and shortage of standard online Arabic handwriting datasets, the thesis introduces two datasets of this handwriting. The XML representation is selected for the datasets design. The construction of the datasets is achieved by developing two software tools (collection tool, verification tool) and then using these tools to collect and maintain online Arabic handwriting for the datasets.Therefore, we have acquired standard datasets which can be used by researchers to train and test their recognition techniques for this handwriting.

These acquired datasets are used in the training and testing of the system approach which developed in this thesis. This system involves a position invariant feature extraction method. Moreover, it involves an algorithm which specifically designed in this thesis to divide the given cursive words of the handwriting into segments of Arabic letters. To fulfill the recognition, the system employs hidden Markov models so as to model the handwriting.The experiments which performed to evaluate the system have shown that the system performance is influenced by the feature arrangements within those feature vectors of the letter segments; such that the well done choice of the arrangement has enhanced the performance.

المستخلص

يقدم هذا البحث منهجية جديدة (Novel approach) لنظام التعرف الآلي على الكتابة العربية الآنية إضافة الى مجموعات بيانات (Datasets) لهذه الكتابة.

نظراً لقلة ومحدودية مصادر بيانات (Datasets) الكتابة العربية الآنية فقد تضمن هذاالبحث على إنشاء مجموعتي بيانات (Two datasets) لهذه الكتابة. لقد تم اختيار البنية اكس ام ال (XML) لتصميم هاتين المجموعتين، حيث تعتبر هذه البنية هي الأفضل والأحدث في تصميم مجموعات البيانات للكتابة اليدوية. تطلب إنشـــاء المجموعتين لتطوير أداتين برمجيتين (Two software tools): أداة لجمع الكتابة العربية الآنية وحفظها في مجموعات البيانات، واداة أخرى لمراجعة وتنقيح محتوى مجموعات البيانات. لقد أدى استخدام هاتين الأداتين الي إنشاء مجموعات بيانات ذات أحجام مناسبة بحيث يمكن إستخدامها من قبل الباحثين في تدريب واختبار التقنيات التي تطور للتعرف على هذه الكتابة.

تم استخدم مجموعات البيانات الآنفة الذكر في تدريب واختبار النظام الذي تم تطويره في هذا البحث للتعرف على الكتابة العربية الآنية. يحتوي النظام على طريقة لإستخلاص سمات (Features) لا تتأثر بمواضع واتجاهات الكتابة. إضافة الى ذلك يشتمل النظام على خوارزمية تم تصميمها في هذا البحث لتقوم بتجزئة كلمات هذه الكتابة الى مقاطع لحروف عربية. تشتمل فكرة التعرف بالنظام على إنشاء نماذج ماركوف خفية (HMMs) للكتابة التي يتم تقديمها للتعرف. بينت التجارب التي أجريت لقياس أداء النظام بأن هذا الأداء يتأثر بتنظيم وترتيب السمات المضمنة في المتجهات (Feature vectors) التي تشتمل على سمات الكتابة، حيث أن الاختيار المناسب لتنظيم وترتيب السمات يعطي اداء أفضل في التعرف على الكتابة.

ACKNOWLEDGEMENT

This thesis could not have been completed without the help, support and encouragement from several individuals.

First of all, I would like to thank my supervisor, Professor Izzeldin M. Osman, the sponsor for this thesis. Special thanks and deepest gratitude goes to my co-supervisor, Dr. Mohamed E. M. Musa,who has provided the original inspiration for this research. Once again, I am very grateful to my co-supervisor for the provision of several ideas, patience, guidance, and proofreading of my work.

I would like to extend my thanks to Dr. Nagmeldeen Abdo Mustafa at college of engineering in Sudan University for science and Technology for helping me collect most of the references on which we have made the thesis. My research would not have been possible without his helps.

I am grateful to Karary University. This thesis was financially supported by this university.

My sincere thanks also go to Dr. Abd Alrheem Sati, the antecedent head for the Department of Computer and Electrical Engineering at Karary University, for nominating me to do this thesis as a requirement for a PhD.

Special thanks go to Dr. Khalid Ahmed Ibrahim, thehead for the department of Computer Science at Karary University, and the department staff for their endless support and encouragement to complete the thesis.

During my time in Sudan University for Science and Technology, I came across many interesting people such as teaching staff, technical staff,

employees, and workers. I thank them for being friendly and positive every time Iwas in and out from the university.

I am grateful to all of my friends and colleagues, both staff and PhD students, in the College of Computer Science and Information Technology at the Sudan University for Science and Technology for their stimulating conversations, intellectual discussions, debates and hilarity during the seminars and sessions of the research.

I am grateful to all of my friends and colleagues in the College of Engineering at Karary University for their stimulating conversations.

I would like to thank all of the participants who volunteered to give up their time to offer handwriting to the datasets which documented in this thesis. Without them I would have no results.

Most importantly, I want to thank my relatives in Kharoum, Algadarif, and other cities of Sudan for their endless interest and encouragement.

I am very grateful to my sisters, Hajir and Sohair, and my brothers, Mohamed Moled and Abd Alshafy, for their emotional and moral support.

I would like to express my deepest gratitude to my wife, Suad Mohamed Mahmoud. She was always there cheering me up and stood by me through the good and bad times.

Last but not the least, I would like to thank my wonderful parents, Amna Ibrahim and Adam Abd Alshafy, for giving birth to me at the first place and supporting me spiritually throughout my life.

DEDICATION

To our nation martyrs, the symbol of sacrifice;

To my mother, who never stop giving of her unconditional emotion and support in countless ways;

To my father, who would have been happy to see the completion of the thesis and follow the science pathway;

To my wife, which her patience, support, and understanding have lightened my spirit to finish the thesis;

To my beloved brothers and sisters; particularly my dearest sister, Hajir Adam, who encourage me when the things look bleak;

To my friends and relatives, who encourage and support me;

I dedicate this thesis…

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
CBD	**C**haracters' **B**oundaries **D**etection
DTW	**D**ynamic **T**ime **W**arping
EM	**E**xpectation **M**aximization
HCI	**H**uman **C**omputer **I**nterface
HMM	**H**idden **M**arkov **M**odel
InkML	**InkM**arkup **L**anguage
ML	**M**aximum **L**ikelihood
PC	**P**ersonal **C**omputer
PDA	**P**ersonal **D**igital **A**ssistant
SUSTOLAH	**S**udan **U**niversity of **S**cience and **T**echnology **O**n**L**ine **A**rabic **H**andwriting
W3C	**W**orld **W**ide **W**eb **C**onsortium
XML	e**X**tensive **M**arkup **L**anguage
pdf	**p**robability **d**ensity **f**unction
pmf	**p**robability **m**ass **f**unction

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF RELATED CONCEPTS

Handwriting is one of those methods which have been laid within the context of the human communication approaches.There is no doubt that handwriting has taken an ad hoc prominence in human life. It has been used for a long time as an expansion for human memory, as well as communications. Through handwriting, people can conduct and record their opinions, activities, impressions, and etc. Many of old cultures and civilizations, sciences, and religious have been corroborated by the virtue of handwriting.

Actually, handwriting is a collection of artificial graphical marks on a surface; these marks are combined with mark's conventional relation which normally defined within a context of a natural language [1]. Exactly these marks represent language characters or alphabets.Each of human languages (e. g., Arabic, English, and etc.) is characterized by a specific set of characters. Beside the character set, there is a group of rules that specify the character connection forms; and by virtue of these rules, the characters are placed so as to form high level units (i.e., words).

The computer technology has brought handwriting to a new era. Because of this technology, handwriting hastaken two situations which are: On-line handwriting and off-line handwriting [1]. On-line handwriting refers to the handwriting which formed by a pen movement on the screens of Tablets, Tablet PCs, or PDAs screens; and the handwriting can be handled by computing devices at the same time of its formalization. On the other hand, the

input writing of off-line handwriting was previously prepared on papers, and then later it is completely entered to computers, by the use of scanners, as an image.In many situations, such as the notes taking by students in classrooms, we can find that a pen together with paper or a notepad is more convenient than a keyboard. Therefore, it is becoming increasingly difficult to ignore on-line handwriting.

Both online handwriting and offline handwritinghave been regarded with several types of analysis, interpretation, and recognition [1]. Handwriting recognition is a task of mapping the graphical marks of handwriting into their symbolic representations. In English and many languages based on the Latin alphabets, these symbolic representations are typically the 8-bit ASCII codes. The characters of most written languages in the world are represented in these days in the form of 16-bit Unicode. Handwriting interpretation is a task of determining the meaning of a body of handwriting (e. g., handwritten address). Handwriting identification is a task of determining the author of a sample of handwriting from a set of writers, assuming that each person's handwriting is individualistic. Signature verification is a task of determining whether or not the signature is that of a given person.Identification and verification, which have applications in forensic analysis, are processes that determine the special nature of the writing of specific writer, while handwriting recognition and interpretation are processes whose objectives are to filter out the variations so as to determine the message.Therefore, online handwriting recognition is anapplication of pattern recognition.

Actually, human beings are the best pattern recognizers in most instances. Nevertheless, there is no understanding of how humans recognize patterns. By the time they are five years old, most children can recognize digits and letters. Small characters, large characters, handwriting, and machine printed are easily recognized by the young. The machine learning specialists have done to

provide computers with this ability. Pattern recognition and machine learning are activities which can be viewed as two faces of the same field; such that pattern recognition has its origins in engineering and machine learning has emerged from computer science [2].

Automatic (machine) recognition, description, classification, and grouping of patterns are important problems in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, artificial intelligence, and remote sensing [3]. The field of pattern recognition is concerned with the automatic discovery for regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories [2]. For example, a pattern could be a fingerprint image, handwritten cursive word, human face, or speech signal. Given a pattern, its recognition or classification may consist of one of the following two tasks:

1) Supervised classification (e.g., discriminate analysis) in which the input pattern is identified as a member of a predefined class.

2) Unsupervised classification (e.g., clustering) in which the pattern is assigned to a hitherto unknown class.

The remainder of this chapter formally defines the problem statement for this thesis, states the research objectives of the thesis, describes the researchapproach, presents the thesis contributions, showsthe primary publications of the thesis, and outlines the thesis structure.

1.2 PROBLEM STATEMENT

As illustrated in Table 1.1 and Table 1.2, Arabic alphabet comprises 28 letters as well as 8 auxiliary letters. Arabic handwrittenwords are scripted horizontally from right to left in a cursive style. Most of these letters have

four different shapes depending on weather they appear at the beginning, middle, or end of the handwritten words, or on their own isolated shape.

It is widely accepted that machine recognition of online Arabic handwriting is a difficult problem. The problem is very challenging because of the Arabic writing issues. We classify these issues into three categories of difficulties: difficulties of Arabic nature, writer dependent, and machine dependent.

Difficulties of Arabic nature: This category includes the difficulties which are inherent to the nature and characteristics of the Arabic scripts (i.e., writer independent).The following list shows some of these difficulties.

- The different shapes of Arabic letters which depend on the position of the letters in the handwritten words.
- The cursive of Arabic letters which occurs in the handwriting.
- The presence of ligatures in the handwriting.

Writer dependent: This category includes those difficulties which fall in the responsibility of the writer. The difficulties involve the various writing and calligraphic styles for the same Arabic letter. In fact, every writer has an individual writing style. Furthermore, the condition and state of the writer during writing significantly influences the handwriting.

Machine dependent: This category considers those difficulties which induced by the quality of the tablets and writing screens.

Beside the above mentioned difficulties, there is a limitation for the data of online Arabic handwriting [4, 5, 6, 7, 8]. Large volumes of data (i. e., datasets) are of great significance to researches who work in the field of

pattern recognition. The data enable the field researchers to train and test their derivative recognition systems and techniques.

Table 1.1: Primary category of Arabic letters

Letter Name	**Letter Shapes**			
	Beginning	Median	End	Isolated
Alif			ـا	ا
Ba	بـ	ـبـ	ـب	ب
Ta	تـ	ـتـ	ـت	ت
Tha	ثـ	ـثـ	ـث	ث
Jim	جـ	ـجـ	ـج	ج
HHA	حـ	ـحـ	ـح	ح
KHA	خـ	ـخـ	ـخ	خ
Dal			ـد	د
Thal			ـذ	ذ
Ra			ـر	ر
Zai			ـز	ز
Sin	سـ	ـسـ	ـس	س
Shin	شـ	ـشـ	ـش	ش
Sad	صـ	ـصـ	ـص	ص
Dhad	ضـ	ـضـ	ـض	ض
Tua	طـ	ـطـ	ـط	ط
Zua	ظـ	ـظـ	ـظ	ظ
Ain	عـ	ـعـ	ـع	ع
Gain	غـ	ـغـ	ـغ	غ
Fa	فـ	ـفـ	ـف	ف
Qaf	قـ	ـقـ	ـق	ق
Kaf	كـ	ـكـ	ـك	ك
Lam	لـ	ـلـ	ـل	ل
Mim	مـ	ـمـ	ـم	م
Non	نـ	ـنـ	ـن	ن
Ha	هـ	ـهـ	ـه	ه
Waw			ـو	و
Ya	يـ	ـيـ	ـي	ي

Table 1.2: Auxiliary category of Arabic letters

Letter Name	**Letter Shapes**			
	Beginning	Median	End	Isolated
Alif Maqsurah			ـــى	ى
Alif+Mad			ـــآ	آ
Hamzah				ء
Alif+Upper Hamzah			ـــأ	أ
Alif+Lower Hamzah			ـــإ	إ
Ta-Marbotah			ـــة	ة
Waw+Hamzah			ـــؤ	ؤ
Ya+Hamzah	ئـــ	ـــئـــ	ـــئ	ئ

1.3 RESEARCHOBJECTIVES

This thesis aims to present a novel approach of online Arabic handwriting recognition as well as datasets for online Arabic handwriting. To achieve this goal, the thesis sets out four research objectives which aim to understand and improve the recognition. These objectives are:

1. Understanding the background for the development techniques of online Arabic handwriting recognition, the prior works of the handwriting recognition, and the theoretical background of hidden Markov models (HMMs).

2. Creating datasets of online Arabic handwriting. This objective will be achieved if two software tools are developed and employed to collect and maintain the handwriting.

3. Designing a segmentation algorithm which acts to address the cursive challenge of the handwriting. The algorithm breaks the online Arabic handwritten words to their constituent handwritten letters.

4. Using the knowledge which gathered by the achievement of the first objective and the algorithm of the third objective to develop a novel system approach of online Arabic handwriting recognition. The approach is evaluated by the datasets of the second objective.

1.4 RESEARCH APPROACH

The recognition of online Arabic handwriting is a task that maps the handwriting into its Arabic characters ASCII representations. The context of this thesis is dedicated to understand and improve this task of online Arabic handwriting recognition.

The literature review of this thesis shows up many difficulties and challenges regarding with the recognition of online Arabic handwriting. Furthermore, the review introduces a good understanding for the essential techniques of the recognition.

Large volumes of data are as important as recognition methods for researches who work in the field of pattern recognition. The data enable those researchers to train and test their derivative recognition systems and techniques. Unfortunately, the literature review has also shown that the area of online Arabic handwriting recognition has been suffered from a lack and limitations of these data. To overcome this problem, the research dedicates a part of the work as a construction project for online Arabic handwriting datasets. The project involves a development of two software tools as well as a design for the datasets. These tools are then used to populate the datasets with huge data of online Arabic handwriting, where more than one hundred and fifty participants –from various high educational institution in Sudan– are contributed in the collection of these data. Finally, good indications, regarding with the use of the datasets, have occurred based on the analysis of the data.

On the other side of the research, a system is implemented in MATLAB environment so as to handle those difficulties of online Arabic handwriting recognition. Many experiments, concerning with the training and recognition performance of the implemented system, are conducted. Based on the results of these experiments, a new sight for the enhancement of online Arabic handwriting recognition has been introduced.

1.5 CONTRIBUTIONS

The main contributions of this thesis can be outlined as follows.

- The creation of SUSTOLAH datasets. The datasets are constructed to be used in the field of online Arabic handwriting recognition. The field researchers can employ these datasets to test and train their derivative techniques.

- A novel algorithm, called Characters' Boundaries Detection (CBD) algorithm, for segmenting online Arabic handwritten words. The algorithm introduces a technique that addresses the cursive problem of the handwritten letters which occurred in the online Arabic handwritten words.

- An approach for online Arabic handwringing recognition. The approach employs the CBD algorithm as well as the technique of HMMs to achieve the recognition. The approach demonstrates that the recognition performance is influenced by the feature vector format of the handwritten segments; such that the well done choice of the format has enhanced the performance.

1.6 PUBLICATIONS

Two parts of the thesis have been reviewed and published in the following publications:

- H. A. Abd-Alshafy, Dr. M. E. Mustafa, "Datasets for Online Arabic Handwriting," In Proceedings of The 2nd International Conference on E-Applications in developing countries (ICEA 2011), Khartoum, Sudan, 27-29 Nov 2011.

- H. A. Abd-Alshafy, Dr. M. E. Mustafa, "Characters' boundaries based segmentation for online Arabic handwriting," In Proceedings of IEEE International Conference on Computing, Electrical and Electronics Engineering (ICCEEE13), Khartoum, Sudan, 26-28Aug. 2013.

1.7 THESIS STRUCTURE

This thesis consists of six chapters. The first chapter, Chapter 1, provides an introduction to the thesis. The theoretical background and priori work of the research are respectively described in chapter2 and chapter 3. Chapter 4 describes the research part for creating the datasets; moreover, chapter 5 involves the research part which introduces the system approach of the recognition. The conclusion of the thesis is given in Chapter 6. The thesis is structured as follows.

Chapter 2 goes through theoretical background for the typical system structure of online Arabic handwriting recognition, HMMs, and representation methods of online handwriting datasets.

Chapter 3 presents a review for influential and current research studies of online Arabic handwriting recognition.

Chapter 4 describes the design of two datasets, named as SUSTOLAH, for online Arabic handwriting; presents two software tools which developed to construct the datasets; and shows several key conclusions of these datasets.

Chapter 5 presents the construction of a novel system approach for online Arabic handwriting recognition; furthermore, it describes the substantial performance findings of this approach.

Chapter 6 draws the conclusions and future work suggestions for the thesis.

CHAPTER 2

BACKGROUND

2.1 INTRODUCTION

This chapter goes through theoretical background for the typical system structure of online Arabic handwriting recognition, HMMs, and representation methods of online handwriting datasets.

Figure 2.1 shows that data acquisition, preprocessing, segmentation, feature extraction, classification, and post processing are the most common stages which form the system structure of online Arabic handwriting recognition [9]. The existence of these stages is not mandatory in the system structure (i.e., the recognition task can be achieved without the complete set of these stages). Therefore, some of the stages are integrated into or passed up others stages [8, 10].

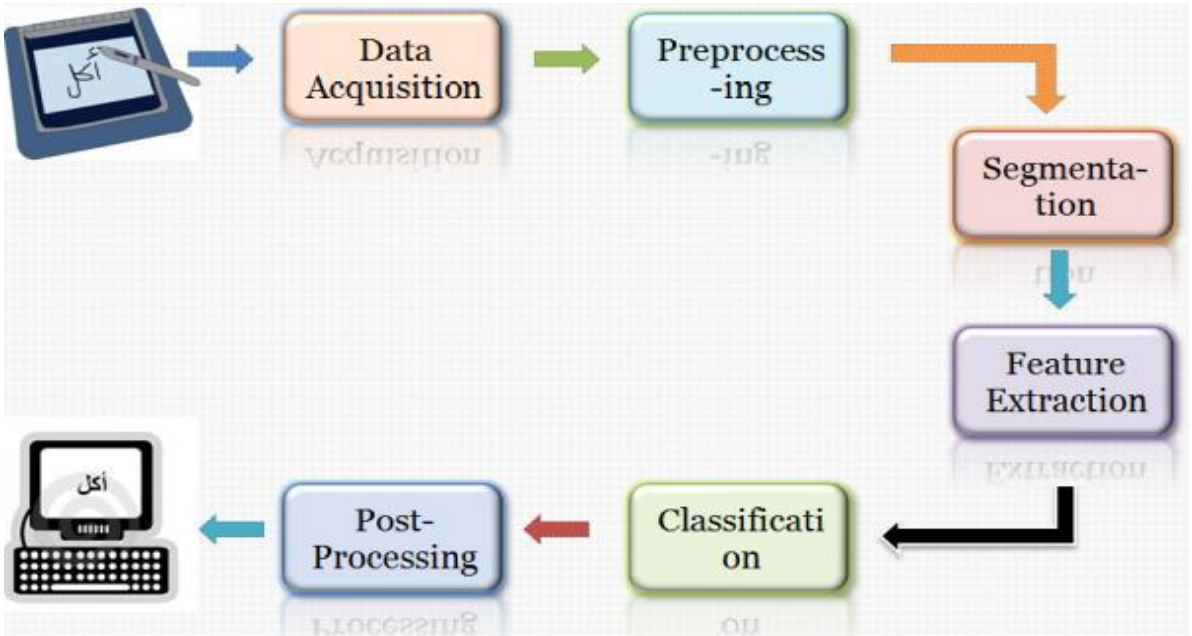

Figure 2.1: System stages of online Arabic handwriting recognition. [9]

As an approach of artificial intelligence, HMMs can be applied in many applications of pattern recognition (e.g., speech recognition, online handwriting recognition, gesture recognition, motion video analysis and tracking, and etc.) where the goal is to recover a data sequence that is not immediately observable [23, 39, 40].

There are many methods which have been used as structure formats for online handwriting datasets [12, 60]. UNIPEN format, which established in the last decade of the 20's century, was the first of these methods [54]. Stefan Jaeger and Masaki Nakagawa created two databases, based on UNIPEN, of online handwritten Japanese characters [12]. XML-based representations of online handwriting were developed to replace the UNIPEN format. They successfully address the shortcomings of UNIPEN.

For the rest of this chapter, section 2.2 explores the typical system structure of online Arabic handwriting recognition; section 2.3 provides a backgroud for HMMs; and section 2.4 presents the representation methods of online handwriting datasets. The chapter summary is presented in section 2.5.

2.2 SYSTEM STRUCTURE OF ONLINE ARABIC HANDWRITING RECOGNITION

Sections 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5, and 2.2.6 below explore the stages which form the typical structure of online Arabic handwriting recognition.

2.2.1 Data Acquisition

In fact, data are required for the training and testing phases of pattern recognition systems. These data are available in the form of datasets [11, 12]. Therefore, the developers of online Arabic handwriting recognition systems

are interested in the datasets which support the developers with the data of online Arabic handwriting.

On-line handwriting, whether it is for Arabic or other languages, is obtained by the move of pens on the screens of Tablets, Tablet PCs, or PDAs screens; and it can be handled by computing devices at the same time of its formalization [1].

2.2.2 Preprocessing

The preprocessing stage follows the data acquisition in the task context of the recognition system. The input data are processed, in this stage, in order to facilitate the remaining task of the recognition system. Usually this stage addresses the problems of data reduction, elimination of imperfections, and normalization.

Most digitizers perform uniform temporal sampling when they used by writers to script the handwriting. Often this sampling results an oversampling of slow pen motion regions and under-sampling of fast pen motion regions [13]. Nowadays, online capturing devices, even the cheaper ones, produce fairly smooth signals; hence, the preprocessing on signal noise, such as re-sampling, is not a crucial component for the recognition systems [14]. There are many techniques which can be used in the preprocessing stage. These are:

- **Size normalization:** This technique characterizes the size independent recognition systems. It supports the systems with the ability to recognize the online handwriting whether the handwritingis in small or large styles.
- **Translation:** This technique computes the image's center of gravity, and then it translates the image such that its origin becomes the center of gravity.

- **Re-sampling**: This technique normalizes the speed of handwriting. It performs the normalization by re-sampling the point sequences and distributing the points uniformly over a sampled curve. Based on the speed of handwriting, the acquired pen tips are distributed unevenly along the handwriting trajectory, there will be less tips in under-sampling where the speed is high and more tips in oversampling where the pen motion is slow.

- **Smoothing and noise elimination**: The input data provided by tablets may contain a considerable amount of noises which complicates the work in the next stages. These noises are usually caused by the used digitizers as well as shaking hands [15]. Usually the technique is used to eliminate the duplicated date points that by forcing a minimum distance between consecutive points. For example, a weighted averaging over the range [i—k, …, i+k] was used in [15]; moreover, Douglas and Peucker's algorithm was adopted to simplify the point sequences in [7].

2.2.3 Segmentation

The segmentation stage takes a great importance especially in the handwriting recognition systems for cursive scripts such as Arabic [16]. Segmentation is the process of dividing an input handwriting into smaller units such as words, sub words, letters, or strokes.

Sometimes the segmentation stage could be eliminated from the recognition task. Therefore, there are two main approaches for the recognition due to this stage: Holistic approach and analytical approach. The holistic approach doesn't include a segmentation stage. The approach deals with an input word as a whole [13]. Therefore, it avoids errors which are generated by the use of inefficient segmentation algorithms [13]. However, the use of the holistic approach is impractical, because the approach requires training for

each word in the script [13]. Holistic based recognition systems are susceptible to achieve low recognition rate when they employed for scripts of large vocabulary such as Arabic; whereas these systems may achieve high recognition rates when they used for the scripts of small vocabulary [17, 18]. The analytical approach acts to divide the input curve, which represents a handwritten word, into individual characters or strokes, where in turn they are recognized and assembled to identify the handwritten word [13, 19]. The approach has an advantage that it requires small set of trained models in order to handle a large vocabulary, whereas this set may contain one model for each letter [19].

Actually, segmentation of handwritten words is faced with some problems such as the absence of consistent baselines, large variations of in handwritten styles, and the vague ligatures between the connected letters [13]. Moreover, there will always be missing segmentation pointsdue to uncertain handwriting.

In 1984, A. Belaid and J. P. Haton published a paper in which they described a direction based segmentation approach [37]. The approach considers the first three handwriting points: P1, P2, and P3 which are exactly refer to C[1], C[2], and C[3] respectively. These points mark the first three specks in a handwritten word .The angle between P1, P2 and P2, P3 and the length of P1, P2 are calculated. If both the angle and the length are larger than specified threshold values, the point P2 is considered as a pivot or segment point; otherwise the value in C[3] is assigned to P2 and that in C[4] is assigned to P3. The process is repeated until all the segment points in the word have been found. Figure (2.2) depicts the segmentation process.

After each finding of a segment point P2, every segment P1 P2 is categorized into one of four directions. For example, if a segment angle (the angle between the line P1 P2 and the positive X-axis) lies between 45 and 135,

then the segment will be of type 1. Figure 2.2 illustrates the directions into which segments are categorized. Adjacent segments having the same direction codes are merged. However, if adjacent direction codes indicate reversal directions, such as (1, 3), (3, 1), some modifications have to be made because such reversals are very variable in handwriting characters. The modification is achieved by inserting a short perpendicular section to that code, so the code (1, 3) is changed to (1, 2, 3) or (1, 4, 3). Also the direction of each segment is recorded, that by taking the tangent of the angle between the segment and its direction code. Figure (2.3) and (2.4) depict the direction categories and the tangent diagram for the segments.

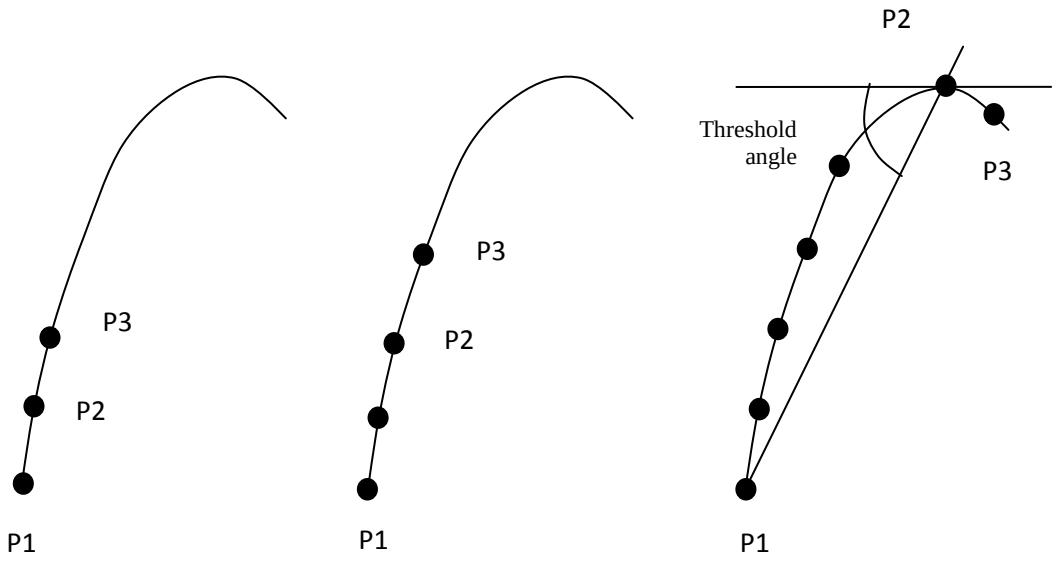

Figure 2.2: Illustration for the direction based segmentation approach. [4]

The key approaches of the segmentation can be listed as follows:

- **Segmentation based on extreme points**

  The extreme points lead to identify ascenders and descenders in Arabic handwriting such as the medial alif (ا), laam (لـ), and the final raa (ـر).

- **Segmentation based on directions**

  This segmentation computes the slope of tangent line for each point in the handwriting. Each of the handwriting points will be considered as a segmentation point, if the slope of its tangent exceeds the amount of specific threshold.

- **Segmentation based on intersection points**

  The intersection points are points in which a handwritten stroke intersects itself.

- **Segmentation based on pen movements**

  As described previously in this section, this segmentation method considers the maximum and the minimum points for signal of the curvilinear velocity for the handwritten strokes.

- **Merging of the previous methods**

  This method identifies segmentation points by applying all the previous mentioned methods, and then gives highest weight to nearly intersection points.

- **Genetic algorithm segmentation**

  Genetic algorithms enact a class of optimization and search methods. The function of the algorithms embodies a generation-by-generation development of possible solutions, as well as a selection scheme that permits elimination of bad solutions and replications of good solutions.

- **Manual (Explicit) segmentation**

  Using this method, segmentation points are manually placed in the horizontal segments of the handwriting.

Dynamic time warping

Dynamic time warping (DTW) is a mathematical matching algorithm. It acts to find the best pattern chain that match an input handwriting [8]. The method implicitly places segmentation points in the input when it finds a best pattern for each part of the input. Dynamic programming is involved in the method so as to find the best chain in a reasonable amount of time [15].

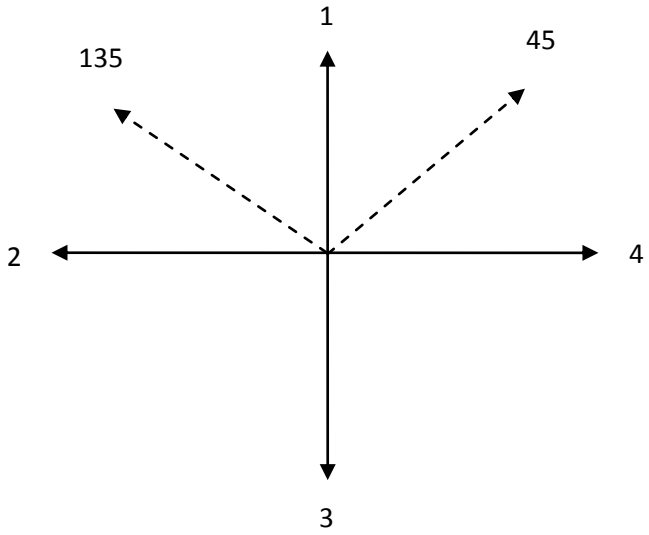

Figure 2.3: Direction categories for segments. [4]

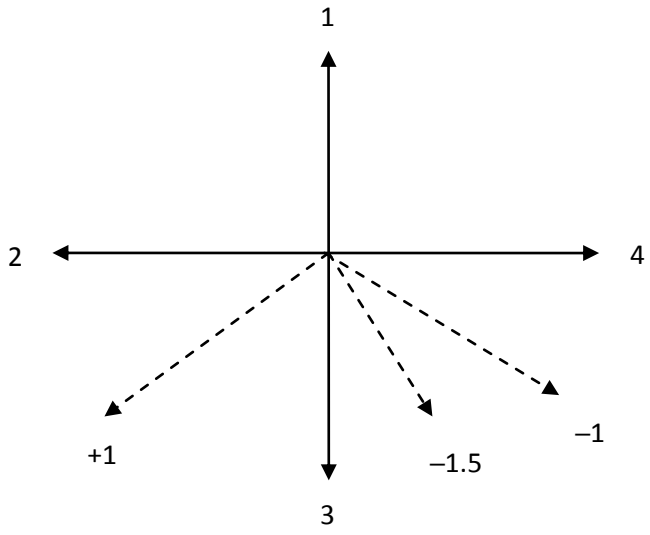

Figure 2.4: Segment tangent diagram. [4]

2.2.4 Feature Extraction

The stage of feature extraction acts to extract relevant features from the raw data of online handwriting. Since it is useless to enter these data directly in the recognition phase, feature extraction is required so as to achieve effective and accurate recognition [5].

Many features can be extracted from online handwriting, where most of these features have been employed in various recognition systems of online Arabic handwriting [4, 5, 7, 8, 18, 21, 22]. In fact, handwriting recognition should be performed based on a selected class of these features, such that this class should be small and capable to efficiently achieve the recognition. The trade-off between the selected features and their computational requirements should carefully be handled [21]. The next paragraphs of this section describe some classes of these features.

Freeman code is the simplest and common of these feature classes [9]. It has eight directions where each direction has a specific code (see figure 2.5). By the use of this code, each segment in a handwritten stroke is attributed to one Freeman direction. The final representation of the handwritten stroke will be a vector of codes which form the stroke (see figure 2.6).

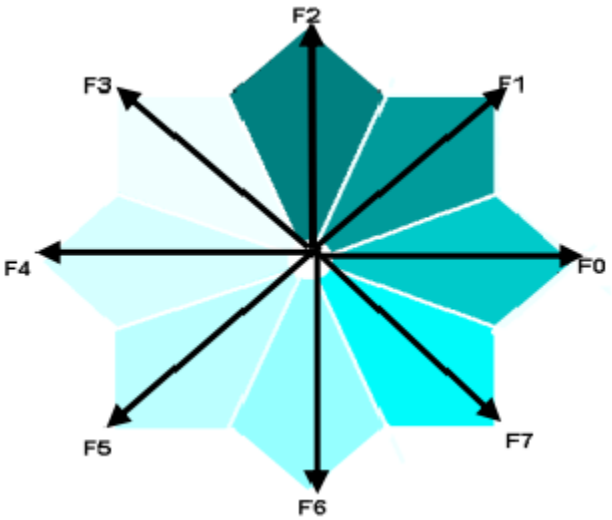

Figure 2.5: Freeman Code. [8]

Local point features are the finest features of grains [69]. They figure a feature class which has been used in the modeling of online Arabic handwriting [8]. The following list introduces some examples of these features.

1. Coordinate series: The X−Y coordinates of the points that constitute the stroke.

2. Tangent angles series: The angles of the line segments that constitute the stroke (see figure 2.7).

3. The winding value: The algebraic sum of direction changes. It is almost 0 for straight or S-shaped strokes and almost 360 for O-shaped ones.

4. The Aspect Ratio: The height to the width ratio.

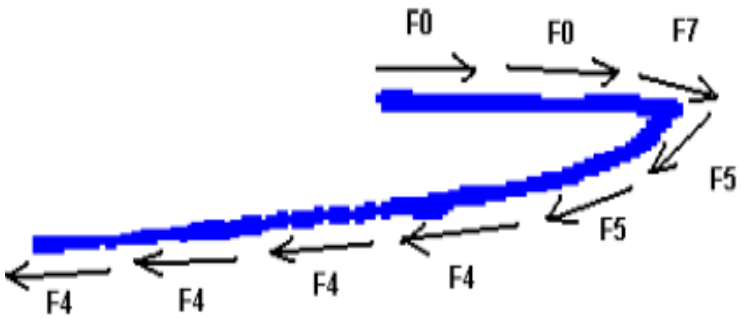

Figure 2.6: Applying the Freeman code to a handwritten stroke. [8]

Dynamic representations have been presented as features of online Arabic handwriting [7]. These features are representations of X(t) and Y(t) components for the pen positions which constitute the handwriting. Figures 2.8 and 2.9 have respectively depicted the pen positions and the dynamic representation for the online handwriting of the (س) character.

Online handwriting can be represented by the motor models of the handwriting. Here, the handwriting is represented as a signal which resulted

from superimposition of the curvilinear velocity of the handwriting strokes [17]. Thus, when a simple stroke is entered, a curvilinear velocity was produced as follows.

$$V_\sigma(t) = \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} \qquad (2.1)$$

$$V(t) = \sum_{i=1}^{n} V_i\,(t - t_i) \qquad (2.2)$$

where equation 2.1 shows the formula which calculates the curvilinear velocity of the strokes and equation (2.2) shows that the generation of a complex trajectory pattern results from an algebraic addition of stroke velocity terms.

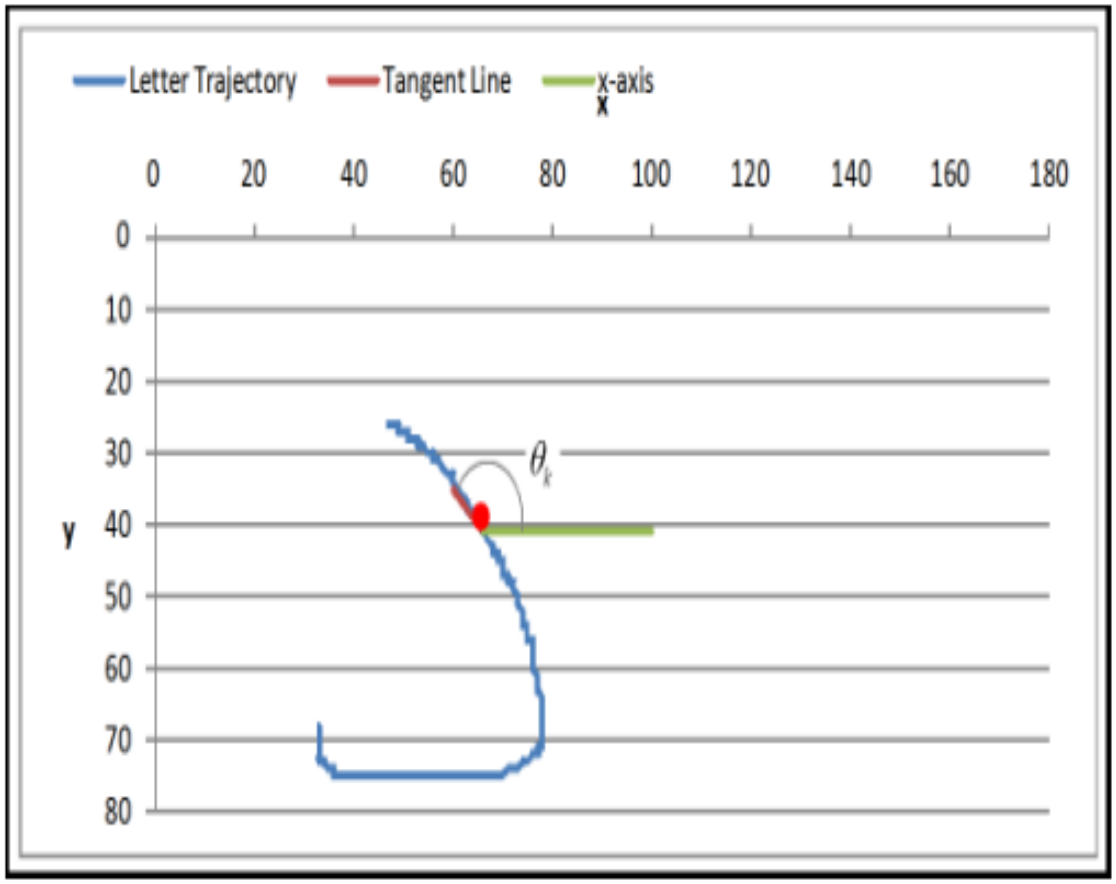

Figure 2.7: Illustration of an angle for one of the line segments which constitute the handwritten letter "ב". [8]

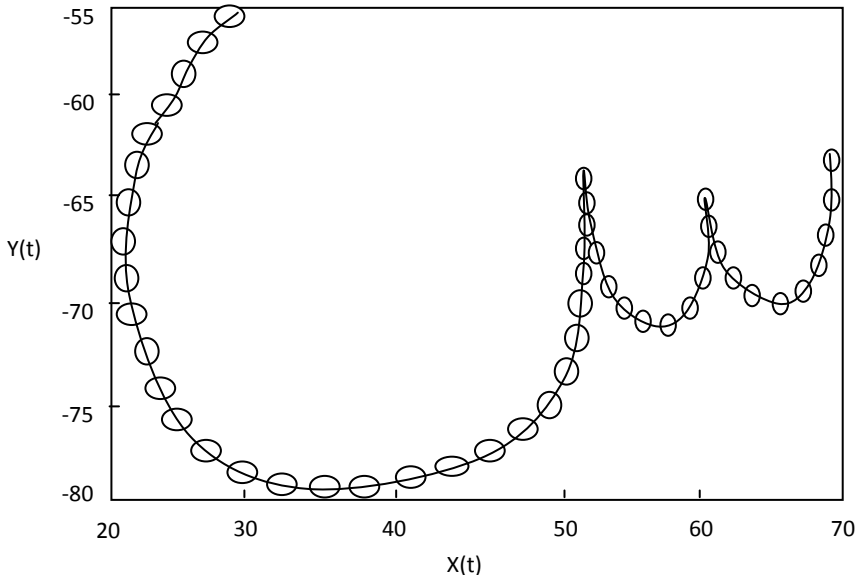

Figure 2.8: Pen positions for the online handwriting of the letter "س". [7]

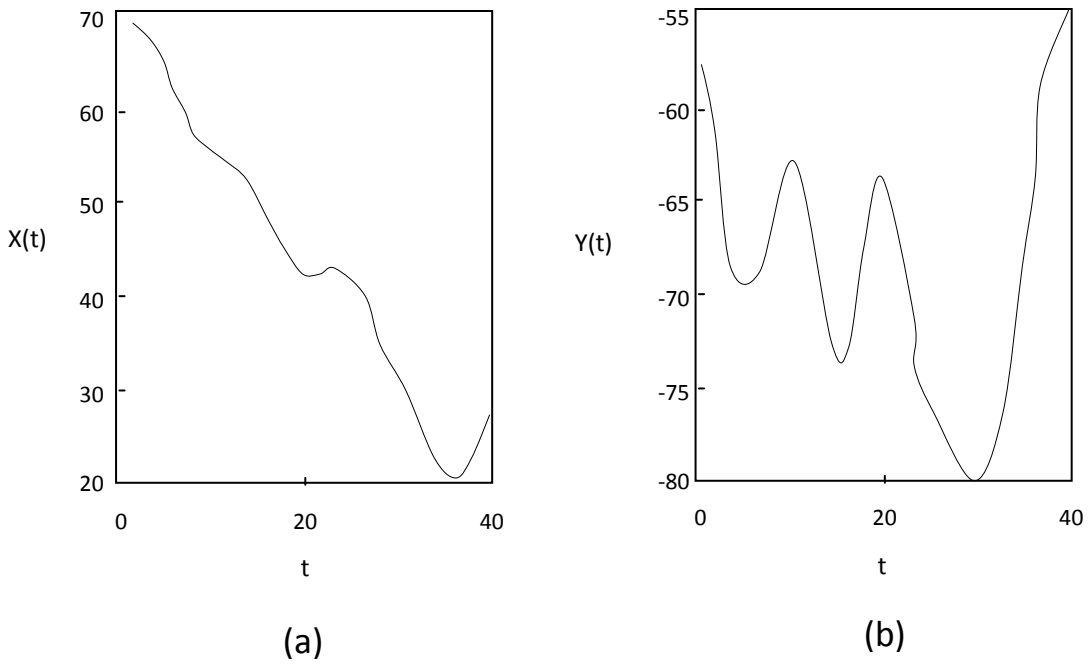

Figure 2.9: Illustration for the dynamic representation of the handwritten for the letter "س". Part (a) of the figure shows the representation of the x-coordinate for the handwriting. Part (b) of the figure shows the representation of the y-coordinate for the handwriting. [7]

2.2.5 Classification

The classification stage is the essential stage in the recognition. It maps the features, which extracted from online Arabic handwriting, into a set of finite categories or classes. These categories may either be words, sub words, or letters in Arabic language. Pattern recognition has many methods to address the issues of classification [2, 23]. The following are some of such methods.

- **Statistical pattern recognition (e.g., HMMs)**

  Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. Using this approach, the problem is posed in probabilistic terms. HMMs will be detailed in Section 2.3.

- **Structural pattern recognition**

  The approaches of structural pattern recognition assume that pattern structure is quantifiable and extractable so that structural similarity of patterns can be assessed [23]. Typically, these approaches formulate hierarchical descriptions of complex patterns, where the descriptions are built up from simpler primitive elements.

- **Syntactic pattern recognition**

  Syntactic pattern recognition is an approach which acts to apply the methods of mathematical linguistics to pattern recognition [24, 25]. This approach parses the set of features, which discovered in the feature extraction stage, using some type of grammar. If this parsing is successfully achieved (i.e. the whole input can be reduced to the grammar start symbol S) then a pattern is recognized (an object, a set of objects, etc.).

- **The application of fuzzy sets**

  Pattern recognition using fuzzy sets is a technique for determining the non-linear transfer functions that map features into classes [23]. These transfer functions enable the HCI designer to develop systems that can find patterns in the data which obtained from input sensors, and then maps these patterns to specific actions which can control the operation of the computer.

  Crisp sets are the sets which have been used in most of humans' life. In a crisp set, an element is either a member of the set or not. Fuzzy sets, on the other hand, allow elements to be *partially* in a set. Each element is given a degree of membership in a set. This membership value can range from 0 (not an element of the set) to 1 (a member of the set).

- **Neural networks**

  Neural networks are computational models which attempt to mimic the learning function of the brain. They are composed of a set of neurons or processing units which are connected together by means of connecting weights. These networks are structured so as to learn by continuous adjustment of the weights of the connections. [66]

  Pattern recognition is an important application of neural networks. Pattern recognition can be implemented by using a trainable feed-forward neural network. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. [67]

- **Approximate reasoning approach to pattern recognition**

  Approximate reasoning approach to pattern recognition consists of linguistic rules tied together by means of two concepts: fuzzy implications and a compositional rule of inference [26].

- **A logical combinatorial approach to pattern recognition**

  This approach, as the statistical approach, works with the descriptions of objects [27].

- **Support vector machines (SVMs)**

  SVMs are a type of pattern classifier which based on a statistical learning technique. Unlike Neural Networks, which minimize the empirical training error, SVMs aim at minimizing an upper bound of the generalization error through maximizing the margin between the separating hyper plane and the data [28]. Since SVMs are known to generalize well even in high dimensional spaces under small training sample conditions and have shown to be superior to traditional empirical risk minimization principle employed by most of neural networks, they have been successfully applied to a number of applications such as face detection, verification, object detection and recognition, handwritten character and digit recognition, text detection and categorization, speech and speaker verification, information and image retrieval, and prediction.

- **Using higher-order local autocorrelation coefficients to pattern recognition**

  This method uses higher order autocorrelation functions for pattern recognition [29]. The autocorrelation feature vectors reside in a high dimensional space, such that their computing can easily be avoided.

- **A novel method and system of pattern recognition using data encoded as Fourier series and Fourier space**

  This method anticipates an ensemble neurons signal processing as a unit and intends to simulate aspects of the brain. Therefore, it brings

capabilities like pattern recognition and reasoning which have not been produced with other approaches such as neural networks. [29]

2.2.6 Post Processing

The role of the post processing stage is to increase the recognition accuracy and to produce more meaningful results. Some effort have been made to correct the Arabic words which obtatined by the classification stage. In most cases, dictionaries of high frequently Arabic words are used to return the words which are closest to the results of the classification stage.

2.3 HMMs

HMMs enact a class of stochastic techniques which in turn figure a sort of uncertainty reasoning methods [23, 39, 40].

There are some situations of reasoning where conclusions can be derived from completely and explicitly defined data. Usually, predicate calculus provides the inference procedures for these situations [23]. The model of reasoning used in predicate calculus is based on sound inference rules which will produce new guaranteed correct conclusions from correct premises. In addition to these situations, there are other situations of reasoning where information might be missing, incomplete, changing, or incomplete. People do this reasoning very successfully in almost every aspect of their daily life. They deliver correct medical diagnosis and recommend treatment from ambiguous symptoms; they analyze problems with their cars or stereos; they comprehend language statements that are often ambiguous or incomplete; they recognize friends from their voices or their gestures; and so on. This type of reasoning is referred to as *abductive* and *uncertainty reasoning*.

Stochastic approaches figure a class of several ways for managing *uncertainty reasoning*. They employ the probability theory to determine the

chances of events occurring from a priori argument, as well as how combinations of events are able to influence each other [23].

HMMs are invented as an extension of Markov models which sometimes called Markov chains [40]. An HMM is a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols [39]. HMMs have two stochastic processes: one is hidden and the other is clear; whereas Markov models have one clear stochastic process.

2.3.1 Elements of HMMs

HMMsare characterized by the following:

1) N, the number of states in the model. Although the states are hidden, for many practical applications, there is often some physical significance attached to the states or to sets of states of the model. Generally the states are interconnected in such a way that any state can be reached from any other state. The individual states are dented as $S = \{s_1, s_s, \cdots, s_N\}$, and the state at time $t$as q_t.

2) M, the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled. The symbols are dented as $V = \{v_1, v_s, \cdots, v_M\}$.

3) The state transition probability distribution $A = \{a_{ij}\}$ which will be formed as follows.

$$a_{ij} = P\left(q_{t+1} = S_j \middle| q_t = S_i\right), \quad 1 \leq i, j \leq N. \qquad (2.3)$$

For special cases where any state can reach any other state in a single step, $a_{ij} > 0$ for all i, j. For other types of HMMs, it is found that $a_{ij} = 0$ for one or more (i, j) pairs.

4) The observation symbol probability distribution in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P\left((v_k att)\middle| q_t = S_j\right),$$
$$1 \le j \le N, \quad 1 \le k \le M. \qquad (2.4)$$

5) The initial state distribution is defined as follows.

$$\pi_i = P(q_1 = S_i), \qquad 1 \le i \le N. \qquad (2.5)$$

Given appropriate values of N, M, A, B, and π, the HMM can be used as a generator to give the following observation sequence

$$O = O_1 O_2 \cdots O_T \qquad (2.6)$$

where each observation O_t is one of the symbols from V, and T is the number of observations in the sequence. The following is the procedure which generates the sequence.

1. Choose an initial state $q_1 = S_i$ according to the initial state distribution π.
2. Set $t = 1$.
3. Choose $O_t = v_k$ according to the symbol probability distribution in state S_i (i.e., $b_i(k)$).
4. Transit to a new state $q_{t+1} = S_j$ according to the state transition probability distribution for state S_i (i.e., a_{ij}).

5. Set $t = t + 1$; return to step 3 if $t < T$, otherwise terminate the procedure.

The above procedure can be used as both a generator of observations and a model for how a given observation sequence was generated by an appropriate HMM.

It can be seen from the above discussion that a complete specification of an HMM requires specification of two model parameters ($N$and M), specification of observation symbols, and the specification of the three probability measures $A, B,$ and π. For convenience, the notation

$$\lambda = (A, B, \pi) \qquad (2.7)$$

is used to indicate the complete parameter set of the model.

2.3.2 HMM Computation Problems

There are three basic problems of interest that must be solved for the model so as to be useful in real-world applications. These problems are:

Problem 1: Given the observation sequence $O = O_1 O_2 \cdots O_T$and a model $\lambda = (A, B, \pi)$, how does the probability of the observation sequence$P(O|\lambda)$ is efficiently computed?

Problem 2: Given the observation sequence $O = O_1 O_2 \cdots O_T$and the model λ, how to choose a corresponding state sequence $Q = q_1 q_2 \cdots q_T$ which is optimal in some meaningful sense (i.e., best "explains" the observations)?

Problem 3: How the model parameters $\lambda = (A, B, \pi)$ are adjusted so as to maximize $P(O|\lambda)$?

Solution to Problem 1

Problem 1 is referred to as evaluation problem. The problem can be viewed as one of scrolling how well a given model matches a given

observation sequence. There are three ways for solving problem 1. These ways are:

1) The way of summing over all possible sequences of hidden states.

2) Forward procedure.

3) Backward procedure.

The way of summing over all possible sequences of hidden states: It is the most straightforward way. Consider one such fixed state sequence

$$Q = q_1 q_2 \cdots q_T \qquad (2.8)$$

where q_1 is the initial state. The probability of the observation sequence O for the above state sequence is

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) \qquad (2.9a)$$

Assume that there is statistical independence of the observations. Thus, the probability of the observation sequence O for the state sequence is

$$P(O|Q,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T) \qquad (2.9b)$$

The probability of the state sequence Q is

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \qquad (2.10)$$

The probability that O and Q occur simultaneously (i.e., the joint probability of O and Q) is the product of the above two terms, such as

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda)$$
$$= \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} b_{q_3}(O_3) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \qquad (2.11)$$

Eventually, the probability that O (given the model) is obtained by summing this joint probability over all possible state sequences

$$P(O|\lambda) = \sum_{all\ Q} P(O|Q,\lambda)P(Q|\lambda) \qquad (2.12)$$

Given N possible states, the number of distinct sequences of T states is N^T. Moreover, there are about $2T$ calculations in the above formula for each of these sequences. Therefore, the calculation of $P(O|\lambda)$, using the above formula, is in the order of $2T \cdot N^T$. Thus the way of summing over all possible sequences of hidden states is intractable.

Forward procedure: This procedure relies on a forward variable $\alpha_t(i)$ which containing the probability of being in state at time t (i.e., $q_t = S_i$) having observed the sequence $O_1 O_2 \cdots O_t$

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i|\lambda) \qquad (2.13)$$

The procedure computes the forward variable inductively as follows:

1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N. \qquad (2.14)$$

2) Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(O_{t+1}), \quad 1 \le t \le T-1,$$

$$1 \le j \le N. \qquad (2.15)$$

3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad (2.16)$$

The complexity of the procedure is $T \cdot N^2$.

Backward procedure: This procedure relies on a backward variable $\beta_t(i)$ that denotes for probability of the partial observation sequence from $t+1$ to the end, given state S_i at time t and the modelλ.

$$\beta_t(i) = P(O_{t+1}O_{t+2}\cdots O_T|q_t = S_i, \lambda) \qquad (2.17)$$

The procedure inductively solves for the backward variable as follows:

1) Initialization:

$$\beta_T(i) = 1, \qquad 1 \le i \le N. \qquad (2.18)$$

2) Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1})\beta_{t+1}(j),$$

$$t = T-1, \qquad T-2, \qquad \cdots, 1, \qquad 1 \le i \le N. \qquad (2.19)$$

As in the forward procedure, the complexity of the backward procedure is $T \cdot N^2$.

The backward procedure is the counterpart of the forward procedure, when going back in time. It will be used for parameter learning.

Solution to Problem 2

Problem 2 is called a decoding problem. Unlike problem 1 for which an exact solution can be given, there are several possible ways of solving problem 2, namely finding the "optimal" state sequence associated with the given observation sequence. The difficulty lies with the definition of the optimal state sequence, because there are several possible optimality criteria. The most

widely used of these criteria is the criterion which finds the single best state sequence (path). To find single best state sequence means to maximize $P(Q|O,\lambda)$ which is equivalent to maximizing $P(Q,O|\lambda)$. The formal technique for finding the single best state sequence is called the Viterbi algorithm.

Viterbi algorithm: To find the single best state sequence, $Q = \{q_1 q_2 \cdots q_T\}$, for the given observation sequence $O = \{O_1 O_2 \cdots O_T\}$; the following quantity should be defined.

$$\delta_t(i) = max_{q_1 q_2 \cdots q_{t-1}}\big(P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t|\lambda)\big) \qquad (2.20)$$

$\delta_t(i)$ is the highest probability along a single path, at time t, which accounts for the first t observations and ends in state S_i. The quantity is drown by induction as follows

$$\delta_{t+1}(j) = \big[max_i\big(\delta_t(i)a_{ij}\big)\big] \cdot b_j(O_{t+1}) \qquad (2.21)$$

An array $\psi_t(j)$ is used to keep track the arguments of the best state sequence. Thus, the complete algorithm for finding this sequence can be stated as follows:

1) Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N. \qquad (2.22a)$$

$$\psi_t(i) = 0. \qquad (2.22b)$$

2) Recursion:

$$\delta_t(j) = \big[max_{1\le i\le N}\big(\delta_{t-1}(i)a_{ij}\big)\big] \cdot b_j(O_t), \quad 2 \le t \le T,$$

$$1 \le j \le N. \qquad (2.23a)$$

$$\psi_t(i) = argmax_{1\le i\le N}\left[\delta_{t-1}(i)a_{ij}\right], \qquad 2 \le t \le T,$$

$$1 \le j \le N. \quad (2.23b)$$

3) Termination:

$$P^* = max_{1\le i\le N}[\delta_T(i)] \qquad (2.24)$$

$$q_T^* = argmax_{1\le i\le N}[\delta_T(i)] \qquad (2.25)$$

4) Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \cdots, 1. \qquad (2.26)$$

Solution to Problem 3

Problem 3 is most difficult problem of HMMs. The problem is to determine a method to adjust the model parameters (A, B, π)so as to maximize the probability of observation sequences given the model. There is no known way to analytically solve for the model which maximizes the probability of the observation sequences. In fact, given any finite set of observation sequences as training data, there is no optimal way of estimating the model parameters which maximize the probability of the sequences. However, a model $\lambda = (A, B, \pi)$ can be chosen such that $P(O|\lambda)$ is locally maximized using an iterative procedure such as

1) Baum-Welch method which equivalently called expectation maximization(EM) method [44].
2) Gradient techniques [45].

The following section describes the EM method.

2.3.3 EM Algorithm

The EM algorithm is a general method for finding the maximum-likelihood estimate of underlying distribution parameters from a given data

set, such that the data is incomplete or has missing values [68]. There are two main applications of the EM algorithm:

- The first application occurs when the data indeed has missing values, due to problems with or limitations of the observation process.
- The second application occurs when optimizing the likelihood function is analytically intractable but it can be simplified by assuming the existence of values for additional (*missing* or *hidden*) parameters. This second application is more common in the computational pattern recognition community.

Let Y denotes the sample space of the observations, and let $y \in \mathbb{R}^m$ denotes an observation from Y. Let X denotes the underlying state space (i.e., space of hidden states) and let $x = \mathbb{R}^n$ be an outcome from$X$, with $m \leq n$. The data x is referred to as *complete data*. The complete data is not observed directly, but only by means of y, where $y = y(x)$, and $y(x)$ is many-to-one mapping from X to$Y$. An observation y determines a subset of X, which is dented as $X(y)$ (see figure 2.10). [68]

The probability density function (pdf) of the complete data is $f(x|\theta)$, where $\theta \in \Theta \subset \mathbb{R}^r$ enacts the set of parameters. (The reference *density* of the random variables is used for convenience, that even for discrete random variables for which *probability mass function* (pmf) would be appropriate. Subscripts indicating the random variable are pressed, with the argument to the density indicating the random variable). The pdf f is assumed to be a continuous function of θ and appropriately differentiable [68]. The maximum likelihood (ML) estimate of θ is assumed to lie within the region Θ. The pdf of the incomplete data is

$$p(y|\theta) = \int_{X(y)} f(x|\theta)dx \qquad (2.27)$$

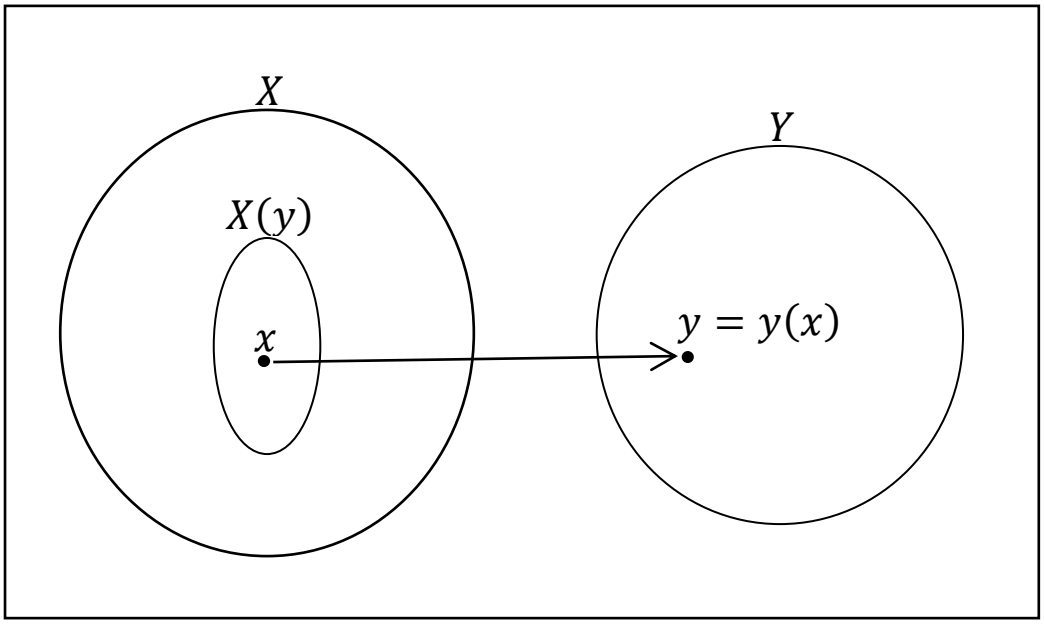

Figure 2.10: Illustration of many-to-one mapping from X to Y. The point y is the image of x, and the set $X(y)$ is the inverse map of y. [68]

The function $p(y|\theta)$denotes the likelihood of y given θ. The function is thought of as a function of the parameters θ where the data y is fixed. Let $L_y(\theta) = \log p(y|\theta)$ denote the log-likelihood function.

In the ML problem, the goal is to find the parameters θ that maximize $p(y|\theta)$ [68]. That is, the intent is to find θ^* where

$$\theta^* = \arg\max_\theta p(y|\theta) \qquad (2.28)$$

Usually, the log-likelihood function $\log p(y|\theta)$ is used, in this maximization, instead of $p(y|\theta)$ because it is easy to analyze. Depending on the form of $p(y|\theta)$, the maximization problem may either be easy or hard. For example, if $p(y|\theta)$is simply a single Gaussian distribution where $\theta = (\mu, \sigma^2)$, then the problem is directly solved that by set the derivative of $\log p(y|\theta)$ to zero. However, it is impossible to find such analytical expressions for many

cases of the maximization. The EM algorithm is one of such elaborate techniques which used for these cases.

The basic idea behind the EM algorithm is to find θ which maximizes $L_y(\theta)$ that in the absence of x. This maximization is achieved through two steps: E-step and M-step. [68]

Let θ^k be our estimate of the parameters at the k^{th} iteration.

E-step: It is the first step in the EM algorithm. It finds the expected value of the complete-data log-likelihood $(i.e., \log p(x, y|\theta))$ with respect to the unknown data y and the current parameter estimates θ^k. That is,

$$Q(\theta|\theta^k) = E[\log p(x, y|\theta)\,|x, \theta^k], \qquad (2.29)$$

where θ^k are the current parameters estimates that we used to evaluate the expectation and θ are the new parameters that we optimize to increase Q.

The key thing to understand is that x and θ^k are constants, θ is a normal variable that we wish to adjust, and y is a random variable governed by the distribution $f(y|x, \theta^k)$. Therefore, the right side of equation 2.29 can be re-written as:

$$E[\log p(x, y|\theta)\,|x, \theta^k] = \int_{y \in Y} \log p(x, y|\theta) f(y|x, \theta^k) dy. \qquad (2.30)$$

Note that $f(y|x, \theta^k)$ is the marginal distribution of the unobserved data and it depends on both the observed data x and on the current parameters θ^k. Y is the space of values that can be taken on y. In the best of cases, this marginal distribution is a simple analytical expression of the assumed parameters θ^k and perhaps the data. In the worst of cases, this density might be very hard to obtain.

It is important to distinguish between the first and second arguments of the function Q. The first argument θ corresponds to the parameters that ultimately will be optimized in an attempt to maximize the likelihood. The second argument θ^kcorresponds to the parameters that we use to evaluate the expectation.

Due to the various formats of the expectation formula, the parameters θ may either computed as $\theta_1$or θ_2.

M-step: This step maximizes the expectation which computed in the first step. That is,

$$\theta^{k+1} = \arg\max_{\theta} Q(\theta|\theta^k), \qquad (2.31)$$

where θ^{k+1} is the value of θ which maximizes $Q(\theta|\theta^k)$.

A modified form of the M-step is to, instead of maximizing $Q(\theta|\theta^k)$, assign the biggest value (e.g., θ_1^{k+1}) of θ^{k+1}which confirms the following formula

$$Q\left(\theta_1^{k+1}|\theta^k\right) > Q\left(\theta_2^{k+1}|\theta^k\right).$$

The EM algorithm consists of choosing an initial parameter θ^0, then performing the E-step and the M-step successively until convergence. Convergence may be determined by examining when the parameters quit changing (i.e., stop when $|\theta^{k+1} - \theta^k|\epsilon$ for some ϵ and some appropriate distance measure). Therefore, the algorithm may be diagrammed starting from an initial guess of the parameter θ^0 as follows:

$$\theta^0 \xrightarrow{\text{E-step}} t^1 \xrightarrow{\text{M-step}} \theta^1 \xrightarrow{\text{E-step}} t^2 \xrightarrow{\text{M-step}} \dots$$

2.3.4 Types of HMMs

HMMs are generally divided into two types: ergodic models, left-right models.

An ergodic model has the property that every state can be reached (in a single step) from any other state. As shown in Figure 2.11(a), that every a_{ij} coefficient is positive. Hence, there is

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Concerning with the left-right model (or Bakis model), the underlying state sequence associated with this model has the property that as time increases the state index increases (or stays the same), i.e., the states proceed from left to right (see Figure 2.11(b)). The fundamental property of all left-right HMMs is that the state transition coefficients have the property

$$a_{ij} = 0, \qquad j < i. \qquad (2.30)$$

Furthermore, the initial state probabilities have the property

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \qquad (2.31)$$

Often, with left-right models, additional constraints are placed on the state transition coefficients to make sure that large changes in state indices do not occur; hence a constraint of the form

$$a_{ij} = 0, \qquad j < i + \Delta \qquad (2.32)$$

is often used. For instance, the value of Δ is 2 for the example of figure 2.11(b) such that no jumps of more than 2 states are allowed. The form of the state transition matrix for this example is

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}.$$

It should be clear that, for the last state in a left-right model, the state transition coefficients are specified as

$$a_{NN} = 1 \qquad (2.33a)$$

$$a_{Ni} = 0, \qquad i < N. \qquad (2.33b)$$

Although, HMMs are divided into ergodic and left-right models, there are many possible variations and combinations possible. For example, figure 2.11(c) shows a cross-coupled connection of two parallel left-right HMMs. This model is a left-right model (it obeys all the a_{ij} constraints); however, it can be seen that it has certain flexibility not present in a strict left-right model (i.e., one without parallel paths).

2.3.5 Continuous Observation Densities in HMMs

Notice that, all our discussion about HMMs, up to this point, has considered the case when the observations are discrete.

A discrete probability density is used is used within each state of HMMs, when the observations are characterized as discrete symbols from a finite alphabet. However, for some applications, the observations are continuous signals (or vectors). Although it is possible to quantize such continuous signals via codebooks or any other way, there might be serious degradation associated

with this quantization. Therefore, it would be advantageous, for the continuous signals, to use HMMs with probability density functions.

To ensure the pdf is reestimated in a consistent way, some restrictions have to be placed on the form of the pdf. The most general representation of the pdf, for which a reestimation procedure has been formulated [46, 47], is a finite mixture of the form

$$b_j(O) = \sum_{m=1}^{M} c_{jm}\, \Psi\left[O, \mu_{jm}, U_{jm}\right] \qquad (2.34)$$

where O is the vector being modeled; c_{jm} is the mixture coefficient for the mth mixture in state j; and Ψ is any log-concave or elliptically symmetric density (e.g., Gaussian) [46], with men vector μ_{jm} and covariance matrix U_{jm} for the mth mixture component in state j. Usually a Gaussian density is used for Ψ.

The mixture gains c_{jm} satisfy the following stochastic constraint

$$\sum_{m=1}^{M} c_{jm} = 1, \qquad 1 \le j \le N \qquad (2.35a)$$

$$c_{jm} \ge 0, \quad 1 \le j \le N, \;\; 1 \le m \le M \qquad (2.35b)$$

so that the pdf is properly modified, i.e.,

$$\int_{-\infty}^{+\infty} b_j(x)dx = 1, \qquad 1 \le j \le N \qquad (2.36)$$

This mixture form can be used to approximate, arbitrarily closely, any finite, continuous density function. Therefore, it can be applied to a wide range of problems.

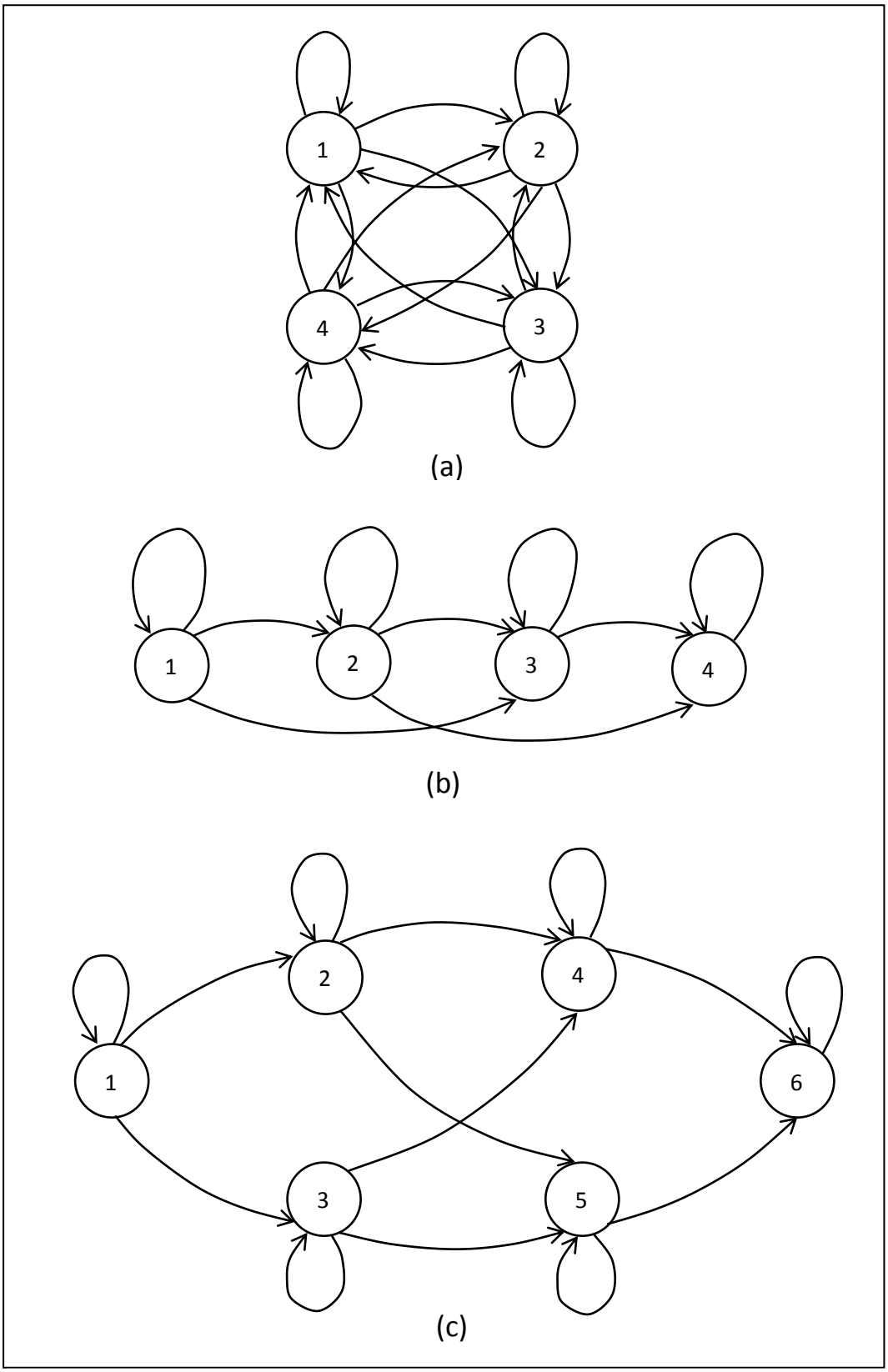

Figure 2.11: Illustration of 3 distinct types of HMMs. (a) A 4-state ergodic model. (b) A 4-state left-right model (c) A 6-state parallel path left-right model. [40]

2.4 REPRESENTATION METHODS OF ONLINE HANDWRITING DATASETS

There are two methods for the structure of online handwriting datasets: UNIPEN format, XML-based representations [54]. The UNIPEN format was the first of these methods but it has been succeed by the XML-based representations. In the next two sections (section 2.4.1 and section 2.4.2), we give a short description for these methods.

2.4.1 UNIPEN Format

UNIPEN format has been developed by I. Guyon et al. in collaboration with a work group of 14 experts [50]. It is an ASCII format that designed specifically for the data which collected with any touch sensitive, resistive, or electromagnetic device providing discretized pen trajectory information. The minimum number of signal channels in the format is two (i. e., X and Y coordinates), but more signals are allowed (e. g., pen angle or pressure information). There are provisions in the UNIPEN format for data annotation about recording conditions, writers, segmentation, data layout, data quality, labeling and recognition results.

The UNIPEN format involves a succession of instructions consisting of keywords followed by arguments, where

- The keywords are reserved words starting with a dot in the first column of a line, and
- The arguments are strings or numbers which separated by spaces, tabulation, or new-lines. The arguments relative to a given keyword start after the keyword and end with the apparition of the next keyword or the end of file.

Databases written in the UNIPEN format may optionally be organized in different files and directories, but all the data can also be concatenated into a single file.

The UNIPEN format is thought of as a sequence of pen coordinates which annotated by various information. The pen trajectory is encoded as a sequence of components *".PEN_DOWN"* and *".PEN_UP"*, containing pen coordinates (e.g., *XY* or *XYT* as declared in *.COORD*). The instruction *".DT"* permits and précising the elapsed time between two components. The content of databases is divided into one or several datasets starting with *".START_SET"*. The components within the datasets are implicitly numbered, starting from zero.

Segmentation and labeling are provided by the *".SEGMENT"* instruction. Component numbers are used by *".SEGMENT"* to delineate sentences, words, characters. A segmentation hierarchy is declared with *".HIERARCHY"*. Because components are referred by a unique combination of set name and order number in that set, it is possible to separate the *".SEGMENT"* from the data itself.

2.4.2 XML-Based Representations

The UNIPEN representation employs ASCII flat files to store handwriting data and associated annotations. This brings with it the advantages such as simplicity, ease of viewing and editing using a simple text editor, as well as the ability to define additional keywords to describe additional attributes of the data or the writers. However, UNIPEN suffers from some shortcomings which are:

- UNIPEN is unstructured. There is no way of organizing the information in semantically well-categorized classes such as dataset information,

writer definitions, label sources, or annotation hierarchy. Instead, UNIPEN provides a number of keywords that can be specified in any order.

- UNIPEN is not strict. Moreover, many relevant aspects of the data collection process and of the data itself are described in the UNIPEN COMMENT expressions. Also, keywords like SETUP often contain information about writers, recording devices, software, form layout, etc.

- UNIPEN has a scope problem. Given that the order of the entered keywords is not fixed, the scope of UNIPEN expressions is defined as follows: Any coordinate that is specified in UNIPEN, is described by the context of preceding UNIPEN tags. Any UNIPEN tags that are specified bellow other tags, are not valid for these tags.

- The focus of the original UNIPEN effort was the recognition of cursive Latin text, and the support for non-Latin scripts, and for modalities such as drawings and math is limited.

XML-based representations are successors to the UNIPEN format, such that they try to overcome the UNIPEN shortcomings [55]. XML is a natural choice for the representations because it has an advantage of being extensible with hierarchical nature. In the following, section 2.4.3 and section 2.4.4 present two examples of the XML-based representations. These examples are: hwDataset, UPX.

2.4.3 hwDataset

hwDataset was proposed as an XML representation for the annotation of handwriting data that is inspired by the UNIPEN standard. It makes use of Digital Ink Markup Language (InkML) for the representation of the digital ink being annotated [56].

Digital ink refers to a series of pen positions and optional attributes (related to time-stamp, pen pressure, pen tilt and so forth) captured from a suitable pen input device.

There are literally thousands of different digital ink aware devices (ranging from standalone digitizing tablets to PDAs, Tablet PCs and mobile phones, and proprietary devices for different vertical markets) supporting different proprietary representations of digital ink. Digital Ink Markup Language (InkML), from the World Wide Web Consortium (W3C), is an emerging standard representation of digital ink [57]. This representation is platform and device independent. InkML is designed to support the input, storage and processing of handwriting, gestures, sketches, music and other notational languages in ink-aware applications. InkML also provides a common format for the exchange of ink data between components such as handwriting and gesture recognizers, signature verifiers, and other ink-aware modules. Although InkML provides many proper features when it comes to the specification of digital ink, it lacks certain more advanced aspects necessary for annotations.

Fortunately, InkML provides means for application specific extensions. By virtue of being an XML-based language, it allows users to easily add specific information to ink files so as to suit the needs of the application at hand. In this sense, hwDataset may think of as an application-specific extension of InkML (see figure 2.12).

The hwDataset representation includes a set of XML elements for detailed annotation of handwriting. The *hwDataset* element is the root of the document which captures metadata about the dataset as part of the *datasetinfo* element, various definitions as a part of *datasetDefs*, and hierarchical

annotation of handwritten data as a part of one or more *hwData* elements (see figure 2.13).

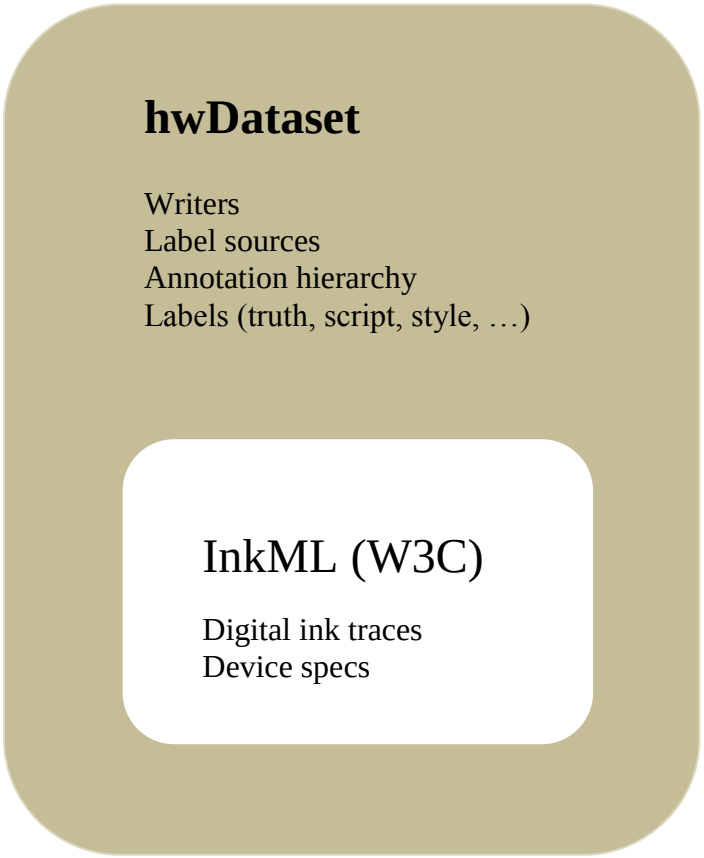

Figure 2.12: The Conceptual relationship between hwDataset and InkML. [55]

The hwDataset document may contain one or more *hwData* elements corresponding to different writing trails, or different fields of writing captured from a writer in a single trail. These instances may be distinguished using the *id* attribute. Each of the *hwData* elements follows one of the annotation schemes defined earlier. It contains one or more *H(i)* elements, where *i* refers to an appropriate level of the annotation hierarchy defined as a part of the specification of the *annotationScheme* element (see figure 2.14). Moreover, each of the *H(i)* contains one or more *label* elements which capture an

annotation information at that level. In addition, *H(i)* may contain one or more *H(i+1)* elements; also it may contain a *hwTraces* element. The *hwTraces* element is the leaf element of the hierarchy that refers to digital ink traces which represented using InkML. Furthermore, the *hwData* element may also include a *uiInfo* element that describes the writing area or field used to capture ink.

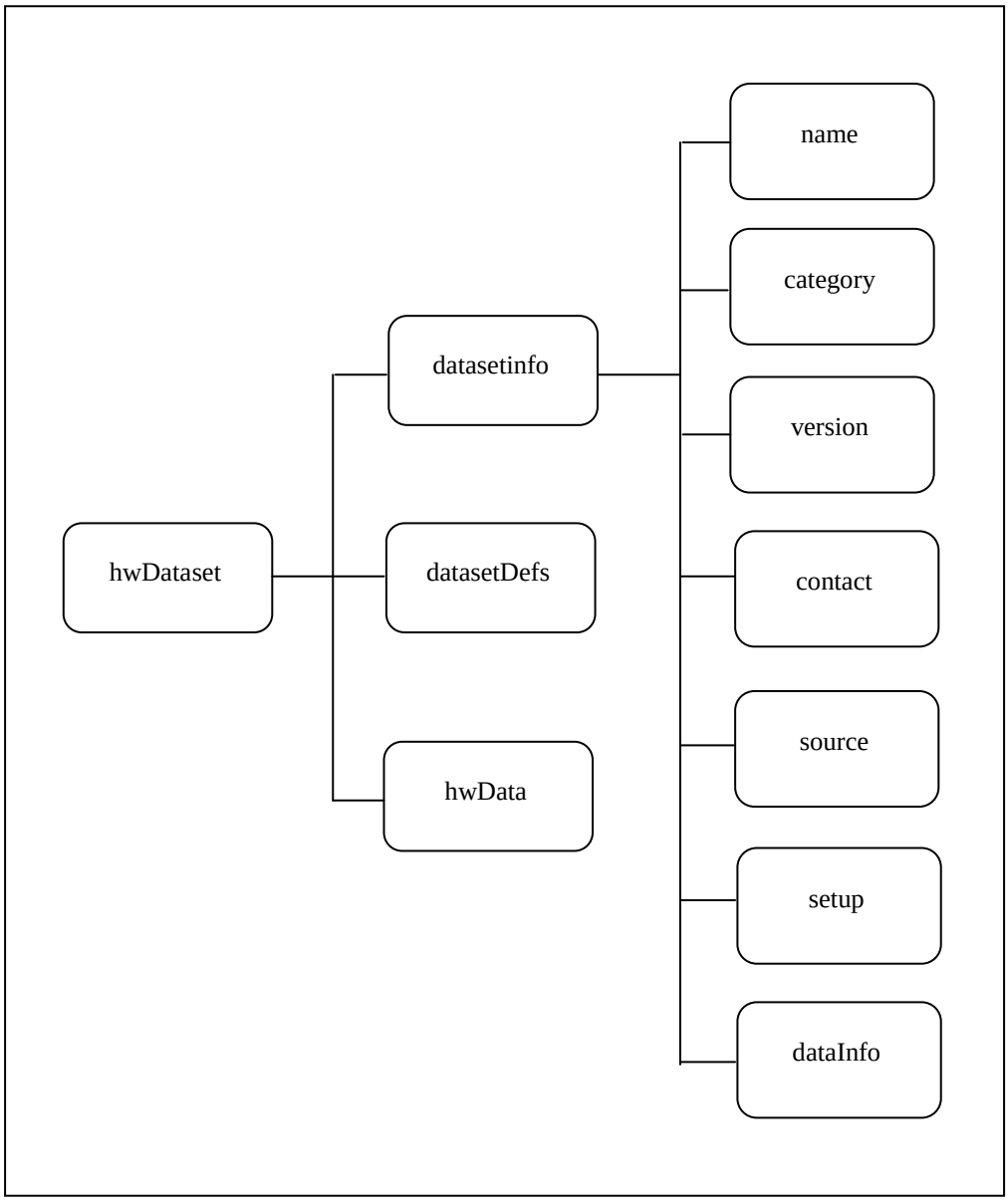

Figure 2.13: Illustration of the root element and its successors in the hwDataset representation. [55]

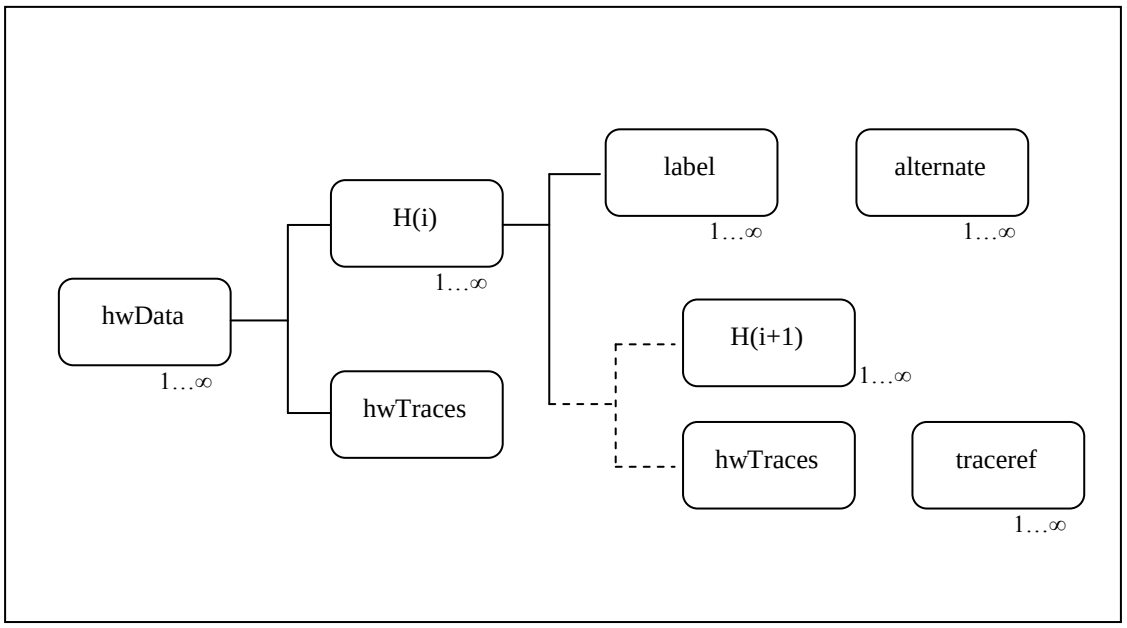

Figure 2.14: Illustration of the *hwData* element. [55]

2.4.4 UPX

The UPX definition complies with InkML. Just like hwDataset, it has been influenced by UNIPEN [55, 58]. Whereas hwDataset was created primarily to support new data collection, the starting point for UPX effort has been to assess the validity of hwDataset for storing existing handwriting data repositories.

A dataset, in UPX, comprises of multiple InkML and UPX documents which organized in a directory structure, with perhaps common information stored in a separate UPX document and referred to by other UPX documents. The *upx* element is the root element for each of the UPX documents [55]. It contains three main sub-elements for specifying annotated online ink (see figure 2.15). These elements are:

1. *datasetInfo* element specifies metadata related to the dataset as a whole. In other words it specifies a high-level characterization of the dataset (see figure 2.16).

2. *datasetDefs* element, containing information about writers, sources of annotation and annotation hierarchies referred to in the UPX document.

3. *hwData* element, contains detailed labeling of digital ink traces, organized in a customizable hierarchy. Gestures and other notations like music and math can also be accommodated in the same structure.

The *hwData* element allows hierarchical organization of digital ink data and its annotation (see figure 2.17). It typically contains the root of the annotation hierarchy, denoted by *hLevel* element, which defined for each user. At each hierarchy level, *hLevel* contains a label element that captures annotation information at that level. *hLevel* may also contain one or more nested *hLevel* elements. Furthermore, *hLevel* contains *hwTraces* element which refers to raw ink traces using InkML *traceView*.

The *hLevel* elements are meant to be used to indicate the structural makeup of handwriting as well as assign meaningful names such as PARAGRAPH and WORD using the corresponding attributes of the *hwData* element. One or more *label* elements at each level can be used to capture alternative choices. Although primarily intended to describe the truth value of a particular set of ink traces, they may also be used for describing other characteristics such as writing style, quality and script. The *alternate* element can be used to facilitate the process of manual validation by prompting options for human validation. Formally, the attributes of *label* are:

i. *id* - identification of the label

ii. *labelSrcRef*- reference to the label source defined earlier.

iii. *category* - category of the label (e.g. truth, quality, script, style, etc)

iv. *timestamp*- time for the act of annotation.

Alternatively, *hwData* can merely contain references to one or more *hLevel* elements defined in the same or other UPXdocument.

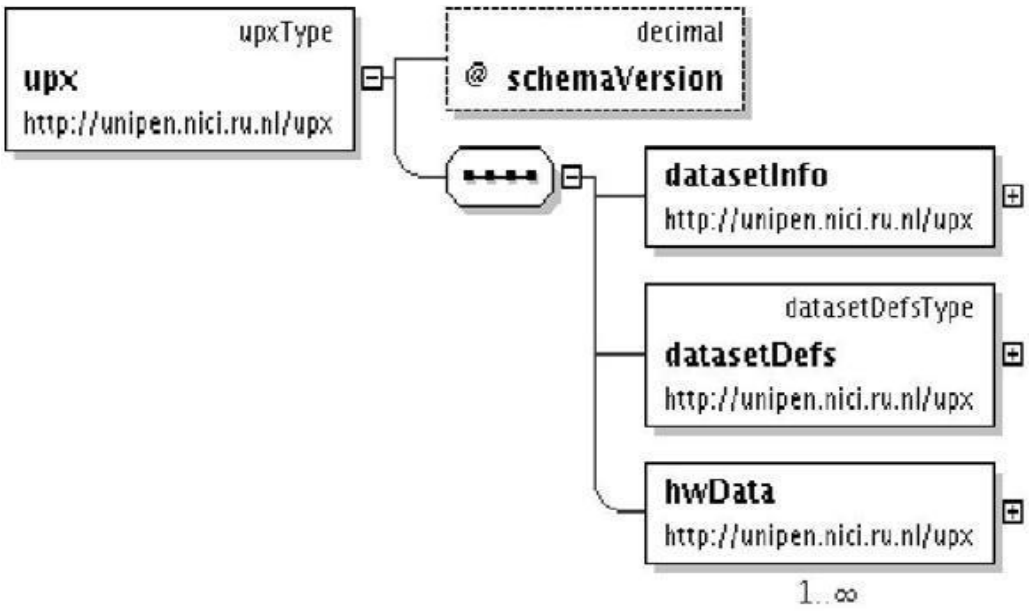

Figure 2.15: The specification of the *UPX* element. [59]

2.5 SUMMARY

This chapter has presented theoretical background for the typical system structure of online Arabic handwriting recognition, HMMs, and representation methods of online handwriting datasets. Firstly, the chapter explored the stages of the typical recognition system for online Arabic. These stages are: data acquisition, preprocessing, segmentation, feature extraction, classification, and post processing. Then the chapter has demonstrated the elements and computation problems of HMMs. Eventually, the chapter has reported the two representation methods (i. e., UNIPEN format and XML-representations) which have been used to format the datasets of online handwriting, such that UNIPEN format was the first of these methods but it has been succeed by the XML-based representations.

```
<datasetInfo id="dataSet_Firemaker-On/Off-Natural">
  <name>
    Firemaker-On/Off-Natural
  </name>
  <category>
      As part of the Wanda project, we collected various multimodal (i.e.
      online+offline) data. Subjects were asked to write their signature, to
      copy signatures from others and to write various pages of text (in
      natural or forged conditions). The text each writer had to write was the
      same for all writers. Texts were designed in cooperation with two
      forensics labs and included city names, legal amounts, numbers,
      etcetera. This Firemaker-On/Off-Natural collection contains
      handwriting acquired in natural writing conditions. Writers (mainly
      students) were rewarded with 5 euro and were asked to write the texts
      as they normally would write.
  </category>
  <version> October 2006 </version>
  <contact> Louis Vuurpijl, vuurpijl@nici.ru.nl </contact>
  <source> NICI Wanda/Firemaker Project </source>
  <setup>
      Each writer wrote on an A4 delineated paper with digital inking pen.
      The paper was optimally aligned and taped to the tablet (Wacom A4)
      to minimize translation and rotation effects. On each of the four
      corners of the paper, writers marked a cross for later calibration
      purposes. All writers managed to write all lines of text on a single A4
      paper.
  </setup>

  <dataInfo>
    <numWriters> 30 </numWriters>
    <quality> good </quality>
    <style> mixed </style>
    <groundThruth> Bob, David and sexy Xantippe.... </groundThruth>
  </dataInfo>
</datasetInfo>
```

Figure 2.16: XML code example for the *datasetInfo* element which taken from a dataset named as "Firemaker on/off collection". [60]

```
<hwdata desc = "field 1" annotationSchemeRef = "#scheme01"
            id = "FIELD01">
    <hlevel id = "ID018485" writerRef  = "WRITER01"
              level = "CHARACTER">
        <label id = "char01"  labelSrcRef  = "HPLRECOGNIZER"
                 labelType = "truth" timestamp = "100000234">
           <alternate rank = "1" score = "0.97">a</alternate>
           <alternate rank = "2" score = "0.92”>u</alternate>
        </label>
        <hwtraces>
           <traceRefhref = "sampleDoc.inkml#trace1">
           </traceRef>
        </hwtraces>
    </hlevel>
</hwdata>
```

Figure 2.17: Example of XML code for *hwdata* element. [59]

CHAPTER 3
OVERVIEW OF PRIOR WORK ON ONLINE ARABIC HANDWRITING RECOGNITION

3.1 INTRODUCTION

This chapter presents a review for influential and current research studies of online Arabic handwriting recognition.

Online handwriting recognition has been studied since 1960s [6]. Up to 1990, much research work had been done for the recognition of English characters for both isolated and cursive modes; and there had been a little research on Chinese, Indian, Korean, and some other languages; but it was very limited on Arabic characters [4].

In the late of 1990s, A. M. Alimi has reported some research efforts of online handwritten isolated Arabic characters [5]. Such efforts are:

1. Amin et al.'s IRAC I system [30]. This system was tested by 3 writers on set of 73 characters to obtain a character recognition rate of 95.45%.

2. El-Wakil and Shoukry's system [31]. This system was tested by 7 writers on sets of 60 characters to obtain a character recognition rate of 93%.

3. Alimi and Ghorbel's system [32]. This system was tested by one writer on sets of 28 characters to obtain a character recognition rate of 96%.

In addition to these efforts, Alimi reported other researches of online handwritten cursive Arabic words.These researches are:

i. Amin et al.'s IRAC II, IRAC III, and IV systems [33, 34, 35]. The IRAC II system was tested on 400 words to obtain a recognition rate of 80%.

The IRAC III system recognizes words without segmentation, and it was tested on 400 words to obtain a recognition rate of 90%. The IRAC IV has a recognition rate of 90%.

ii. S. Al-Emami and M. Usher system [4]. This system has been employed based the structural method for pattern recognition.

Two research efforts, concerning with the online Arabic handwriting recognition, had been done through the first seven years of the third millennium [7, 8]. The first of these efforts is an investigation regarding with a recognition system of online handwriting Arabic characters using Kohonen network [7]; whereas the second one is an online Arabic handwriting recognition system (AraPen) which was implemented by the use of Java for mobile devices (J2ME) [8]. Recently, some studies have shown that online Arabic handwriting recognition based on HMMs [18, 22].

Structural approach, evolutionary neuro-fuzzy, Kohonen neural network, dynamic time warping followed by simple neural, and HMMs are the classifier approaches which have been used for these research efforts. Moreover, many aspects of pattern recognition systems (such as the volume and representation of the training and testing data, the extracted features, and the recognition rates) were involved in these efforts.

In this chapter, section 3.2 reviews two of the efforts which support the structural approach, section 3.3 reviews one of the efforts which supports the evolutionary neuro-fuzzy approach, section 3.4 reviews another one of the studies that supports Kohonen neural network, section 3.5 reviews an study in these efforts that supports the approach of DTW followed by simple neural, and section 3.6 reviews a work in these efforts that supports the approach of HMMs. Eventually, the chapter is summarized in section 3.7.

3.2 STRUCTURAL APPROACH

The structural approach, also named as syntactic approach, is a pattern recognition method which relies on syntactic grammars to discriminate an object from different groups of objects based upon the morphological interrelationships (or interconnections) present in the object [36]. The following two sections (section 3.2.1 and section 3.2.2) introduce two recognition systems for online handwriting Arabic characters based on this approach.

3.2.1 S. Al-Emami and M. Usher System

In 1990, S. Al-Emami and M. Usher presented a research work for the online recognition of handwriting Arabic characters [4]. The work yields a system which employs the structural method for the recognition. This consequential system involves a preprocessing stage followed by a learning stage. Moreover, the system involves a segmentation stage which employs Balid and Halton segmentation approach. A detailed demonstration for this approach is given in section 2.2.3.

The handwritten words, those introduced to the system, are entered by means of a pen and tablet. The data obtained by the tablet are string of X and Y coordinates for the pen positions. The dots of the words create a difficulty in the segmentation process because the length of the dots is less than the threshold which specified for segment lengths, moreover some of these dots may be very close. If a segmentation length is found and less than a specified limit (40 pixels), a special flag is used to signify the presence of dots.

Some features are produced at the end of the system preprocessing stage. These features are:

i. The code for the word (groups of code directions separated by 0s).

ii. The length of each segment.

iii. The X and Y coordinate of each segment point.

iv. Flags referring to dots.

v. A tangent value representing the slope of each segment.

In the learning stage, the features which generated by the preprocessing stage are entered into a decision tree. The same preprocessing and learning stages are also applied so as to achieve the recognition task for unknown characters. An experiment for the system training and testing was performed by limited words and characters.

This research work was the first step towards the development of a comprehensive system which could be used for:

1. Typewritten characters. A word skeleton should be obtained, and its points should be coded its points to give a similar string similar to that created by the tablet. The recognition should be 100% for any font and size.

2. Predirected write. The recognition in this case was 100%.

3. One writer. The parameters were adjusted according to the writing of one person. There is a very high percentage of recognition for that person, and a lower one for others.

3.2.2 T. S. El-Sheikh and S. G. El-Taweel System

In 1990, T. S. El-Sheikh and S. G. El-Taweel demonstrated a recognition system of real-time handwritten Arabic characters [6]. The structural approach is used in the system to recognize these characters. The

system doesn't involve segmentation; it assumes that the characters result from an outer reliable segmentation stage.

The system uses a set of different features in the recognition process. The features are elementary shapes which constitute the characters (e. g., straight lines, cusps, and etc.)

Depending on the character positions within words, the Arabic characters are divided into four subsets: Isolated characters, end characters, beginning characters, and middle characters. The first subset includes those characters which appear in an isolated form, wherever their positions are in different words. The second subset includes characters which encountered at the tail of words or sub words, while the third subset includes characters which found at the head of words or sub words. The last subset includes those characters which appeared within words or sub words. Each of these subsets is further divided into other subsets according to the number of strokes which constitute the characters. For each of these subsets, the system dedicates a classification tree. Thus, the system involves various classification trees. When the system is used to recognize a character, its recognizer takes the features of the character and examines them through the classification trees. The system includes 1200 samples of more than thirty different characters in the first of the four subsets, it also includes 1000 samples in the second, and it includes of approximately eight hundred and nine hundred for the third and last subsets. These enclosed samples achieve 100% classification rate for the first subset, 99.9% for the second, 98% for the third, and 100% for the last subset.

3.3 EVOLITIONARY NEURO-FUZZY APPROACH

Seventeen years later, A. M. Alimi (1997) presented a recognition system for on-line Arabic handwriting [5]. The system involves a fuzzy neural network which used to recognize probable characters for online Arabic

handwriting. It also has a genetic algorithm which employed to select the best combinations of the recognized characters.

The features, within this system, were represented by the curvilinear velocity of the handwriting. The definition of the curvilinear velocity is given in section 2.2.4. The produced curvilinear velocity is described by a nonlinear function of time and many parameters $V(t) = f(P_1,\ P_2, \cdots,\ P_n)$, where P_1, $P_2, \cdots,\ P_n$ are neuro-physiological and biomechanical parameters. These parameters are used as features in the recognition phase.

The system was developed base on a set of 56 different shapes of Arabic characters (see figure 3.1). The set comprises different shapes for each one of the Arabic characters; such that the different shapes for the character come from its positions within words (beginning, middle, end, isolated). All the diacritical marks as well as the points under and over the characters and the hamza symbol (ء) were isolated in the preprocessing phase of the system. Therefore, the set takes only the principal component of the characters. Each of these characters can be generated at most by 6 basic strokes which in turn were represented by 6 feature vectors. Each of these vectors was further represented by n elements (the n parameters $P_1,\ P_2, \cdots,\ P_n$ of the velocity profile of each stroke). Some of the 6 vectors may be zeros if their corresponding character contains less than 6 basic strokes.

The neural network which involved in the system has a hierarchical structure. It contains six stages, where each of these stages involves a three layers network that named as Beta fuzzy neural network (see figure 3.2). The Beta fuzzy neural network was a new architecture which created by the system developer. It was more flexible and universal than its contemporary architectures. Briefly, it can be seen as a radial basis function neural network in which the Gaussian kernels are replaced by beta functions.

Figure 3.1: The different shapes of Arabic letters. [5]

In each one of the six stages, the input layer has Ni = n neurons corresponding to the n parameters ($P_1, P_2, \ldots, P_n$) of the velocity profile of each stroke, the hidden layer has Nh = 100 neurons (Nh was determined by experimental guessing), and the output layer has No = 56 neurons corresponding to the different Arabic character shapes. Therefore, the entire network has (6×Ni) + (6×Nh) + (6×No) = 6n+936 neurons.

The recognition of characters for a given handwritten word is performed by the neural network. Prior to the recognition, the velocity of the handwritten word is reconstructed from the superposition of the basic 6 strokes. The strokes extracted from this modeling phase are used as locations for segmentation; furthermore, they introduced to the network so as to achieve the recognition of the word characters. Each character of the real handwritten word may correspond to a consecutive group of the extracted strokes (it can be a character of one, two, three, four, five, or six strokes). The neural network assigns many fuzzy values for each group of the extracted strokes, where each fuzzy value represents a degree of match between the group and a character specified in the neural network output.

The extracted strokes for any given handwritten word (strokes which produced as a result of the reconstruction for the velocity of the handwritten word) may refer to many combinations of characters. The genetic algorithm

works to find the best combination of characters which figure the handwritten word.

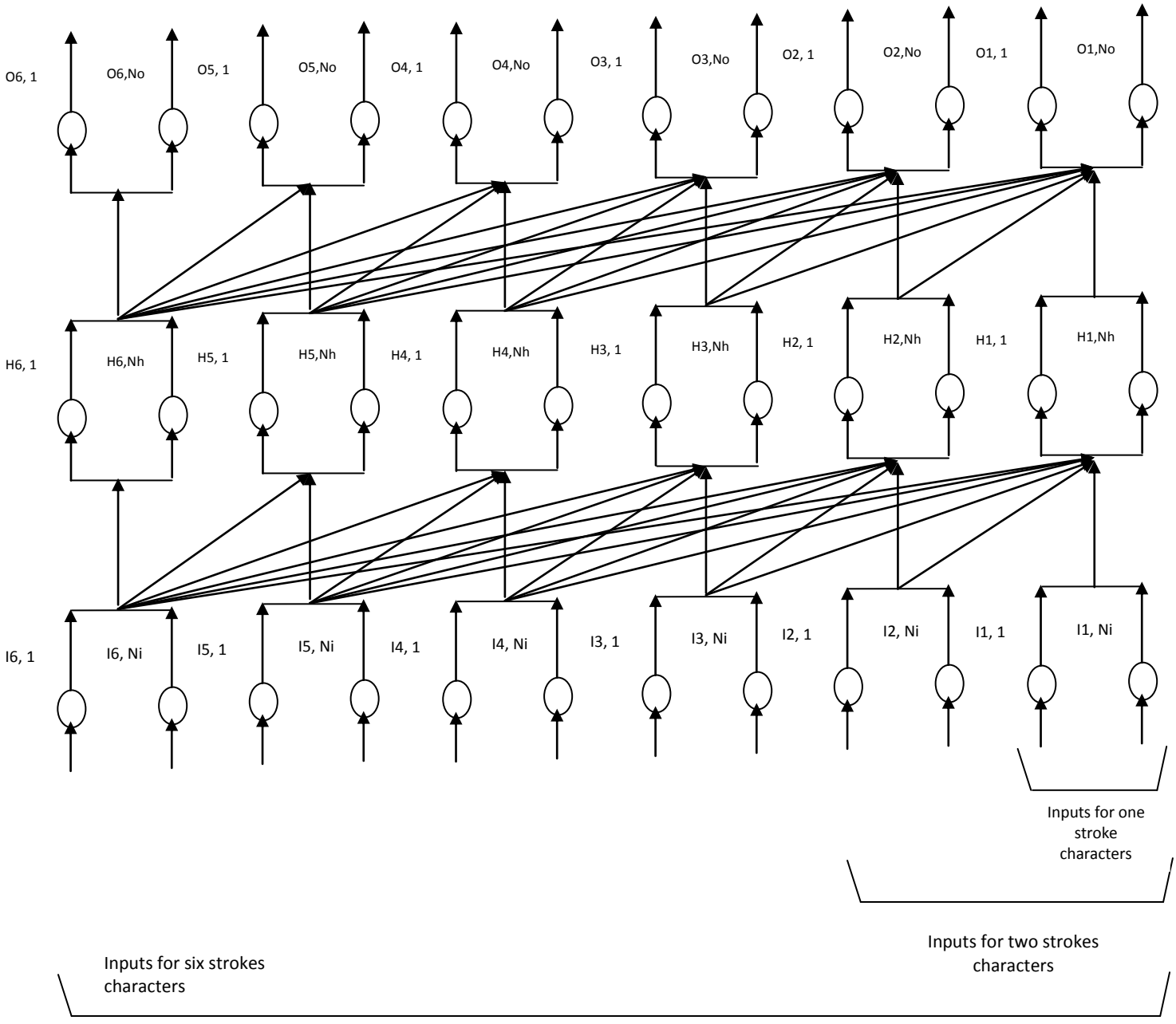

Figure 3.2: Architecture of the beta fuzzy neural network. [5]

The system is writer dependent. The beta fuzzy neural network was trained with 2000 characters written by one writer; and then it was tested on 100 replications of the word (عزّالدّين) written by the same writer. For each

word, the best solution is taken after ten executions (one should remember that genetic algorithms are probabilistic methods so one can obtain different solutions on each run). The system achieves a recognition rate of 89% without the integration of the diacritical marks information as well as positions and number of dots.

3.4 KOHONEN NEURAL NETWORK

An on-line recognition system of handwriting Arabic characters in which Kohonen neural network is used was studied by N. Mezghani and colleagues (N. Mezghani et al 2005) [7]. The system was investigated through a run of three experiments. The results of these experiments show that the network is successfully recognize both clearly and roughly written characters with a good performance.

The neural network involved in the system was the Kohonen memory (also called Kohonen self-organizing map) which used to represent points in a source into a lesser number of points in a target space, such that the neighboring memory points have neighboring values to preserve a topological ordering. The memory was organized as a two-dimensional array as shown in figure3.3. $X = (x_1, x_2, \dots, x_I)$ was the input to the memory. $W = (w_{1j}, w_{2j}, \dots, w_{Ij})$ was the weight for the memory node j.

The input of the network is a vector of features which extracted from the dynamic representation of a handwritten character. These features are Fourier coefficients for the X(t) and Y(t) components of the pen positions that constitute the handwritten. More discussion for the dynamic representation is given in section 2.2.4.

A database containing about 7400 samples of Arabic characters is used for the training and testing of the network, such that the training set contained

5000 samples and the testing set about 2400 samples. The database is prepared by the use of 18 shapes for Arabic isolated characters which written 24 times by 17 writers (see figure 3.4). The set of Arabic alphabet consists of 29 isolated characters. The 18 shapes have been got by the remove of all diacriticals marks, which do not carry important information about the character shapes, from the set.

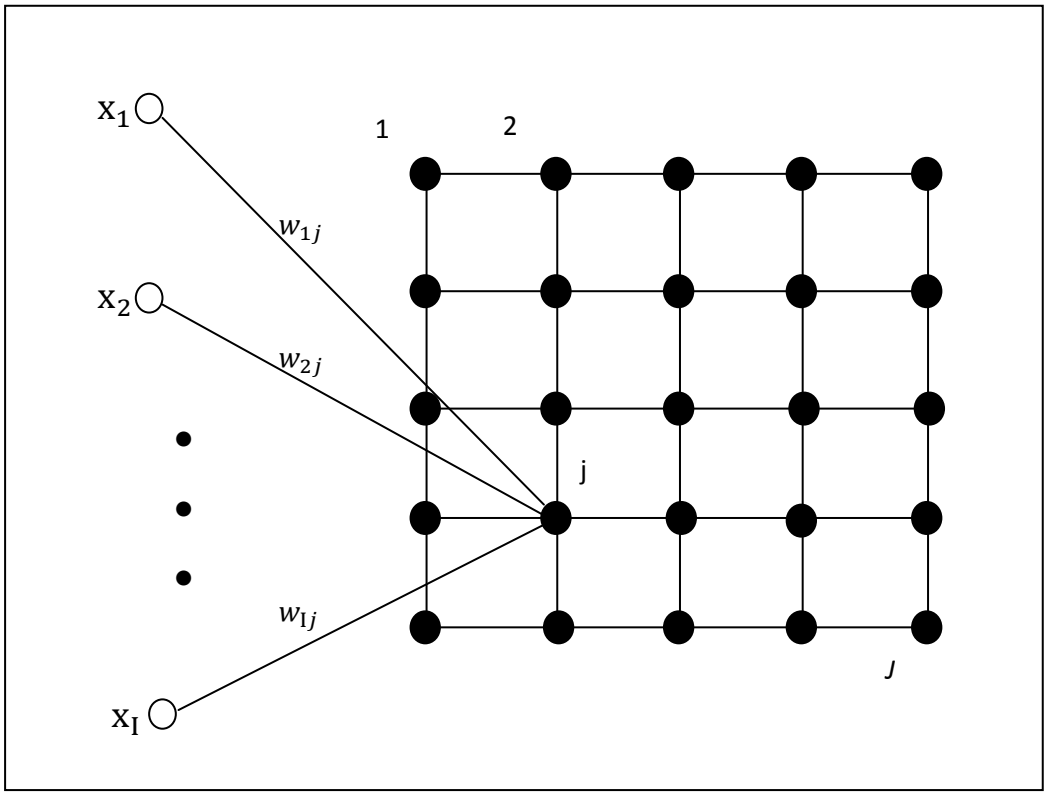

Figure 3.3: Kohonen memory of *J* nodes. [7]

To investigate the effect of various parameter settings on the network recognition rate, three experiments have been done. These experiments are:

i. Experiment 1: investigates the effect of the dimension of the feature vector on the recognition rate.
ii. Experiment 2: investigates the effect of the iteration number of the training algorithm on the recognition rate.

iii. Experiment 3: investigates the effect of the number of nodes in Kohonen memory which occurs on the recognition rate.

Experiment 1 shows that the best recognition rate is obtained with a dimension of the feature vector equal to 17, which corresponds in Fourier descriptors to n = 5 with omitting the features $a_1^{**}, b_1^{**}, c_1^{**}$(see figure 3.5). The results of experiment 2 show that the recognition rate increases rapidly and reaches a maximum to remain approximately constant, where the dimension of feature vector is fixed to 17, the number of nodes is set to 1600, and 100 iterations are retained for the training algorithm (see figure 3.6). Experiment 3 shows that the recognition rate reaches its maximization at 1600 nodes, where the dimension of the feature vector is set to 17 and the iteration number to 100 (seefigure 3.7).

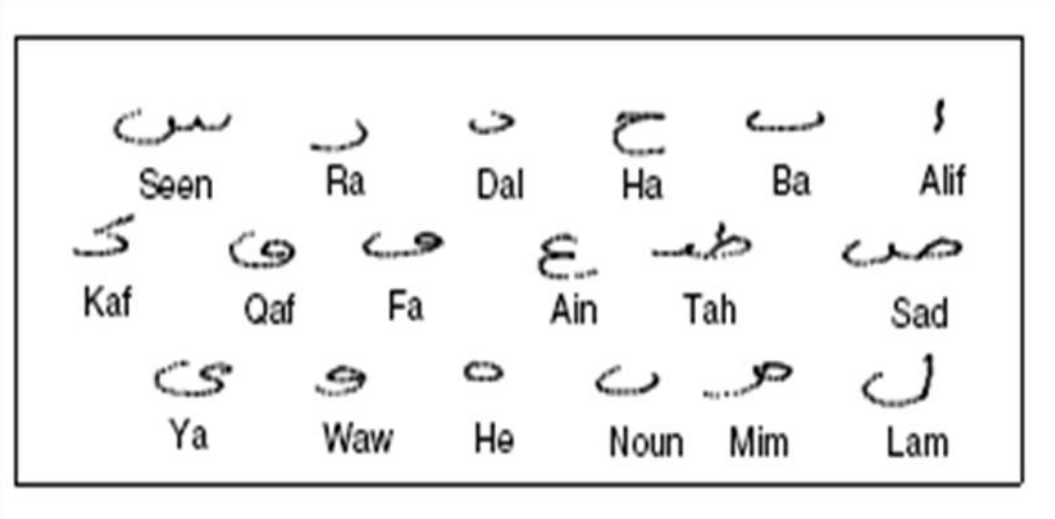

Figure 3.4: The 18 shapes of Arabic isolated characters. [7]

The dimension of the feature vector 17, the number of iterations 100, and the number of nodes 1600 are the best parameters. The training of the system is repeated using these parameters and all database entries (training and testing set). After that the system is tested with 10 writers. The recognition rate is about 88.38%.

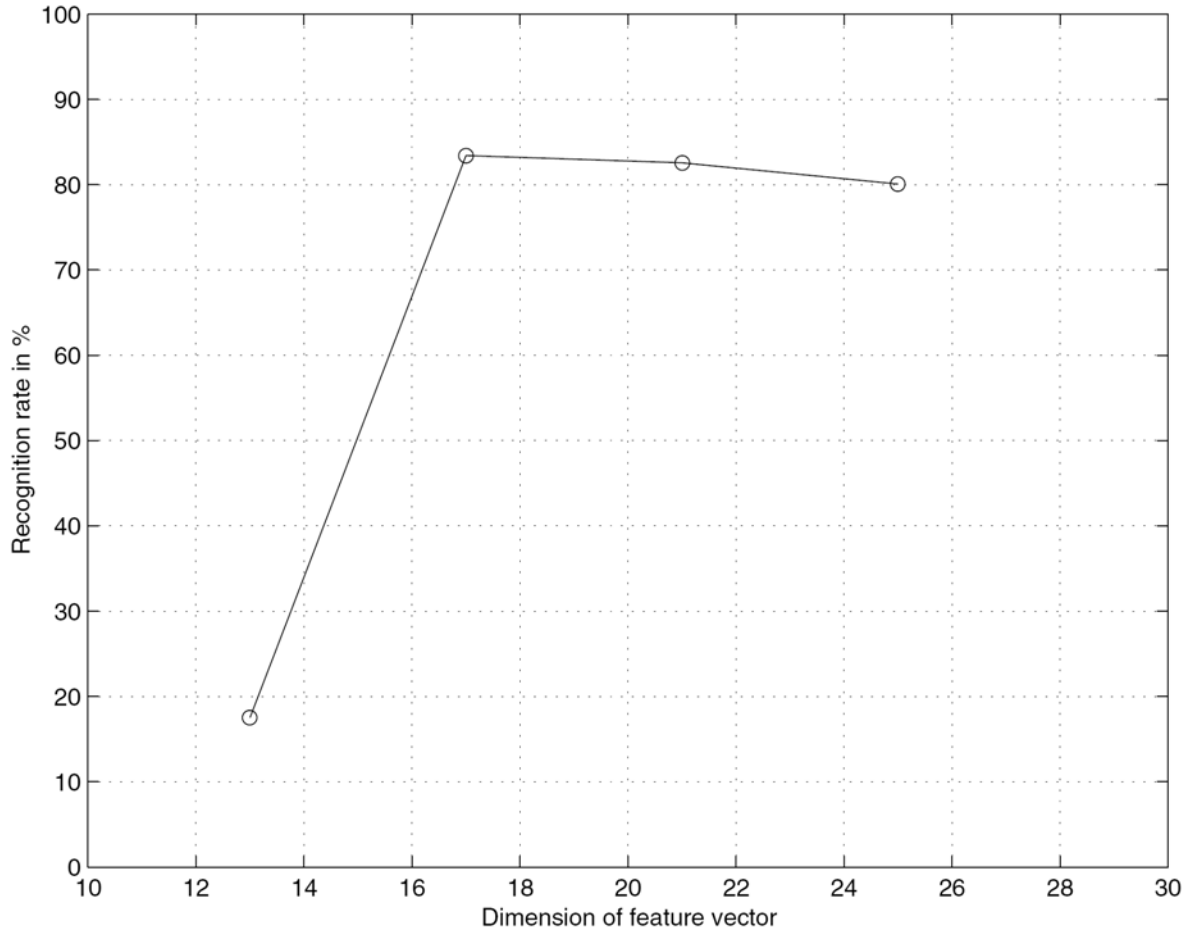

Figure 3.5: Recognition rate versus dimension of features vectors. [7]

3.5 DTW FOLLOWEDBY SIMPLENEURAL APPROACH

A trainable system of online Arabic handwriting recognition (called AraPen) was developed by B. Alsallakh and H. Safadi in 2006 [8]. The system has been constructed based on DTW which is a mathematical matching algorithm. The system structure consists of the five stages of recognition systems: data acquisition, preprocessing, feature extraction, classification, and post-processing stage (see figure 3.8). The local point features which described in section 2.2.4 are used to represent the input of AraPen. The testing results of the system have shown high recognition rate for non-cursive character recognition.

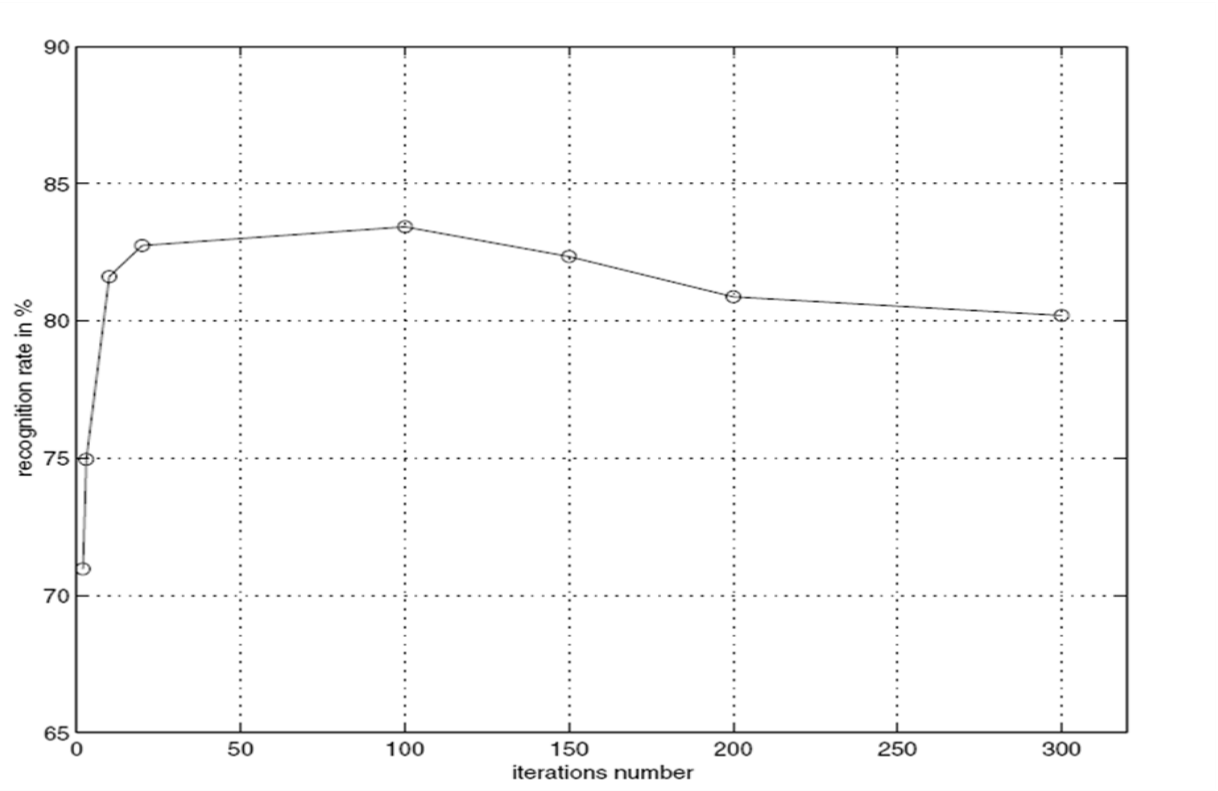

Figure 3.6: Recognition rate versus number of iterations. [7]

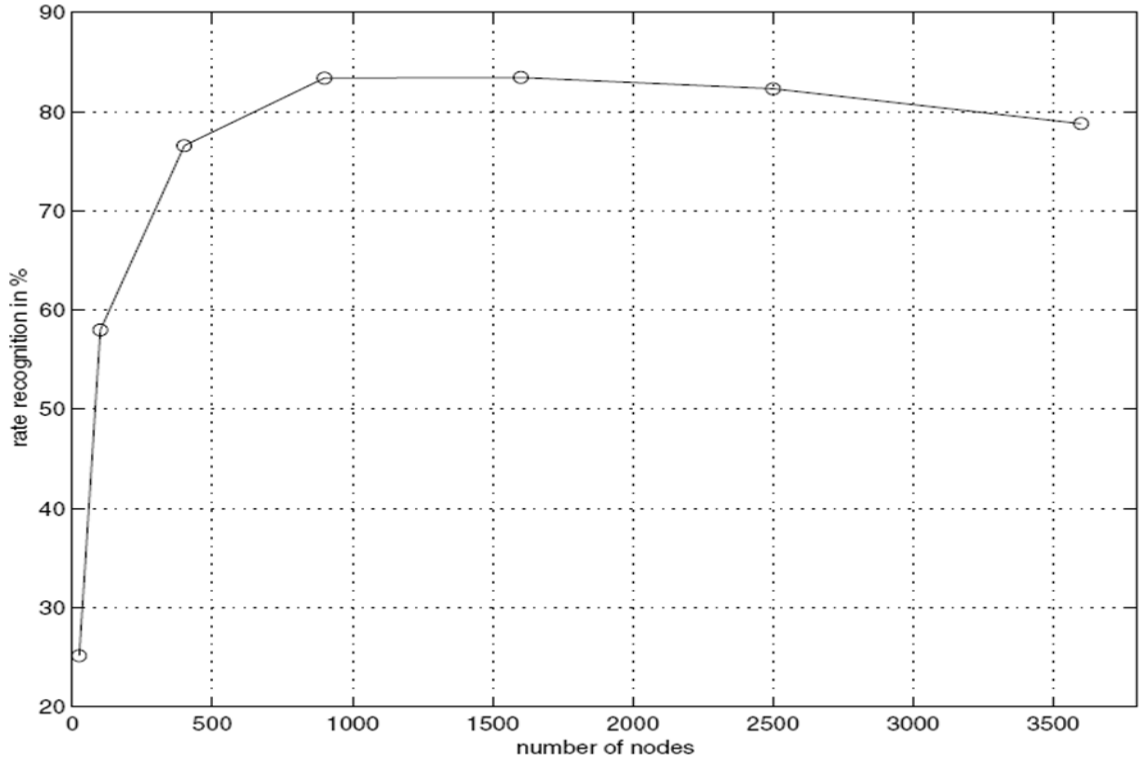

Figure 3.7: Recognition rate versus number of nodes. [7]

The system is deployed with a default pattern set, containing 21 patterns, which represent the set of isolated Arabic letters. Each of these patterns stores the features extracted from a "standard" input. The system user can alter the way a letter or digit is drawn and the features of the patterns that represent that character are updated accordingly. Moreover, the user can add new features to the pattern set and associate certain output to it. This situation enacts the training of the system. One example for each letter is sufficient to train the system and the training time is negligible. AraPen is trainable and alphabet extensible but it is not writer independent.

To involve the segmentation in AraPen, the developer of has implemented two segmentation methods: explicit segmentation, implicit segmentation.

- **Explicit Segmentation:** Usually, the connection between two Arabic letters is horizontal. This means that the input which forms cursive handwriting is portioned into horizontal segments. This segmentation method dedicates a length threshold for the segments, so a segmentation point is placed in the end of horizontal segment if it's long enough. Since the writing is online, the system can inform the user immediately when a segmentation point is placed. This feedback helps the user in controlling the length of the horizontal segments and hence achieving better recognition rate. Color information is used to convey the feedback.

- **Implicit Segmentation:** Tappert generalized in [38] the DTW approach to cursive handwriting recognition that by finding the best pattern chain which match the input. This method implicitly places the segmentation points in the input since it finds the part that matches each letter in the best chain. The number of possible chains is very prohibitive; so once again,

Dynamic Programming is applied to find the best chain in reasonable amount time. The performance of this approach depends on the distance function used. The recognition rate generated by the approach was far from that of the non-recursive case; therefore, the developers promises to investigate some techniques which can achieve satisfactory recognition rate by incorporating other features in the distance function and by combining the results from other classifiers.

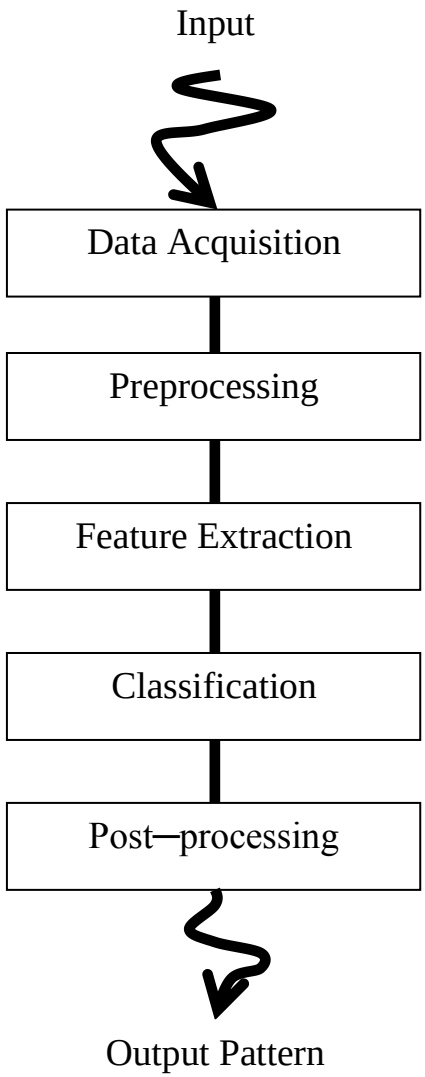

Figure 3.8: System stages of AraPen. [8]

The system was tested using a small corpus of data collected by the developers. The recognition rate in the non-cursive mode was about 91% with

the default pattern set, and 98% after training the system with the user's handwriting. In the cursive mode, using simple distance function the recognition was less than 50%.

3.6 APPROACH OF HIDDEM MARKOV MODELS

Although the approach of HMMs was proposed in the late1960's as a technique for speech recognition, it is recently used for the problem of online Arabic handwriting recognition [18, 22, 39, 40].

In 2011, H. Ahmed and S. A. Azeem introduced an On-line Arabic Handwriting Recognition System based on HMM [22]. The training and testing of the system were achieved by the use of the ADAB-database [61]. The system was implemented by the HMM Toolkit (HTK Engine) to generate 64 Left-to-right HMMs for Arabic letters. Most Arabic letters have four different shapes, depending on their position within a word which results in more than 100 HMM models. After removing the delayed strokes, the number of the models decreased to 64 models where the similar letters were grouped together in one class; for example the letters "ب", "ن", "ت", and "ث" were reduced to one class (see figure 3.9).

The main feature of this system is the removal of the delayed strokes in the training and test phases in order to avoid the confusion caused by the vast differences in the order of the writing of the delayed strokes by different writers. As shown in Table 3.1 and 3.2, the results obtained by the system have clearly outperformed the results of REGIM-HTK and REGIM-CV-HTK. REGIM-HTK and REGIM-CV-HTK are both HTK-based recognition systems which were employed in the ICDAR 2009 Online Arabic Handwriting Recognition Competition [41].

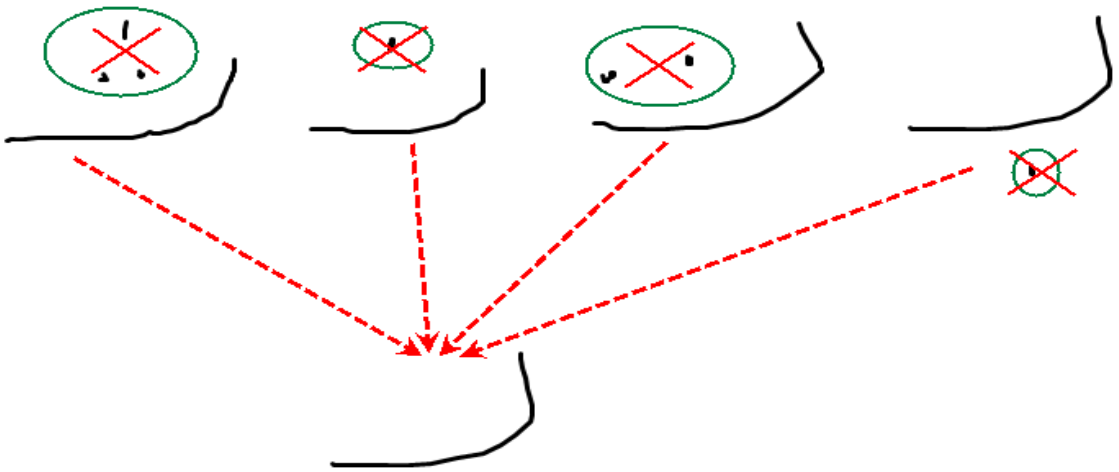

Figure 3.9: Grouping the Arabic letters in classes after the remove of the delayed strokes. [22]

The issue of boundaries detection for the letters of handwritten Arabic words has been a challenging problem in the field of online Arabic handwriting recognition. Researchers, especially those who employ HMMs, are interested in resolving this problem. To mitigate the problem, G. Al-Habian and K. Assaleh use a sliding window of length of 10 samples that goes over the feature vectors of the words [18]. Moreover, Biadsy F. et al ask the persons, who enter the training data, to manually specify demarcation points that separate letter shapes such that all delayed strokes of a letter shape are horizontally between the letter shape's demarcation points [21].

Table 3.1: Recognition rates for the system introduced by H. Ahmed and S. A. Azeem. [22]

Training Data	**Testing Data**	**Recognition Rate**
Set 1 and 2 of the ADAB-database	Set 3 of the ADAB-database	95.27%
Set 1 and 3 of the ADAB-database	Set 2 of the ADAB-database	89.72%
Set 2 and 3 of the ADAB-database	Set 1 of the ADAB-database	92.71%

Table 3.2: Recognition rates of REGIM-HTK and REGIM-CV-HTK. [22]

System	Set 1 of the ADAB-database	Set 2 of the ADAB-database	Set 2 of the ADAB-database
REGIM-HTK	57.87%	54.26%	53.75%
REGIM-CV-HTK	28.85%	35.75%	30.6%

3.7 SUMMARY

This chapter presents a review for most priori studies of online Arabic handwriting recognition. The review has been presented in a categorization context based on the involved classifier approaches of the studies. These approaches are: structural method, evolutionary neuro-fuzzy, Kohonent neural network, DTW followed by simple neural, and HMMs approach. Moreover, the review explores many aspects of the studies such as the volume of data used in the training and testing experiments, the extracted features, and the acquired results.

CHAPTER 4
ONLINE ARABIC HANDWRITING DATASETS (SUSTOLAH)

4.1 INTRODUCTION

This chapter describes the design of two datasets, named as SUSTOLAH, for online Arabic handwriting; presents two software tools which developed to construct the datasets; and shows several key conclusions of these datasets.

The first four letters of the term SUSTOLAH stands for Sudan University of Science and Technology and the remaining letters of the term stands for Online Arabic Handwriting. SUSTOLAH is a combination of two datasets which are: a) online handwriting dataset of isolated Arabic letters; b) online Arabic handwriting dataset of person's names. A software tool, named as collection tool, has been developed to gather the data of SUSTOLAH datasets. In addition, another software tool, called verification tool, has been developed to investigate and check the content of the datasets.

The creation of SUSTOLAH and the two software tools was motivated by the limitation and lack concerning with data of online Arabic handwriting [4, 5, 6, 7, 8]. SUSTOLAH has been created to be used in this thesis, as well as in any development effort regarding with online Arabic handwriting recognition.

Publications on pattern databases have been increasing and attracting research works [11, 49, 50, 51, 52]. These datasets are as important as recognition methods for the research works of pattern recognition field [48].

The datasets enable the field researchers to train and test their derivative recognition systems and techniques.

Datasets of online handwriting are the resources which involve huge data of online handwriting. On-line handwriting refers to the handwriting which formed by a pen movement on the screens of Tablets, Tablet PCs, or PDAs screens; and the handwriting can be handled by computing devices at the same time of its formalization [1].

Section 4.2 of this chapter presents the design for SUSTOLAH datasets. Section 4.3 gives a description for the collection tool. The verification tool is explored in Section 4.4. In section 4.5 we present the results and discussion of this chapter. Section 4.6 summarizes the chapter.

4.2 DESIGN OF SUSTOLAH DATASETS

After their effort in introducing the UPX, M. Agrawal et al have invited the research community of the handwriting to make UPX a reality [55]. According to this invitation, we set UPX to be the format for the datasets of SUSTOLAH.

SUSTOLAH datasets involve two classes of data which are: ink data and annotations. The ink data embodies online handwritten objects which stored in the datasets. The annotations provide descriptions of the datasets. Moreover, they describe names, occupations, and addresses of the writers who induct the handwritten objects. Furthermore, the annotations carry the corresponding text of the handwritten objects.

The ink data is extracted from the pen tips which form the handwriting. These extracted data include:

- Coordinates of the pen tips.

- The time of the pen tips.
- The information of the pen tip status.

We use the same design for the two datasets of SUSTOLAH. Each of the datasets consists of annotated online Arabic handwritten objects. The design sets the datasets in form of XML documents which employ UPX and most aspects of InkML schemas [55, 58]. These documents combine the ink data and annotations of the datasets. The design consists of two types of documents: main document and ink documents. Section 4.2.1 describes the main document and section 4.2.2 describes the ink documents.

4.2.1 Main Document

The main document is an XML document that conforms to the UPX schema. It includes information which provides a detailed description to the datasets. Moreover, it involves personal data of writers who providing handwritten objects to SUSTOLAH datasets. For each of the writers, the main document involves personal data as well as references to the documents which contain the writer handwritten objects.

The *upx* element is the root element of the main document. It involves two elements: *datasetInfo* element and *hwdata* element (see figure 4.1). The *datasetInfo* element includes a *name* element to capture a dataset name, *category* element to describe the dataset, *version* element to refer to the dataset creation date, *contact* element to enclose the dataset designer name, *source* element to refer to the institution which sponsor the dataset, and *setup* element to describe for the device specification which used for the collection of the dataset content.

The *hwdata* element involves many *hLevel* elements, such that each of these elements is dedicated to a specific writer. Each of the *hLevel* elements

includes a *writerRef* attribute to hold a writer name; moreover it includes two *label* elements as well as *hwtraces* element (see figure 4.2). In turn, each of these two *label* elements has two attributes: *id* attribute and *labelSrcRef* attribute. These attributes assign a writer address and occupation to the two *label* elements. The *hwtraces* element encompasses one or more *raceRef* elements, where each of these *traceRef* elements involves *href* attribute that refers to an ink document.

4.2.2 Ink documents

Each of these ink documents embodies a handwritten object (i.e., handwritten name or letter). The ink documents are an XML documents that conform to the InkML schema. In addition to a handwritten object, each of the ink documents includes information that describe the document, information to describe the format of the handwritten object, and personal data for the writer who disserted the handwritten object.

The *ink* element is the root element of the ink documents. This root element includes *annotation* element that describes its ink document, other three *annotation* elements which captures a writer personal data, and a *traceFormat* element which introduces the format of a handwritten object (see figure 4.3). Moreover, the root element contains one or more *trace* elements for the handwritten object, such that each of these elements is dedicated for one stroke of the object.

Each of the three *annotation* elements, which capture the personal data of the writer, includes *type* attribute. The value of this attribute indicates whether the element involves the name, the occupation, or the address of the writer. The *traceFormat* element includes four *channel* elements which in turn support four pen tip properties for the handwritten object. Each of these *channel* elements involves *name* attribute that indicates a property name, *type*

attribute which implies the value type of the property, and *units* attribute that indicates the measurement units of the property value.

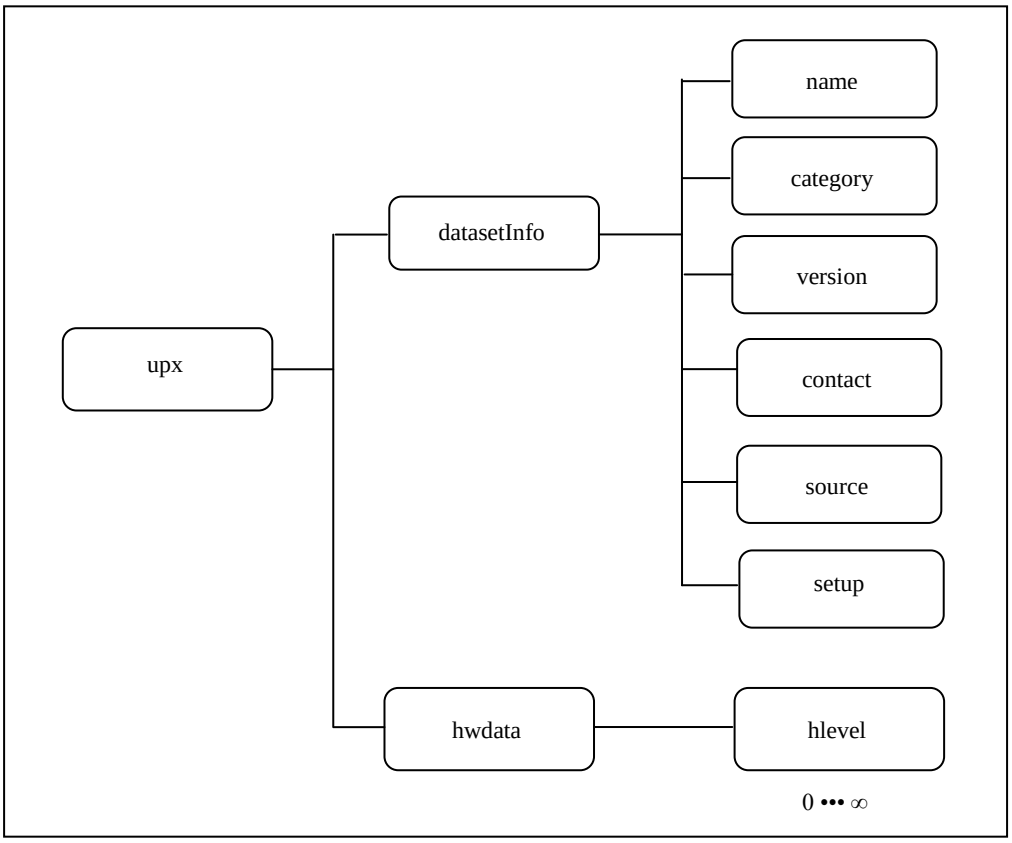

Figure 4.1: Illustration of the *root* element for the Main document of SUSTOLAH.

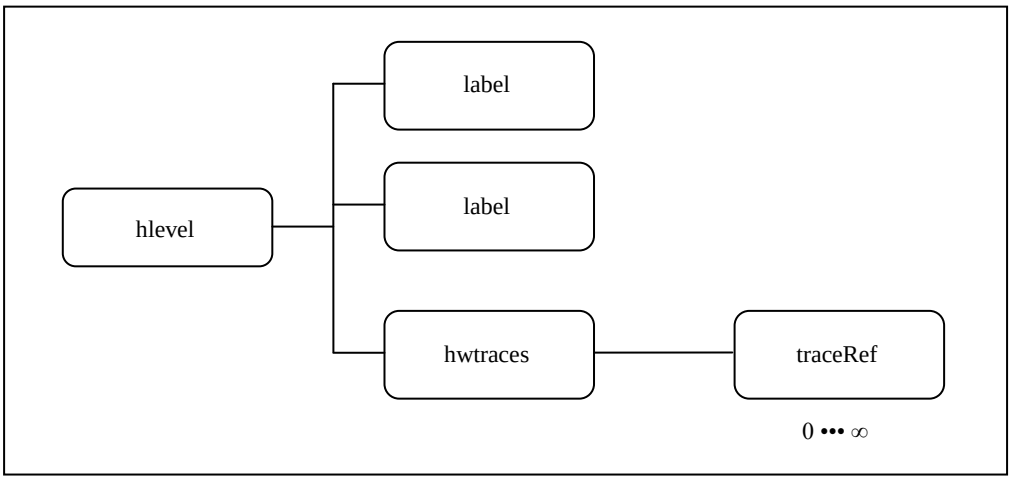

Figure 4.2: Illustration of the *hlevel* element for the Main document of SUSTOLAH.

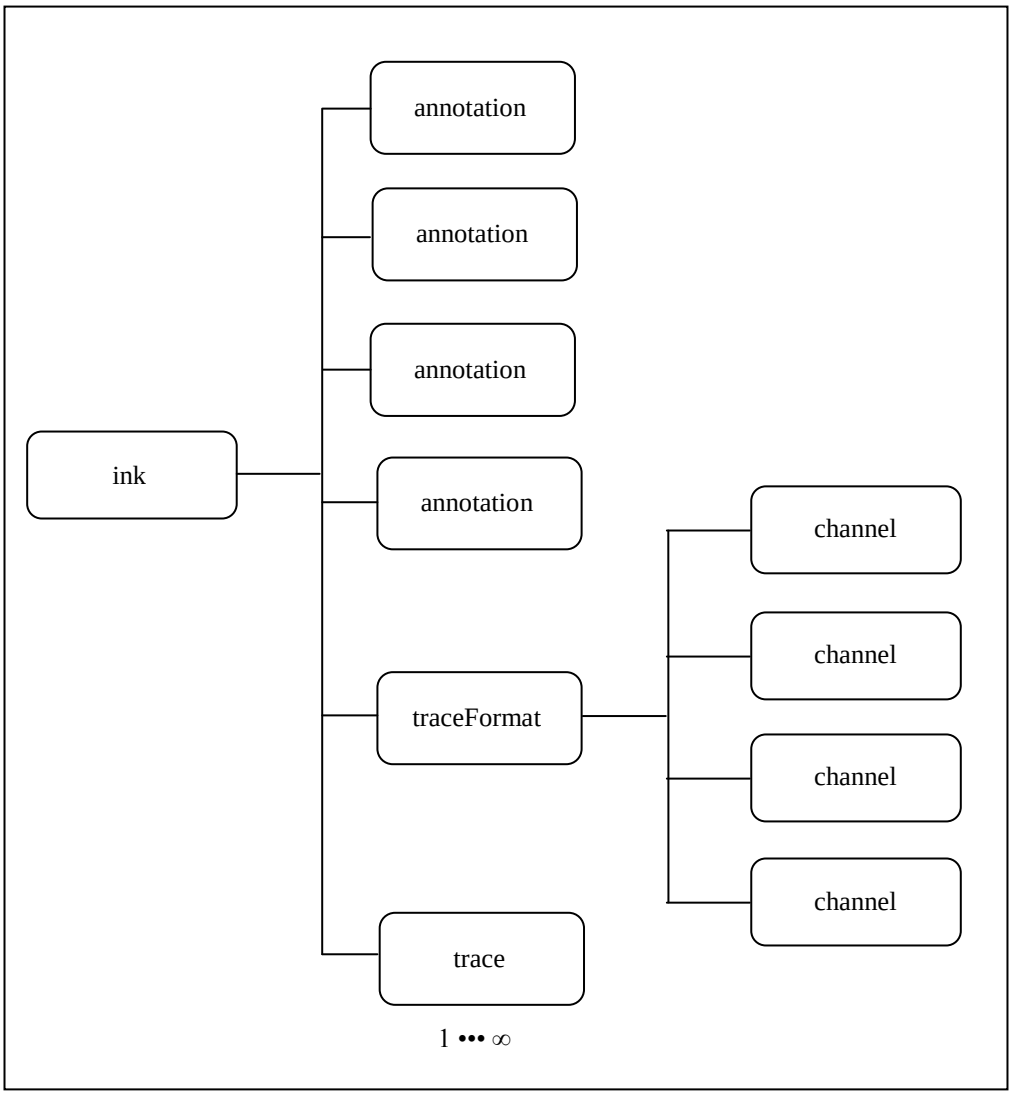

Figure 4.3: Illustration of the included elements for the ink documents in of SUSTOLAH.

4.3 COLLECTION TOOL

A software tool (called collection tool) has been developed to collect online Arabic handwriting which build the SUSTOLAH datasets. It is a visual .net based application that can be executed on any Tablet PC. A tablet PC is a convertible notebook that combines digital ink technologies with the basic aspects of PCs. In turn, the digital ink technologies allow movements of digital pens to be captured as digital ink.

This tool provides two forms to capture online Arabic handwritten objects (i.e., handwritten words and letters) and some of writers' personal data for the datasets. These forms are:1) form for capturing handwritten letters; 2) form for capturing handwritten person names (see figure 4.4 and figure 4.5). Each of these forms provides entries to capture the writers' personal data as well as to select texts to be written by the writers. Moreover, the form of handwritten letters provides a pane of size 473X358 pixels to write the selected letters, whereas the form of handwritten words provides a pane of size 1121X419 pixels to write the selected words.

Figure 4.4: The form for capturing the handwritten letters.

In fact, the tool accepts each of the handwritten objects as a trajectory of points which regulated in one or many strokes. Then it generates an ink document for this object. In this document, the tool states a *trace* element for

each of the strokes, *traceFormat* element for the format of the handwritten object, *annotation* element for the dataset description, and three other *annotation* elements for the writer personal data (see figure 4.6). The *trace* element includes lines of integers, where each of these lines corresponds to specific point in the trajectory and involves four integers which respectively figure the X coordinate, Y coordinate, time, and status of the point.

For each write, the software tool inserts a *hlevel* element into the main document. This element captures the personal data of the writer as well as references to ink documents (see figure 4.7).

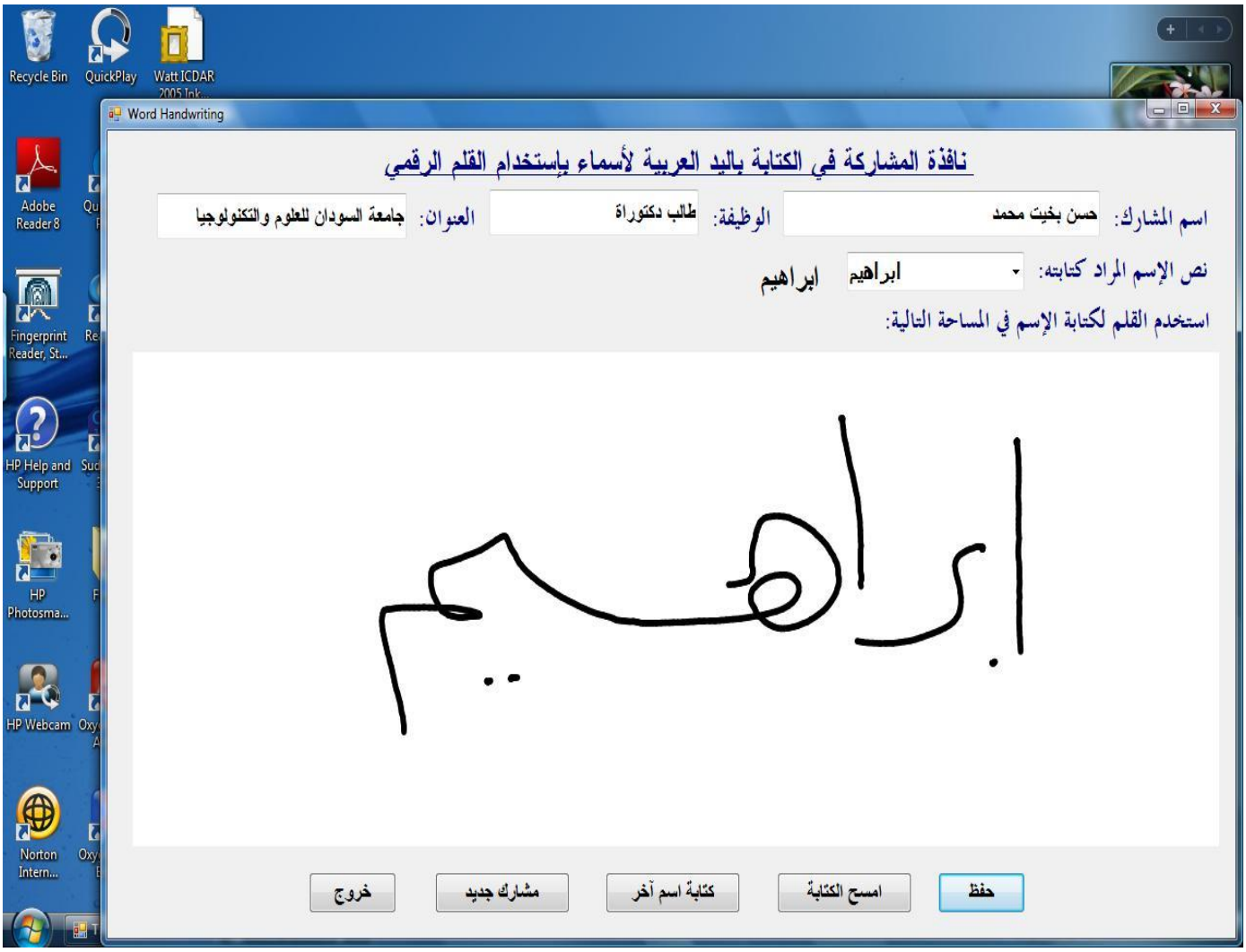

Figure 4.5: The form for capturing the handwritten words.

```
<?xml version="1.0" encoding="utf-8"?>
<ink xmlns="http://www.w3.org/2003/InkML">
   <annotation type="description">
      This document embodies an online handwriting • • • etc
   </annotation>
   <annotation type="WriterName">أ. د. يوسف حسن عبدالرحيم</annotation>
   <annotation type="WriterOccupation">مدير جامعة</annotation>
   <annotation type="WriterAddress">جامعة كرري</annotation>
   <annotation type="Text_of_Handwritten_Character">ب</annotation>
   <traceFormat>
      <channel name="X" type="integer" units="Himetrics, where each
                himetric = 0.01 mm " />
      <channel name="Y" type="integer" units="Himetrics, where each
                himetric = 0.01 mm " />
      <channel name="TimeTick" type="integer" units="MilliSeconds" />
      <channel name="PacketStatus" type="integer" units="Default" />
   </traceFormat>
   <trace id="Stroke0">
      10786 1722 0 1
      10786 1727 0 1
      10786 1727 15 1
            •
            •
            •
   </trace>
   <trace id="Stroke1">
      10968 2730 0 1
      10951 2730 0 1
      10951 2730 16 1
            •
            •
            •
   </trace>
</ink>
```

Figure 4.6: Illustration of the ink document (trace7545.inkml) which generated by the collection tool.

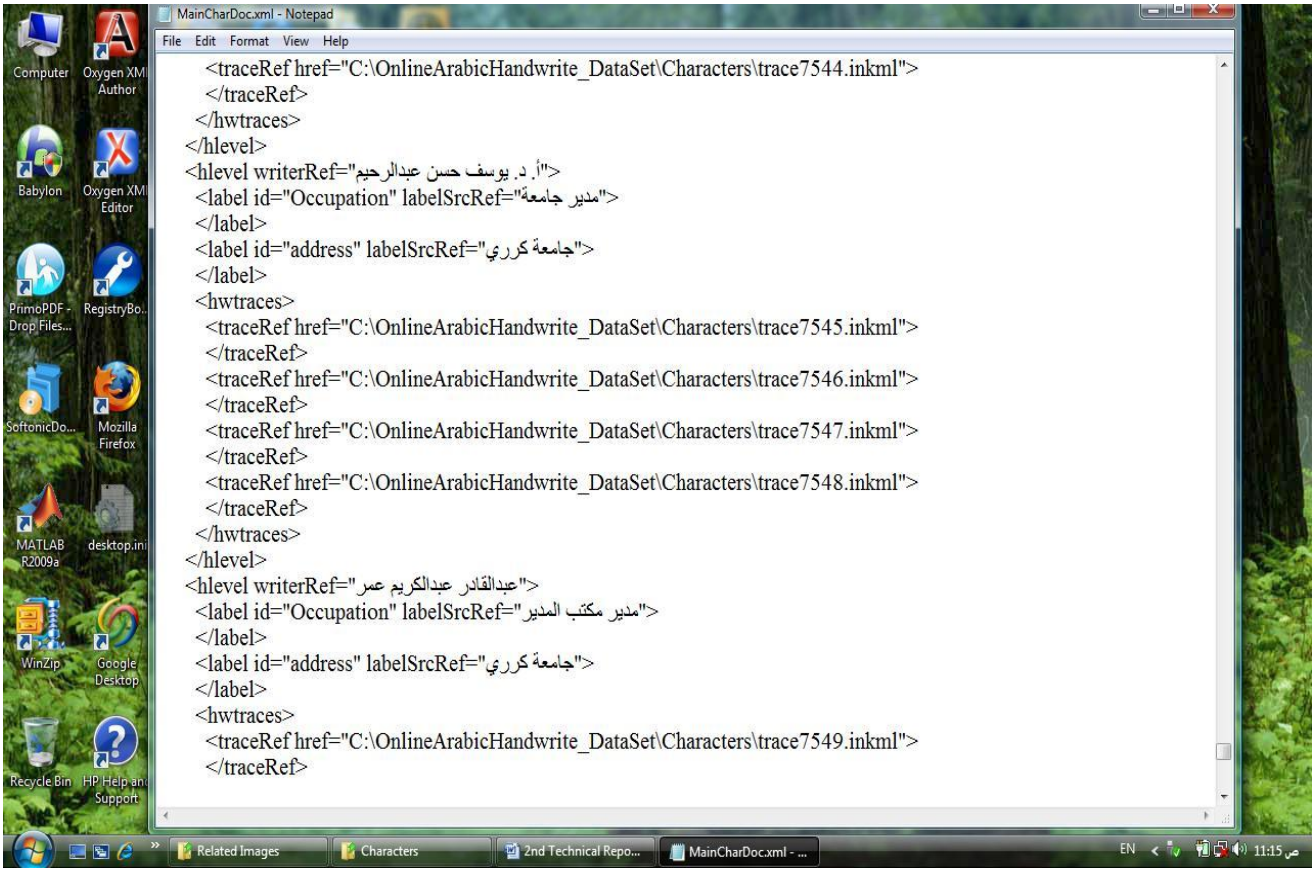

Figure 4.7: Example of *hLevel* elements which included in the main document (MainCharDoc.xml) of SUSTOLAH.

4.4 VERIFICATION TOOL

The verification tool is a software tool which mainly used to verify the content of the SUSTOLAH datasets. Contrary to the collection tool, the verification tool is a visual .net based application which can be executed on any personal computer. The verification tool can be used to review, investigate, and refine the content of the datasets. Moreover, researchers can use the tool to select specific data for the training and testing of their systems.

The verification tool has an ability to show the shape and progression for strokes of each handwritten object included in the datasets (see figure 4.8). Furthermore, it has an ability to give detailed statistics regarding with the

handwritten objects of the datasets; for example the letter "ب" has 238 handwritten objects of 2 strokes, 35 objects of 3 strokes, and 6 objects of 4. Also the tool enables the verifier to delete all the erroneous handwritten objects.

Figure 4.8: The Verification Tool of SUSTOLAH.

4.5 RESULTS AND DISCUSSION

The two datasets of SUSTOLAH can be populated with huge amount of online Arabic handwriting. At the time of writing this report, the first one of these datasets (i.e., the dataset of the letters) was populated with 7827 samples of online handwriting for isolated Arabic letters, and the other dataset (i.e., the dataset of the persons' names) was populated with 3097 samples of online Arabic handwriting for person's names. The collection tool and a Tablet PC (of the type USB TouchController) were used to fill the datasets with these samples. More than one hundred and fifty of individuals (from various high educational institutions in Sudan) have contributed in the collection of these samples. A list of twenty person's names is specified for the collection of the

samples for the dataset of the names. In the other side, the basic Arabic letters have been appointed for the collection of the samples for the dataset of the letters. Freely, we have allowed each of the individuals to choose the number and type of the handwritten objects he wants to write.

We have performed an analysis for the content of SUSTOLAH datasets. The analysis has depicted three variations concerning with the handwriting of the same objects. These variations are: strokes number variation, handwriting trajectory variation, and variation of pen movement time. Table 4.1 shows the number variation for the strokes which form the handwritten samples of some Arabic letters. Furthermore, table 4.2 shows the number variation for the strokes which form the handwritten samples of some persons' names. In addition to the variation of the strokes, the datasets characterize with a variation of handwriting trajectory and variation of pen movement time. For example, figure 4.9 illustrates a variation regarding with the handwriting trajectory for three handwritten samples of the name "سحر"; moreover, figure 4.10 depicts a variation in the pen movement time and length of the trajectory for three handwritten samples of the letter "س".Therefore, with these datasets it is possible to develop and test recognition systems of online Arabic handwriting.

In comparison with the ADAB, the online Arabic dataset which known as a standard benchmark in the ICDAR competition of 2009 [61], SUSTOLAH has the following characteristics.

- SUSTOLAH involve handwritten objects of isolated Arabic letters as well as handwritten objects of cursive Arabic words. This property assigns a pedagogical research importance to SUSTOLAH.

- The datasets have an aspectual representation for the pen tips and strokes which formulate the handwriting.

- SUSTOLAH has a software tool for the collection of online Arabic handwriting as well as a verification tool. Therefore, researchers are able to create their own datasets by these tools.

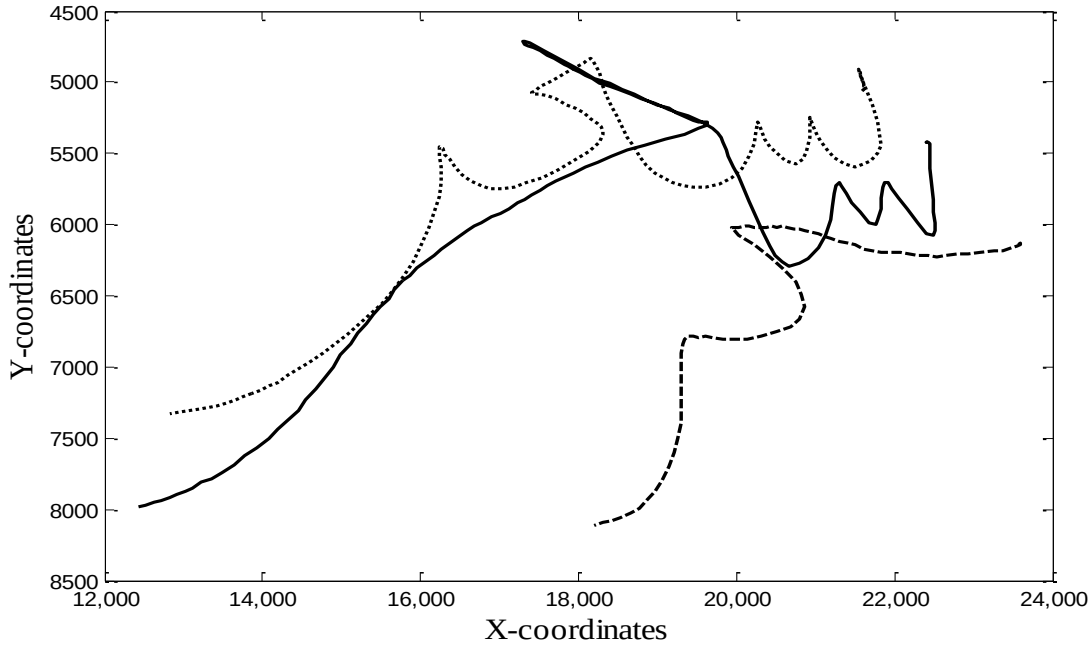

Figure 4.9: Distribution of the handwriting points for three handwritten samples of the name "سحر".

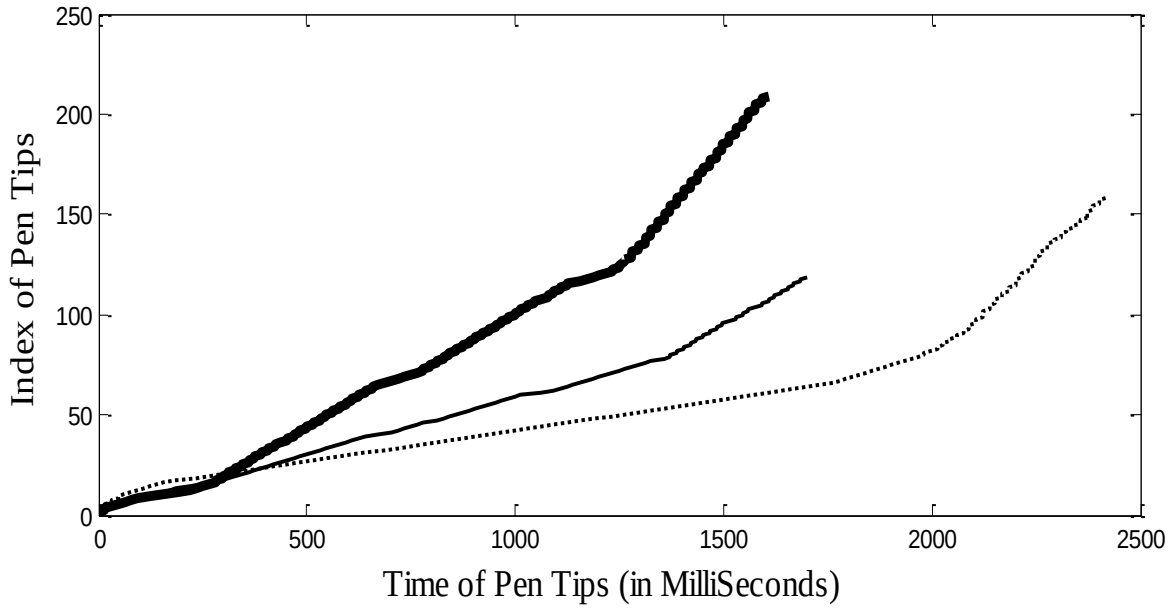

Figure 4.10: Distribution of the handwriting time for three handwritten samples of the letter "س".

Table 4.1: Strokes number variation for handwriting samples of some Arabic letters.

Letter	Number of the Handwriting Samples which take										Total Number of the Samples
	1 Stroke	2 Strokes	3 Strokes	4 Strokes	5 Strokes	6 Strokes	7 Strokes	8 Strokes	9 Strokes	10 Strokes	
ا	252	23	2	2	—	—	—	—	—	—	279
أ	1	193	54	10	3	1	1	—	—	—	263
إ	1	191	39	11	3	—	—	—	—	—	245
آ	—	180	38	4	1	1	1	—	—	—	225
ى	206	32	7	1	1	—	—	—	—	—	247
ب	—	238	35	6	—	—	—	—	—	—	279
ت	—	59	174	27	6	3	—	1	—	—	270
ث	—	27	38	163	23	3	1	—	—	—	255
ج	—	222	30	3	2	1	—	—	—	1	259
ح	214	32	4	1	—	—	—	—	—	—	251
خ	—	198	29	7	2	1	1	—	—	1	239
د	229	18	2	1	—	—	—	—	—	—	250
ذ	—	202	27	3	1	1	—	—	—	—	234
ر	222	18	—	—	—	—	—	—	—	—	240
ز	—	202	21	1	1	—	—	—	—	—	225
س	206	36	5	3	—	1	1	—	—	—	252
ش	—	20	39	146	27	6	1	—	—	—	239
ص	203	27	3	—	2	—	—	—	—	—	235
ض	—	200	32	4	—	—	—	—	—	—	236
ط	10	175	39	10	3	—	—	—	—	—	237
ظ	—	9	151	51	10	3	—	—	—	—	224
ع	194	37	5	2	—	—	—	—	—	—	238
غ	—	179	47	8	2	—	—	—	—	—	236

Table 4.2: Strokes number variation for handwriting samples of some person's names.

Name	Number of the Handwriting Samples which take										Total Number of the Samples
	1 Stroke	2 Strokes	3 Strokes	4 Strokes	5 Strokes	6 Strokes	7 Strokes	8 Strokes	9 Strokes	10 Strokes	
ابراهيم	—	—	—	—	—	27	28	26	8	8	97
ابتهال	—	—	—	—	23	35	16	3	1	—	78
احمد	—	70	30	10	1	1	—	—	—	—	112
ادريس	—	—	—	—	22	25	15	1	1	1	65
ادم	—	—	53	25	8	—	—	—	—	—	86
اسراء	—	—	—	41	11	6	2	—	—	—	60
اسماعيل	—	—	—	15	24	16	4	1	—	3	63
آلاء	—	—	1	7	34	15	4	2	—	1	64
امل	—	59	18	7	3	—	—	1	—	—	88
إيمان	—	—	—	—	4	10	17	19	7	2	59
ايناس	—	—	—	—	14	37	8	2	—	—	61
بابكر	—	—	—	—	26	14	9	5	1	3	58
بشير	—	—	—	7	11	4	32	3	—	1	58
جعفر	—	—	57	7	2	—	—	—	—	—	66
حامد	—	55	25	4	—	—	1	—	—	3	88
حسن	—	87	12	7	3	—	1	—	—	1	111
حسين	—	3	12	32	6	1	—	—	—	1	55
خالد	—	—	55	16	7	1	—	—	—	—	79
ريان	—	—	—	—	15	33	9	1	1	—	59
زينب	—	—	—	—	—	19	27	10	3	1	60
سارة	—	—	—	23	27	24	—	1	—	—	75
سحر	62	54	19	8	3	1	—	—	—	—	147
سعيد	—	27	28	6	1	—	—	—	—	—	62
سلمى	54	9	4	1	—	—	—	—	—	—	68

4.6 SUMMARY

This chapter has presented SUSTOLAH as two datasets of online Arabic handwriting. It is clearly obvious that the modern XML-Based representations are more preferable than the UNIPEN Format. Therefore, SUSTOLAH has been designed based on the UPX which is an instance of XML-Based representations. The creation of SUSTOLAH was associated with the development of two software tools (i.e., a collection tool for online Arabic handwriting and a verification tool) which respectively have been used to gather and investigate the data of SUSTOLAH. Eventually, the chapter introduced an analysis which argues that it is possible to develop and test recognition systems of online Arabic handwriting with these datasets.

CHAPTER 5
ONLINE ARABIC HANDWRITING RECOGNITIONAPPROACH BASED ON HMM

5.1 INTRODUCTION

This chapter presents the construction of a novel system approach for online Arabic handwriting recognition; furthermore, it describes the substantial performance findings of this approach.

Recently, researchers have shown an increased interest in the field of online Arabic handwriting recognition [9, 17, 62, 63]. This interest motivates us to develop this system approach. The system involves four stages which are: preprocessing stage, feature extraction, segmentation algorithm, and classifier stage (see figure 5.1). There are two properties that characterize the system compared with the other system approaches which presented in this field. These properties are:

i. An elegant segmentation algorithm named as CBD algorithm.We have designed this algorithm which segments the cursive words of online Arabic handwriting to their constituent Arabic letters. The algorithm achieves the segmentation automatically through the boundaries detection for the segments of these letters.

ii. A powerful classifier based on HMMs. HMMs with Gaussian mixture mode has been employed in the system to model the handwriting.

The training and testing phases of the system are achieved by the use of SUSTOLAH datasets which presented in chapter 4 of this thesis.

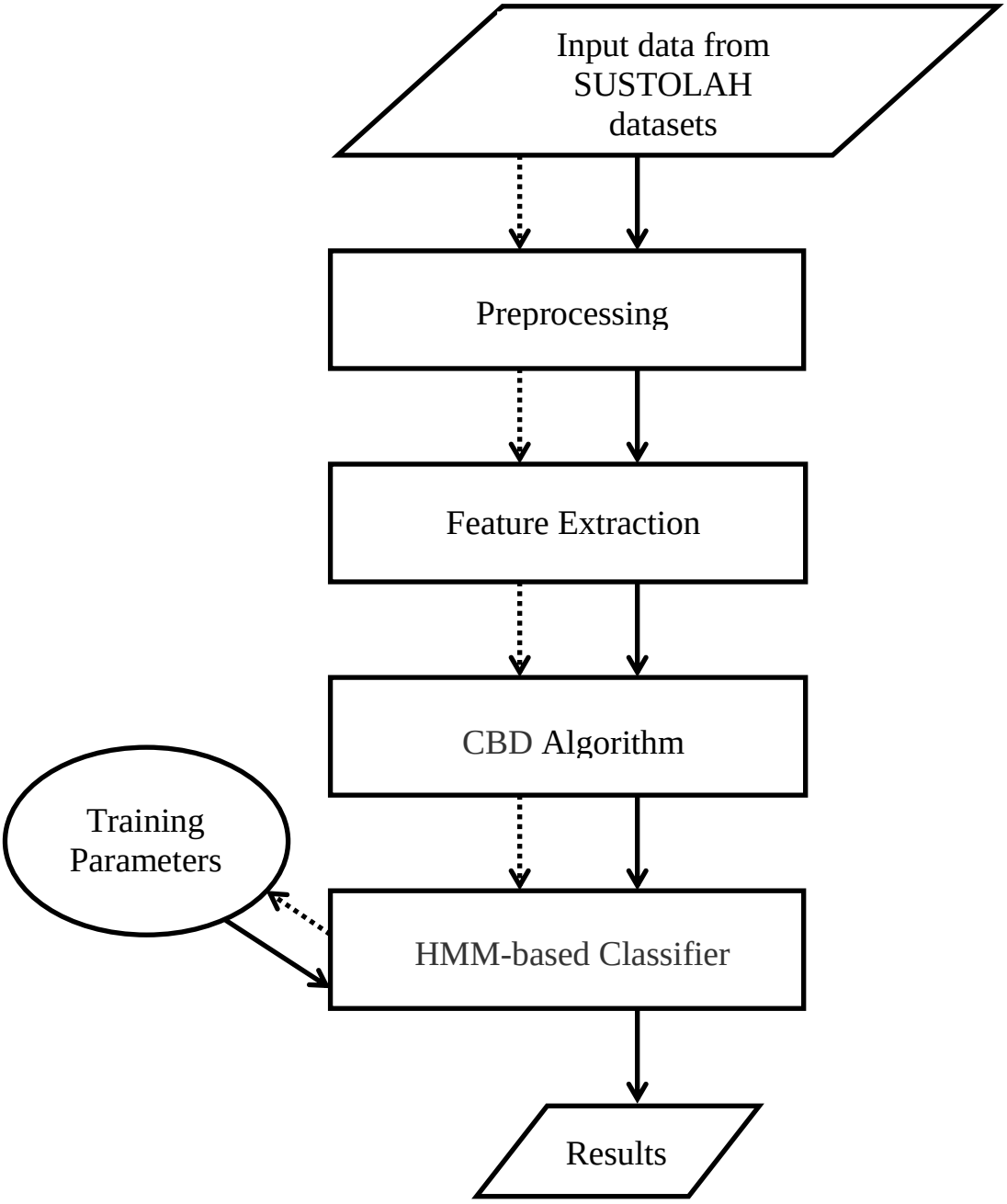

Figure 5.1: The proposed system Diagram. The dashed line arrows show the flow of data in training phase, while the solid line arrows show data flow in testing phase.

The rest of the chapter is organized as follows. Section 5.2 describes the preprocessing. Section 5.3 presents the feature extraction. Section 5.4 describes the CBD algorithm. Section 5.5 presents the classifier. Section 5.6 introduces the results and discussion of the chapter. Finally, the chapter is summarized in Section 5.7.

5.2 PREPROCESSING

Preprocessing usually addresses the problems of data reduction, elimination of imperfections, and normalization. The preprocessing of the proposed system has been implemented as a task which includes three successive stages: resampling, smoothing, and baseline detection and correction.

5.2.1 Resampling

Figure 5.2 shows the pen tip positions of a handwritten sample, for the name "سحر", which taken from SUSTOLAH. The positions are stated with actual measure units of the Tablets (i.e., himetrics).

The resampling stage is a process which transforms pen tip positions of the handwriting objects from himetrics space to pixels space (see figure 5.3). The process employs the filtering which presented in [64]. The filtering eliminates consecutive pen tips into a unique pixel as follows.

$$x_p = \frac{x_h \times 2540}{96} \qquad (5.1)$$

$$y_p = \frac{y_h \times 2540}{96} \qquad (5.2)$$

where (x_p,y_p) is a pen tip position in term of pixel location, and (x_h,y_h) is the position term of himetric location.

5.2.2 Smoothing

Smoothing is used to eliminate or reduce handwriting noises which occur by either hardware problems or erratic hand motion [8, 64]. It usually consists of averaging point position $P_i(x_i, y_i)$ with respect to their neighbors $(P_{i-n}, \dots, P_{i+n})$. The smoothing stage proposed in this work acts to substitutes

each of the pixels, which generated by the resampling stage, with the weighted average of its neighboring pixels (see figure 5.4). The substitution is employed as follows.

$$x_i' = \sum_{k=-m}^{m} \alpha_k \, x_{i+k} \qquad (5.3)$$

$$y_i' = \sum_{k=-m}^{m} \alpha_k \, y_{i+k} \qquad (5.4)$$

where $\propto_k$ is the weight on the point (x_{i+k}, y_{i+k}), and *2m* is the number of neighbor points. We set *m=3* and $\propto_k = 1/6$ in our work.

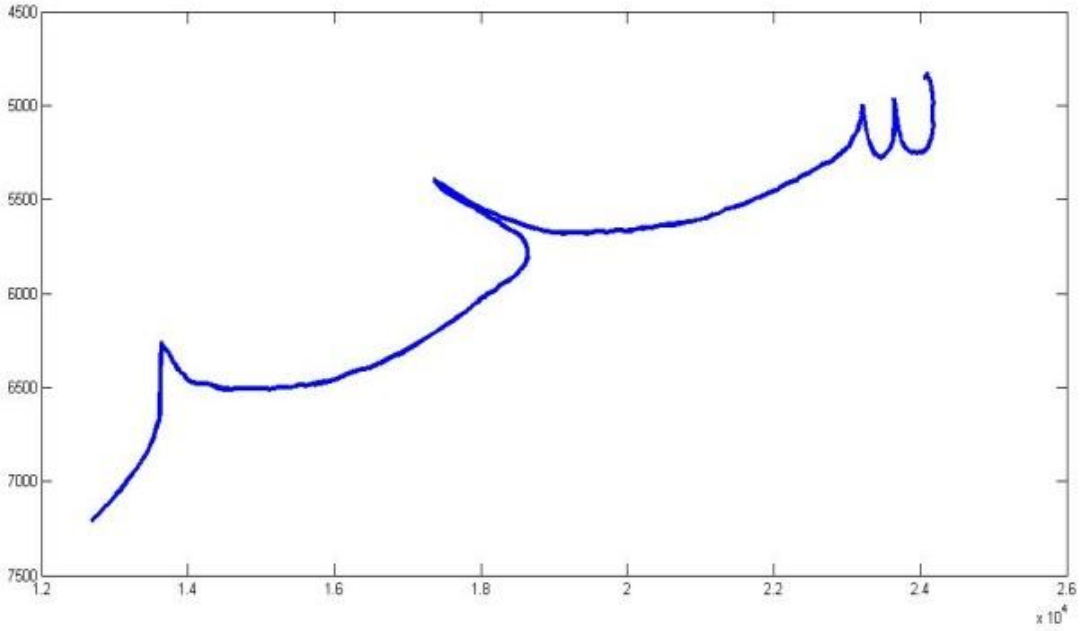

Figure 5.2: Handwriting example for the name "سحر" which taken from SUSTOLAH.

5.2.3 Baseline Detection and Correction

It is the last stage in the preprocessing task of our work. Beside the baseline detection, a process of baseline correction is included so as to adjust the orientation of the smoothed handwriting toward the horizontal level (see

figure 5.5). The stage uses the horizontal projection method which presented in [65]. This method detects the baseline by reducing the 2D of data to 1D of peaks based on the pixels of the handwriting, and the longest peak is considered to be the baseline.

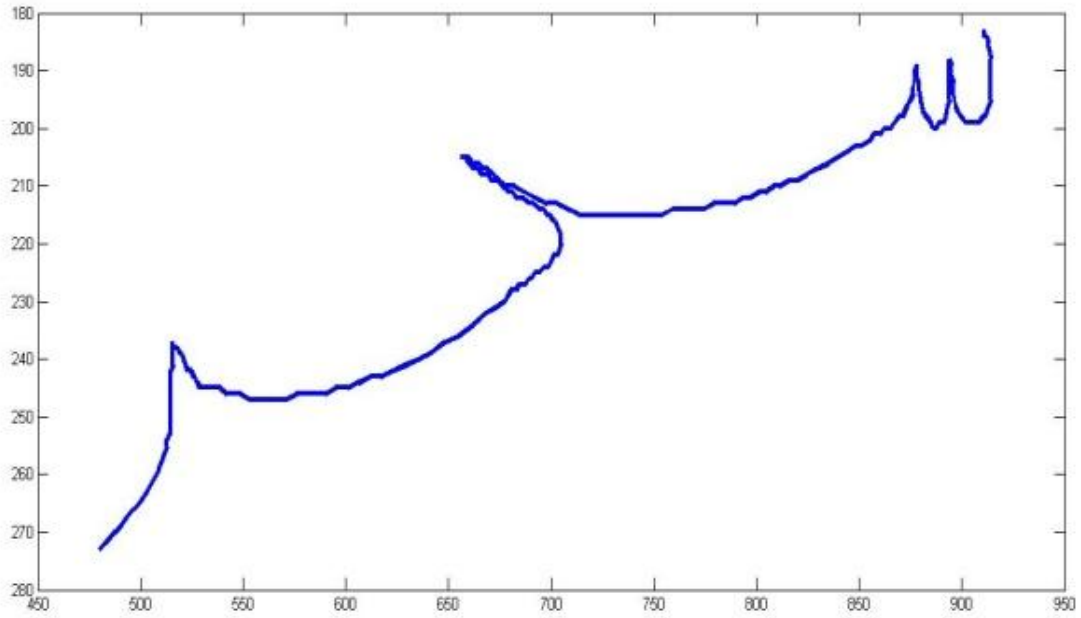

Figure 5.3: Handwriting generated after performing the resampling that by the use of figure (5.2) Handwriting.

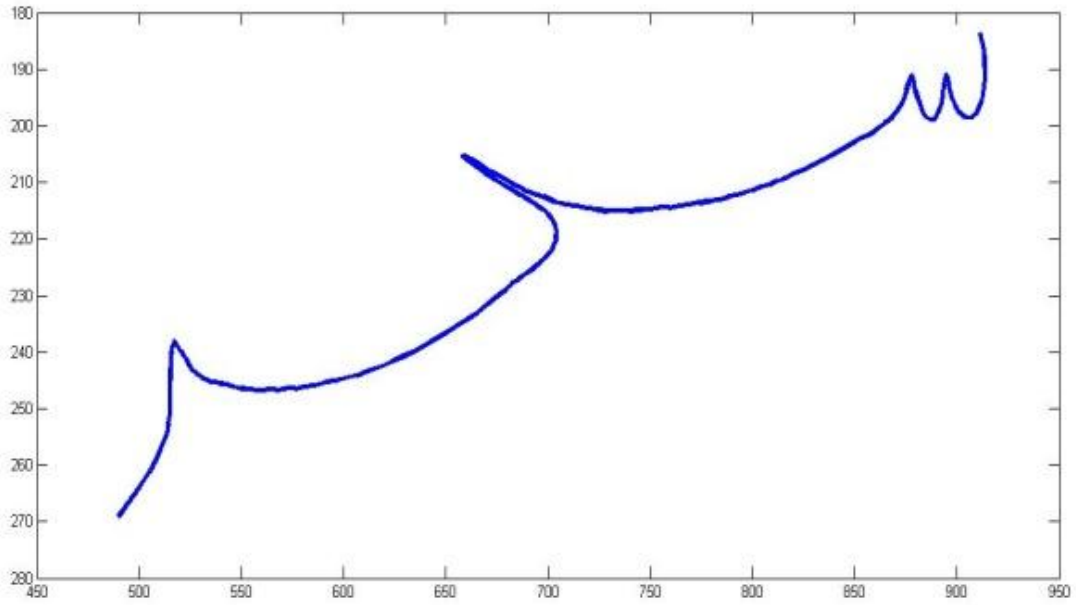

Figure 5.4: Handwriting produced after performing the smoothing by the use of figure (5.3) handwriting.

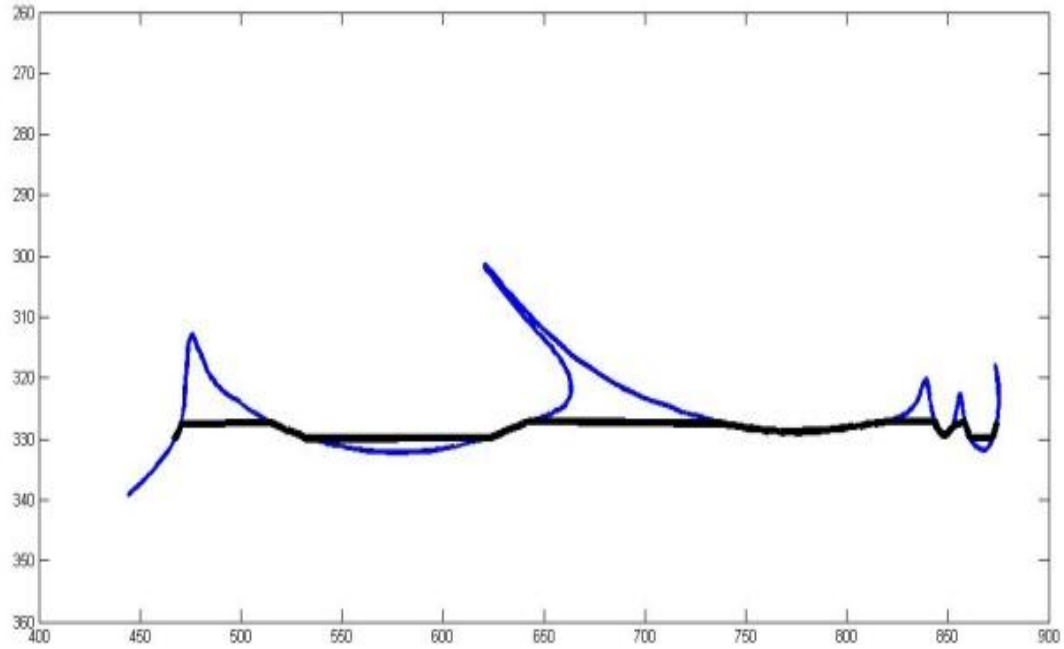

Figure 5.5: Handwriting produced after performing the baseline detection and correction by the use of figure (5.4) handwriting.

5.3 FEATURE EXTRACTION

A trajectory vector of online Arabic handwriting $P = [p_1, \dots, p_n]$ is generated by the stage of preprocessing. The proposed feature extraction derives two sequences from this vector. These sequences are: sine alpha sequence $Q = \left[sin\left(\alpha(1)\right), \dots, sin\left(\alpha(n)\right)\right]$ and cosine alpha sequence $R = \left[\cos\left(\alpha(1)\right), \dots, \cos\left(\alpha(n)\right)\right]$. These sequences enact the direction feature of the handwriting, where $sin\left(\alpha(i)\right)$ and $\cos\left(\alpha(i)\right)$ correspond to the handwriting point p_i.

This direction feature has been presented in [22]. The feature describes an online handwriting point, at an instant *t*, as follows:

$$\sin(\alpha(t)) = \frac{\Delta X(t)}{\Delta S(t)} \qquad (5.5)$$

$$\cos(\alpha(t)) = \frac{\Delta Y(t)}{\Delta S(t)} \qquad (5.6)$$

where $\Delta X(t)$, $\Delta Y(t)$, and $\Delta S(t)$ are in turn defined as follows:

$$\Delta X(t) = X(t-1) - X(t+1) \qquad (5.7)$$

$$\Delta Y(t) = Y(t-1) - Y(t+1) \qquad (5.8)$$

$$\Delta S(t) = \sqrt{\Delta X^2(t) + \Delta Y^2(t)} \qquad (5.9)$$

5.4 CBD ALGORITHM

The CBD algorithm is used to detect the boundaries of the handwritten Arabic letters those laid within those sequences of sine alpha and cosine alpha. The sequences enclose real values which represent the direction features of the online Arabic handwritten words corresponding to the sequences. These boundaries indicate the segmentation points of the letters.

To achieve its function, the algorithm takes the two sequences of the handwriting direction (*Q* and *R*), which generated by the feature extraction stage, as inputs.The algorithm involves two stages: the first one is a stage of specifying the straight writing lines which included in the given online handwriting. These lines are classified into three types: right to left horizontal lines, bottom up vertical lines, and up down vertical lines. The second stage detects the beginning and end boundaries of the letters which composed the handwriting.The following two sections (section 5.4.1 and section 5.4.2) describe the first and second stage of the algorithm.

5.4.1 First Stage of the Algorithm

The stage explores the cosine alpha sequence *R* and sine alpha sequence *Q* for the subsequences which enact the straight writing lines. Let *S* and *T* denote the subsequences.

$$R = \{r_1, r_2, \cdots, r_n\}, \tag{5.10}$$

$$Q = \{q_1, q_2, \cdots, q_n\}, \tag{5.11}$$

where r_i is a sine alpha value and q_j is a cosine alpha value.

S and *T* are defined as follows:

$$S \subset R = \left\{ \langle s_1,\ s_2, \cdots,\ s_m \rangle \middle| \begin{matrix} (s_1 = r_i), \\ (s_2 = r_{i+1}), \\ \cdots, \\ (s_m = r_{i+m}) \end{matrix} \right\} \tag{5.12}$$

$$T \subset Q = \left\{ \langle t_1,\ t_2, \cdots,\ t_p \rangle \middle| \begin{matrix} (t_1 = q_j), \\ (t_2 = t_{j+1}), \\ \cdots, \\ (t_p = q_{j+p}) \end{matrix} \right\} \tag{5.13}$$

such that $1 \leq m,\ p \leq n$

The type of right to left horizontal lines is defined as S so that $\forall x \in S\ (0.8 \leq x \leq 1.0)$. The type of bottom up vertical lines is denoted by the sequence T where $\forall y \in T (0.8 \leq y \leq 1.0)$, whereas the sequence *T*, which constrained by $\forall y \in T (-0.8 \leq y \leq -1.0)$, identifies the type of up down vertical lines.

Eventually, the stage maintains the positions of the writing lines, those included in the handwriting, in three vectors: vector of begin and end positions for right to left horizontal lines, vector of begin and end positions for bottom up vertical lines, and vector of begin and end positions for up down vertical lines.

5.4.2 Second Stage of the Algorithm

To detect the end boundaries for the handwritten letters, the stage scans the vector of right to left horizontal lines based on the standard numbers of

these lines in Arabic letters. The first point of the handwriting indicates the beginning boundary for the first letter of the handwriting, whereas the beginning boundary for each of the letters is identified after the end boundary of the preceded letter.

5.5 CLASSIFICATION

The classifier of the recognition system is designed based on HMMs with Gaussian-mixture model. The classifier goes through two phases: training and testing. In the training phase, the CBD algorithm supplies the classifier with exact letter segments of the written words. However, the algorithm offers may alternatives of segments in the testing phase. The HMMs are generated so as to model these segments. These segments are classified into two types of vector: vector of sine alpha values and vectors of cosine alpha values.

The classifier creates a state for each of the segments. Therefore, a left-to-right sequence of N states is generated for an online handwritten word which consists of N letters (see figure 5.6). These states are regarded with observation probability distribution which represented by a mixture Gaussian density as follows.

The observation probability $b_j(o_t)$ for state j of the generating observation o_t is given by:

$$b_j(o_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} \Psi\left(o_{st}; \mu_{jsm}, \Sigma_{jsm}\right) \right]^{\gamma_s} \qquad (5.14)$$

where M_{js} is the number of mixture components in state j for stream s; The exponent γ_s is a stream weight and its default value is one; $c_{jsm}$is the weight (prior probability) of the m^{th}component; and $\Psi(o; \mu, \Sigma)$ is a multivariate Gaussian with mean vector μ and covariance matrix Σ, that is

$$\Psi(o;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{\frac{1}{2}(o-\mu)^T\Sigma^{-1}(o-\mu)} \tag{5.15}$$

where n is the dimensionality of o.

As a convention, it is also assume that there is an initial state for the first letter of the word, where this state is described by the initial probability distribution over states at time 0. States transition is parameterized by defining a state transition matrix, where the value $a_{i,j}$ is the probability of transitioning from state i to state j at any time t.

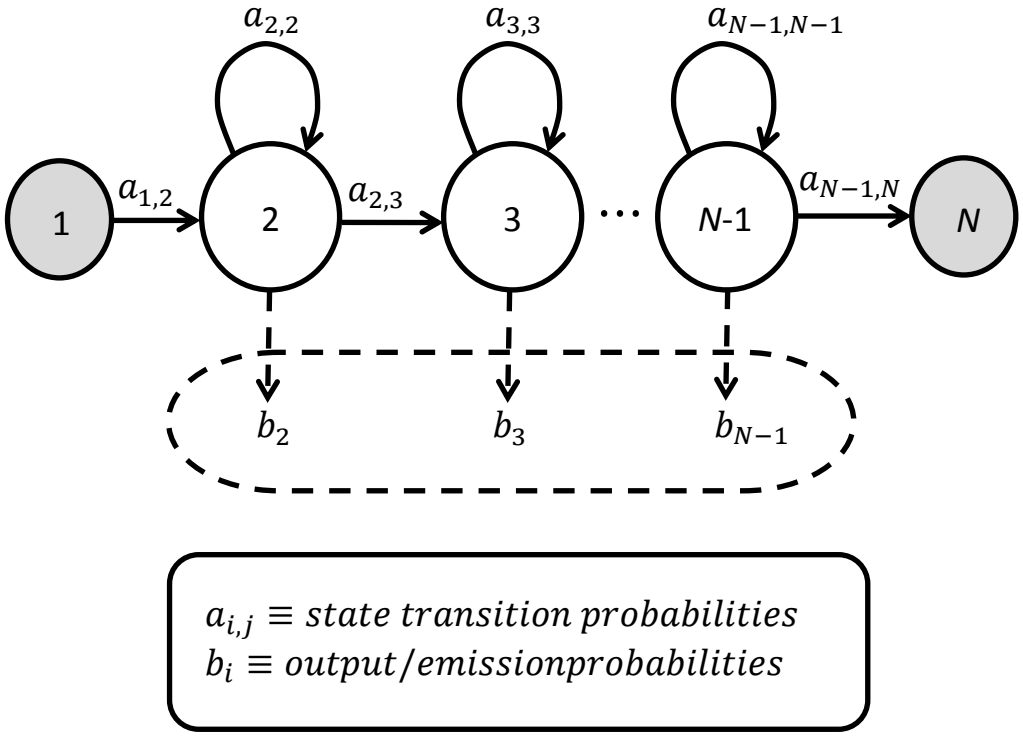

Figure 5.6: Left-to-right sequence of states.

5.5.1 Training

The training involves a supervised learning which employs data that includes both the input and the desired results. The classifier organizes the segments, which provided by the CBD algorithm, in a group of distinct classes

where each of the classes is uniquely labeled by its corresponding Arabic letter.

As it has been defined in section 2.3.3, the observation probability is a density function which governed by a set of parameters. These parameters are: the number of Gaussian-mixture components, mean (μ), and covariance (Σ) of the density. In the training we employ the EM algorithm to find the maximum-likelihood estimate of the parameters of an underlying distribution from these classes. For each of the classes, the parameters are obtained by cycling through possible values and selecting those producing the highest classification rate.

5.5.2 Testing

In the testing phase we employ a search algorithm which comprises the aspects of the Viterbi algorithm. It acts to finds the most likely path through a graph of states given a set of observations. These states are the building blocks of the HMMs which generated for the given online Arabic handwritten words, whereas the observations are given by the segments which presented by the CBD algorithm.

The algorithm looks at each state at time t, and for all sequences of the states which lead to the state, then it decides which of the sequences was the most likely to occur (i.e., the sequence with the greatest log likelihood). The algorithm is defined as follows.

Initialization

$t = 0$;

For all n, where $1 \leq n \leq N$

$\Gamma_{n0} := \ln \pi_n$;

$sp_{n0} := [X_0]$;

End For;

Calculation

For all t, where $1 \leq t \leq N$

For all n, where $1 \leq n \leq N$

For all m, where $1 \leq m \leq N$

$\Gamma_{nt} := Max(\Gamma_{mt-1} + \ln a_{nm} + \ln b_n)$

End For;

$sp_{nt} := Append(X_n, sp_{mt})\ such\ that$

$\Gamma_{mt-1} + \ln a_{nm} + \ln b_n = \Gamma_{nt};$

End For;

End For;

Decision

$If\ t = T\ then$

For all n, where $1 \leq n \leq N$

$\Gamma_T := Max(\Gamma_{nt})$

End For;

$sp_T := sp_{nt}\ such\ that\ \Gamma_{nt} = \Gamma_T$

where the symbols involved in the algorithm are defined as follows.

t	The discrete time index.
N	The total number of states.
X_n	The *n*th state.
o_t	The observation symbol at time t, which can be one of M different symbols.
sp_{nt}	The survivor path which terminates at time t, in the nth state of the search graph. It consists of an ordered list of X_n's visited by this path from time $t = 0$ to time t.
T	Truncation length of the algorithm (i.e. the time when a decision has to be made by the algorithm as to which sp_{nt} is the most likely).
π_n	The probability that the *n*th state at $t = 0$ is the most likely.
a_{nm}	The transition probability for the transition from state X_m at time $t - 1$ to the state X_n at time t.
b_n	The observation probability at time t, for state X_n.
Γ_{nt}	The survivor path probability ofsp_{nt}.

5.6 RESULTS AND DISCUSSION

In order to evaluate the performance of the system, a number of experiments have been carried using the MATLAB environment. The XMLTree (XML toolbox for MATLAB) is used to import and access the data

of SUSTOLAH. Furthermore, the classifier of the system is implemented using the Kevin Murphy's HMM Toolbox for Matlab. The system is evaluated by 100 online handwritten samples, taken from SUSTOLAH, for the names ("سلمى", "سحر") where each of these samples involves a single stroke. A restriction which enforces a specific writing style is stated for these samples. The restriction specifies the number of right to left handwritten lines for the handwritten samples of the same Arabic letters. For instance, all handwriting samples of the letter 'س' are characterized with the number 3 of the lines.

The CBD algorithm has introduced well done boundaries detection for the letters of these samples (see figure 5.7). The evaluation of the algorithm has shown a novel method that achieves the detection more efficiently than the other segmentation methods which mentioned in section 3.6.

A group of twenty experiments was prepared to be performed using these handwritten samples. Each of the experiments involves a process of training followed by a testing process for the recognition system. A distinct batch of these samples was outfitted for each of the experiments; that is, a diverse combination, which involves five of the samples, was dedicated as a testing data for the experiment, whereas the remaining of the samples (i.e., the resting 95 samples) were specified for the training of the experiment. That is to say 20 fold cross validation has been used.

The group of the twenty experiments was performed three times where a different arrangement of features for the handwritten segments, which introduced by the CBD algorithm, is employed in each of these times. Therefore, three different arrangements of the handwritten features were employed in this performance of the experiments. The arrangements are: 1st arrangement, 2nd arrangement, and 3rd arrangement. The 1st arrangement is constructed by the first ten feature values of the handwritten segments, the 2nd

arrangement is constructed by the first twenty feature values of the segments, and the 3rd arrangement is constructed by the first thirty feature values of the segments.

For the performed three experiments sets, the number 8 was found to be the best number of mixture components. Moreover, the number 12 was set to be the maximum number of iterations for the EM training. Primary experiments have been conducted to find the best number of mixture components.

The training has shown that the convergence of the EM algorithm is occurred within the range 2─5 of iterations (see figure 5.8). Figure 5.9 shows the recognition results of the performed three experiments sets. The results have shown a good performance which involves recognition rates that reached 81.7% for the 1st arrangement of the handwritten segments, 86.8% for the 2nd arrangement of the segments, and 90% for the 3rd arrangement of the segments.

In their major study, H. Ahmed and S. A. Azeem (2011) conclude that the careful choice of the HMM parameters have significantly improved the performance of their proposed system over other HMM-based systems [22]. On the other hand, the results of our experiments demonstrate some fluctuations of the recognition rates over the different employed arrangements of those segments feature vector. Therefore, the performance of our proposed system is influenced by the arrangement of the feature vector; such that the well done choice of the arrangement has enhanced the performance.

Much effort is required to obtain better arrangements of the feature vector to be used for the recognition. Selection of values for different parts of the segments can be treated to get the better arrangements.

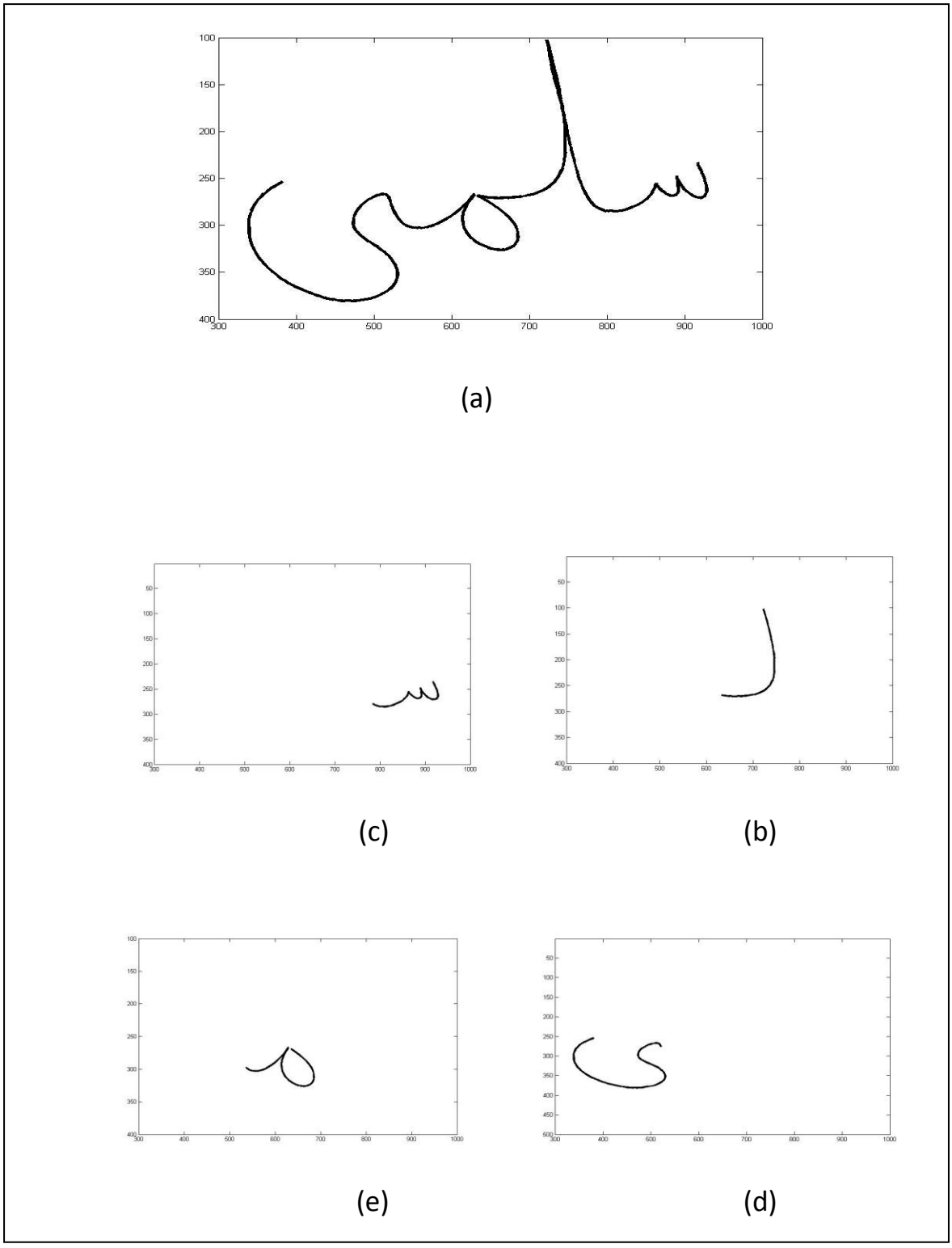

Figure 5.7: Segmentation example of an online handwriting for the name 'سلمی'. Part (a) of the figure shows the name, whereas the parts (b, c, d, and e) show the handwriting segments.

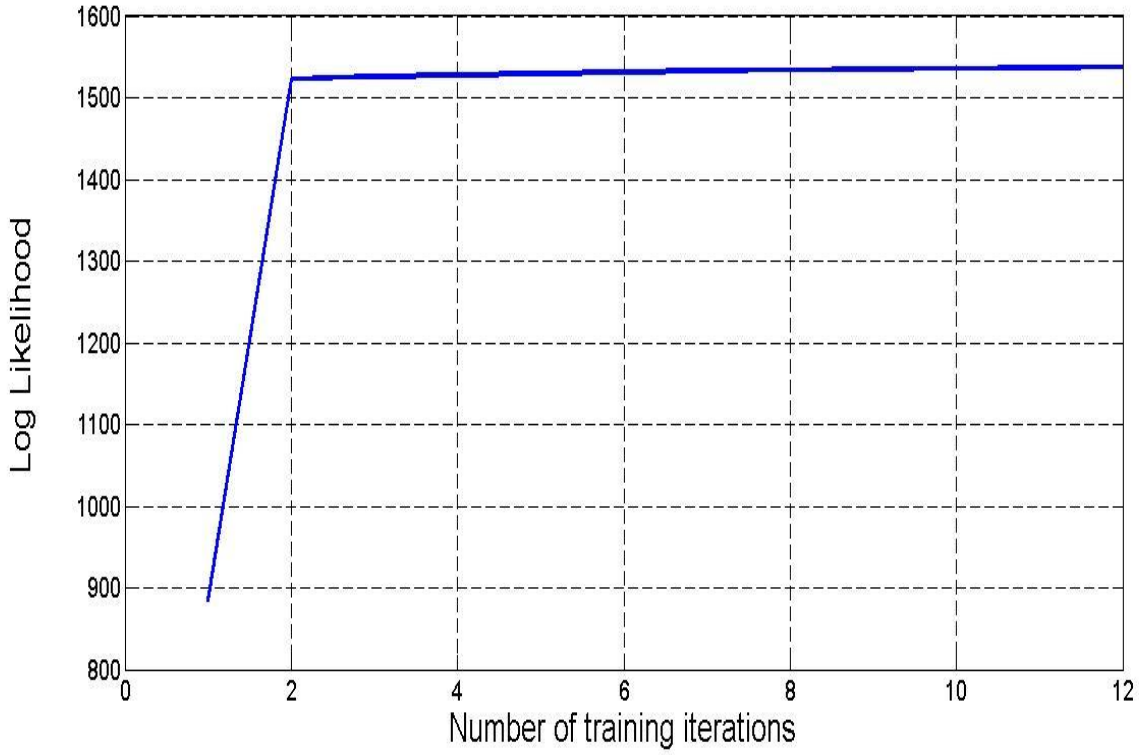

Figure 5.8: Illustration of the EM convergence for the handwritten samples of the letter "ر".

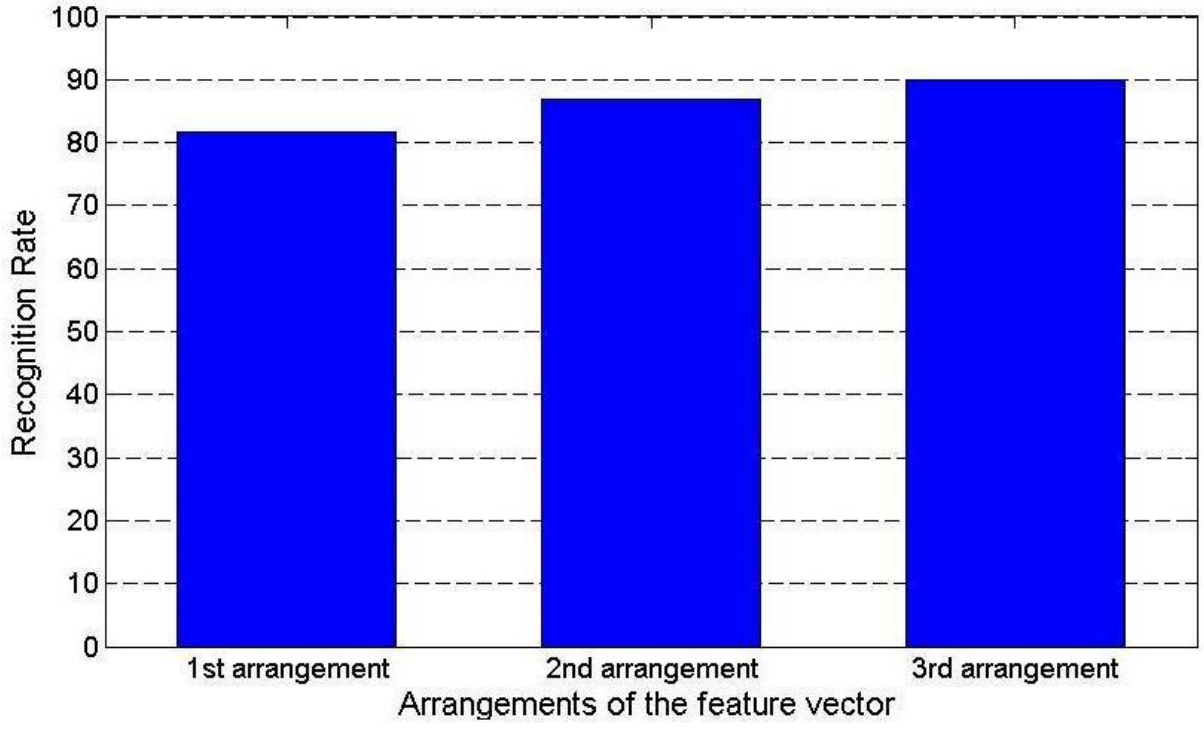

Figure 5.9: Illustration of the recognition rates of the performed experiments.

5.7 SUMMARY

A novel system approach of online Arabic handwriting recognition was presented in this chapter.The approach has characterized with the use of HMMs as well as a segmentation process.

The structure of the system has involved four stages: preprocessing, feature extraction, segmentation, and classification. The preprocessing was achieved by applying the steps of resampling, smoothing, and baseline detection and correction to the given handwritten samples. Considered candidate features were formed by the feature extraction stage. The features have represented the direction of the handwritten samples. A new wonderful algorithm, called the CBD algorithm, has been designed to be used in the segmentation stage. Concerning with the classification, HMMs with Gaussian mixture were employed for addressing the task of the recognition.

By using data from SUSTOLAH, a set of experiments was carried out over the system. The results of these experiments have shown a good performance. The performance is influenced by the arrangement of the feature vector; such that the well done choice of the arrangement has enhanced the performance.

CHAPTER 6
CONCLUSION

This thesis has presented a novel approach of online Arabic handwriting recognition as well as datasets for online Arabic handwriting. Three categories of research objectives were formulated in chapter 1:

- **Prior knowledge**: Understanding of prior knowledge which used in the construction and evaluation of the handwriting recognition.
- **Applicability**: Demonstration of online Arabic handwriting recognition to be learned by the approaches of pattern recognition.
- **Interpretability**: The outcomes from the creation of online Arabic handwriting datasets and the design of a novel system approach for the handwriting recognition.

These three categories return throughout this thesis: prior knowledge (chapter 2), applicability (chapter 3), and interpretability (chapters 4─5). A research was formulated to achieve these objectives as follows.

6.1 PRIOR KNOWLEDE

As prior knowledge, chapter 2 presented theoretical background for the typical system structure of online Arabic handwriting recognition, HMMs, and representation methods of online handwriting datasets. Firstly, the chapter explored the stages of the typical recognition system for online Arabic. These stages are: data acquisition, preprocessing, segmentation, feature extraction, classification, and post processing. Then the chapter has demonstrated the elements and computation problems of HMMs. Eventually, the chapter has reported the two representation methods (i.e., UNIPEN format and XML-representations) which have been used to format the datasets of online

handwriting, such that UNIPEN format was the first of these methods but it has been succeed by the XML-based representations.

6.2 APPLICABILITY

The review in chapter 3 showed that many researchers have attempted to apply different approaches in the several aspects for their studies of online Arabic handwriting recognition. As classifier approaches, the researchers used the structural method, evolutionary neuro-fuzzy, Kohonent neural network, DTW followed by simple neural, and HMMs approach. Moreover, they have employed several sets of the handwriting features such as the freeman code, local point features, features of dynamic representations, and features of handwriting motor models. It is clear that there are no standard datasets were used in the training and testing experiments of these studies. In conclusion, the method of HMMs seems to be the recent applicable approach for online Arabic handwriting recognition. Nevertheless, the segmentation issue has been a challenging problem especially for those researchers who employ HMMs the handwriting recognition.

6.3 INTERPRETABILITY

SUSTOLAH which presented in chapter 4 has introduced datasets to be used in the field of online Arabic handwriting recognition.The analysis which performed for the datasets argues that the datasets can successfully be used to train and test the recognition systems of the handwriting. SUSTOLAH was designed based on the UPX which is an instance of XML-Based representations. The creation of SUSTOLAH was achieved by two software tools (i.e., a collection tool for online Arabic handwriting and a verification tool for this handwriting) which respectively have been used to gather and investigate the data of SUSTOLAH.

A good performance of online Arabic handwriting recognition was presented in chapter 5. To get this performance, a novel system approach for the handwriting recognition was firstly developed, and secondly a set of experiments for this approach were executed by the use of SUSTOLAH. The structure of the system has involved four stages: preprocessing, feature extraction, segmentation, and classification. The preprocessing was achieved by applying the steps of resampling, smoothing, and baseline detection and correction to the given handwritten samples. Considered candidate features were formed by the feature extraction which generated some features from these samples. A wonderful algorithm, called a CBD algorithm, has been designed to achieve the segmentation by producing segments of Arabic letters for the given samples. Concerning with the classification, HMMs with Gaussian mixture were employed to address the task of recognition.

6.4 SUGGESTIONS FOR FUTURE RESEARCH

This thesis has made contributions to the research knowledge in the domain of online Arabic handwriting recognition. Nevertheless; several outstanding issues remain to be explored. Furthermore, the study of the thesis could be extended in several directions. These issues and directions formalize suggestions for future research of the domain. The following list presents these suggestions.

- **Competition of online Arabic handwriting recognition**. Researchers can use the datasets which presented in chapter 4 as a platform for future work in the area of online Arabic handwriting recognition. These datasets should be used to test algorithms and recognition methods so as to give a push in this area.

- **Delayed strokes handling**. In many word recognition systems delayed strokes have been more naively handled by removing them and replacing them by a feature in the base shape sequence and this could potentially cause problems for diacritic intensive scripts [10]. The CBD algorithm presented in chapter 5 could be enhanced so as to address the delayed strokes as well as the handwriting which formed with multiple overlapping strokes. Given a vast wealth of literature regarding with the segmentation techniques of online cursive handwriting and the heuristic strategies, it is anticipated that the algorithm may be of general use to improve the performance for the systems of online handwriting recognition.

- **Employment of post processing**. By using more rigorous post processing, we can achieve higher accuracy for the recognition of online Arabic handwriting.

- **Geometric computation techniques**. In addition to the post processing phase, geometric-computation techniques can be used to reduce the number of errors for the recognition.

- **Transform techniques**. Much effort is required to obtain better arrangements of the feature vector so as to be used for the HMM based approach which presented in chapter 5. The transform techniques, such as Fourier Transform, should further be applied to get these arrangements.

REFERENCES

[1] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 63–84, 2000.

[2] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2007.

[3] A. K. Jain, R. P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," IEEE Transactionson Pattern Analysisand Machine Intelligence, vol. 22, no. 1, Jan 2000.

[4] S. Al-Emami and M. Usher, "On-line recognition of handwritten Arabic characters," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 704–710, July 1990.

[5] A. M. Alimi, "An evolutionary neuro-fuzzy approach to recognize on-line Arabic handwriting," In Proceedings of the 4th International Conference Document Analysis and Recognition, pages 382–386, 1997.

[6] T.S. El-Sheikh and S.G. El-Taweel, "Real-time Arabic handwritten character recognition," Pattern Recognition, vol. 23, no. 12, pp. 1323-1332, 1990.

[7] N. Mezghani, A. Mitiche, and M. Cheriet, "On-line recognition of handwritten Arabic characters using a kohonen neural network," In Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, page 490, Washington DC, 2002.

[8] B. Alsallakh and H. Safadi, " AraPen: An Arabic Online Handwriting Recognition System," In Proceedings of 2nd IEEE International

Conference on Information & Communication Technologies: from Theory to Applications 2006 (ICTTA apds 06), vol. 1, pp. 1844–1849, Damascus, 24 - 28 April 2006.

[9] M. Al-Ammar, R. Al-Majed, and H. Aboalsamh, "Online Handwriting Recognition for the Arabic Letter Set," In Proceedings of the 5th WSEAS international conference on Communications and information technology, Corfu Island, Greece, 14-17 July 2011.

[10] J. Sternby, J. Morwing, J. Andersson, and C. Friberg, "On-line Arabic handwriting recognition with templates," Pattern Recognition, vol. 42, no. 12 , December 2009.

[11] M. Nakagawa and K. Matsumoto, "Collection of On-line Handwritten Japanese Character Pattern Databases and their Analysis," International Journal on Document Analysis and Recognition, vol. 7, no. 1, 2004.

[12] S. Jaeger, M. Nakagawa, "Two On-line Japanese Character Databases in Unipen Format," In Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR), Seattle (USA), pages 566–570, 2001.

[13] H. Shu, "On-line Handwriting Recognition Using Hidden Markov Models," Master's thesis, Massachusetts Institute of Technology, 1996.

[14] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," In Proceedings of IEEE, vol. 77, no. 2, pp. 257-286, Feb 1989.

[15] K. C. Santosh, and C. Nattee, "A Comprehensive Survey on Online Handwriting Recognition Technology and Its Real Application to the

Nepalese Natural Handwriting," Kathmandu University Journal of Science, Engineering, and Technology,vol. 5, no. 1, pp. 31-55, 2009.

[16] R. Saabni and J. El-Sana," Hierarchical On-line Arabic Handwriting Recognition," In Proceeding of 10th International Conference on Document Analysis and Recognition (ICDAR 2009), Pages 867 – 871, Barcelona, Spain, 26-29 July 2009.

[17] M. Kherallah, L. Haddad, and A. M. Alimi," A new Approach for Online Arabic Handwriting Recognition," In Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt, 22-23 April 2009.

[18] G. Al-Habian and K. Assaleh, "Online Arabic Handwriting Recognitionusing Continuous Gaussian Mixture HMMS," In Proceeding of International Conference on Intelligent Advanced System (ICIAS 2007), vol. 1, pp. 1183 – 1186, Kuala Lumpur, Malaysia, 25-28 Nov 2007.

[19] H. Beigi, K. Nathan, G. J. Clary, and J. Subrahmonia, " Challenges of Handwriting Recognition in Farsi, Arabic and Other Languages with Similar Writing Styles An On-line Digit Recognizer," In Proceedings of the 2nd Annual Conference on Technological Advancements in Developing Countries, 1994.

[20] F. O. Deborah, O. E. Olusayo, and F. O. Alade, "Development of a Feature Extraction Technique for Online Character Recognition System," IISTE (International Institute for Science, Technology & Education), Transactions on Innovative Systems Design and Engineering (ISDE), vol. 3, no. 3, 2012.

[21] F. Biadsy, J. El-sana, and N. Habash, "Online Arabic handwriting recognition using hidden markov models," In Proceeding of 10th International Workshop on Frontiers of Handwriting Recognition, La Baule, France, 23-26 Oct 2006.

[22] H. Ahmed and S. A. Azeem, "On-line Arabic Handwriting Recognition System based on HMM," In Proceeding of 11th International Conference on Document Analysis and Recognition (ICDAR 2011), Pages 1324 – 1328, Beijing, China, 18-21September 2011.

[23] G. F. Luger, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving," Pearson Education Limited, Harlow, England, Fifth Edition, 2005.

[24] M. Flasinski, "Mathematical Linguistics Models for Computer Vision," Machine Graphics& Vision, vol. 5, ½, 1996, pp. 87-97.

[25] K. S. Fu, "Syntactic Pattern Recognition and Applications," Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[26] K. S. Ray and J. Ghoshal, "Approximate reasoning approach to pattern recognition," Fuzzy Sets and Systems, vol. 77, no. 2, pp.125-150, January 1996.

[27] J. F. Martínez-Thrinidad and A. Guzmán-Arenas, "The logical combinatorial approach to pattern recognition: an overview through selected works," Pattern Recognition, vol. 34, no. 4, pp. 741-751, April 2001.

[28] H. Byun and S. W. Lee, "Applications of Support Vector Machines for Pattern Recognition: A Survey," SVM 2002, LNCS 2388, pp. 213-236, 2002.

[29] J. Liu, J. Sun, and S. Wang, " Pattern Recognition: An overview," IJCSNS International Journal of Computer Science and Network Security, vol. 6, no.6, June 2006.

[30] A. Amin, A. Kaced, P. J. Haton, and R. Mohr, "Handwritten Arabic Character Recognition by the IRAC System," Proc. Int. Conf. Pattern Recognition, Miami, Florida, USA, pp. 729-731, 1980.

[31] M. S. El-Wakil and A. A. Shoukry, “On-Line Recognition of Handwritten Isolated Arabic Characters,” Pattern Recognition, vol. 22, no. 2, pp. 97-105, 1989.

[32] A. M. Alimi and A. O. Ghorbel, "Error Analysis in an On-Line Recognition System of Arabic Handwritten Characters," Proc. Int. Conf. Document Analysis and Recognition: ICDAR’95, Montreal, Canada, Aug., pp. 671-674, 1995.

[33] A. Amin, "Machine Recognition of Handwritten Arabic Words by the IRAC I1 System," Proceedings of the 6th International Joint Conference on Pattern Recognition, Munich, F.R.G., Oct., pp. 34-36, 1982.

[34] A. Amin and G. Masini, "Machine Recognition of Cursive Arabic Words," SPIE’s vol. 359 Application of Digital Image Processing IV, San Diego, CA, Aug., pp. 286-292, 1982.

[35] A. Amin, G. Masini, and P. J. Haton, "Recognition of Handwritten Arabic Words and Sentences," Proceedings of the 7th International Joint Conference on Pattern Recognition, Montr & l, Canada, Oct., pp. 1055-1057, 1984.

[36] G. Farias, J. Vega, R. Dormido, J. Sanchez, N. Duro, M. Santos, J. A. Martin, and G. Pajares, "Structural pattern recognition approach," In Trans. of Review of Scientific Instruments,vol. 77, no. 10, Oct 2006.

[37] A. Belaid and J. P. Haton, “A syntatic approach for handwritten mathematical formula recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence., vol. PAMI-6, no. 1, pp. 105 – 111, Jan. 1984.

[38] C. C. Tappert, "Cursive Script Recognition by Elastic Matching," IBMJ. RES. DEVELOP., vol. 26, no. 6, 1982.

[39] L. R. Rabiner and B.H. Juang, "An introduction to Hidden Markov Models," IEEE ASSP Mag., pp 4-16, June 1986.

[40] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, February 1989.

[41] H. El-Abed, V. Margner, M. Kherallah, and A. M. Alimi , “ICDAR 2009 Online Arabic handwriting recognition competition,” In Proceedings of 10th International Conference on Document Analysis and Recognition (ICDAR '09), pp. 1383 – 1387, Barcelona, 26-29 July 2009.

[42] R.O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis," New York: John Wiley & Sons, 1973.

[43] L. Devroye, L. Gyorfi, and G. Lugosi, "A Probabilistic Theory of Pattern Recognition," Springer, Apr 1996.

[44] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Stat. Soc., vol. 39, no. 1, pp. 1-38, 1977.

[45] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of Markov process to automatic speech recognition," Bell Syst. Tech. j., vol. 62, no. 4, pp. 1035-1074, Apr. 1983.

[46] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," IEEE Trans. Information Theory, vol. IT-28, no. 5, pp. 729-734, 1982.

[47] B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," IEEE Trans. Information Theory, vol. IT-32, no. 2, pp. 307-309, Mar. 1986.

[48] S. J. Smith, M. O. Bourgoin, K. Sims, and H. L. Voorhees, "Handwritten character classification using nearest neighbor in large databases," IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 16, no. 9, pp. 915–919. Sep 1994.

[49] M. Pechwitz, S. S. Maddouri, V. Maergner, N. Ellouze, and H. Amiri, "IFN/ENIT ─ database of handwritten arabic words," In Proceedings CIFED'02, pages 129–136, 2002.

[50] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "UNIPEN project of on-line data exchange and recognizer benchmarks," In Proceedings of International Conference on Pattern Recognition(ICPR 1994), vol. 2, pp. 29–33, Jerusalem, Israel, 9-13 Oct 1994.

[51] J. J. Hull, "A database for handwritten text recognition research," IEEE Trans. on Pattern Analysis and Machine Recognition, vol. 16, no. 5, pp. 550–554, May 1994.

[52] C. Viard-Gaudin, P. M. Lallican, S. Knerr, and P. Binter, "The IRESTE On-Off (IRONOFF) dual handwritten database," In Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR '99), pp. 455–458, Bangalore, 20-22 Sep 1999.

[53] M. Pechwitz and V. Maergner, "HMM-based approach for handwritten Arabic word recognition using the IFN/ENIT–database," In Proceedings of the Seventh International Conference on Document Analysis and Recognition, pages 890–894, 3-6 Aug 2003.

[54] International Unipen Foundation, "The UNIPEN Project," http://www.unipen.org, 1994.

[55] M. Agrawal, K. Bali, S. Madhvanath, and L. Vuurpijl, "UPX: a new XML representation for annotated datasets of online handwriting data," In Proceedings of the Eighth International Conference on Document Analysis and Recognition, vol. 2, Pages 1161–1165, 29 Aug–1 Sep 2005.

[56] A. S. Bhaskarabhatla, M. P. Kumar, A. Balasubramanian, C. Jawahar, and S. Madhvanath, "Representation and Annotation of Online Handwritten Data," In Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Tokyo, Japan, October 2004.

[57] W3C Multi-modal Interaction Working Group, "Ink Markup Language (InkML)," http://www.w3.org/2002/mmi/ink, 2003.

[58] A. S. Bhaskarabhatla and S. Madhvanath, "An XML Representation for Annotated Handwriting Datasets for Online Handwriting Recognition," In Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal, May 2004.

[59] S. Madhvanath, L. Vuurpijl, K. Bali, M. Agrawal, V. Jagannadan, D.Vijayasenan, D. Willems, and L. Schomaker, "UPX Schema version 0.9.5," Technical report, International Foundation and HP Labs India, October 2006, http://unipen.nici.kun.nl/upx/docs/UPXSchemaVersion0.9.5.doc (Last visited 2007-03-05).

[60] L. Schomaker and L. Vuurpijl, "Forensic writer identification: A benchmark dataset and a comparison of two systems [internal report for the Netherlands Forensic institute]," Technical report, Nijmegen: NICI (2000).

[61] M. Kherallah, N. Tagougui, A. M. Alimi, H. El-Abed, and V. Mârgner, “Online Arabic Handwriting Recognition Competition,” 2011 International Conference on Document Analysis and Recognition (ICDAR), Pages 1454-1458, Beijing, 18-21 Sept. 2011.

[62] H. Boubaker, A. Chaabouni, M. Kherallah, A. M. Alimi, and H. El-Abed, "Fuzzy Segmentation and Graphemes Modeling for Online Arabic Handwriting Recognition," In Proceeding of 2010 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India, 16-18 Nov. 2010.

[63] H. Boubaker, A. El-Baati, M. Kherallah, and A. M. Alimi, "Online Arabic Handwriting Modeling System based on the

Graphemes Segmentation," In Proceeding of 20th International Conference on Pattern Recognition (ICPR 2010), vol. 20, pp. 2061–2064, Istanbul, Turkey, 23-26 August 2010.

[64] W. Guerfali and R. Plamondon, "Normalizing and Restoring Online Handwriting," Pattern Recognition, vol. 26, no. 3, pp. 419–431, 1993.

[65] A. AL-Shatnawiand K. Omar, "Methods of Arabic Language Baseline Detection: The State of Art," IJCSNS International Journal of Computer Science and Network Security, vol.8, no.10, October 2008.

[66] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," IEEE Computer, vol. 29, no. 3, pp. 31-44, Mar 1996.

[67] X. Jiang and A. H. K. S.Wah, "Constructing and training feed-forward neural networks for pattern classification," Pattern Recognition, vol. 36, no. 4, pp. 853–867, April 2002.

[68] T. K. Moon, “The Expectation-Maximization Algorithm in Signal Processing,” IEEE Signal Processing Magazine, v. 13, no. 6, pp. 47–60, Nov 1996.

[69] J. Oh, "An On-Line Handwriting Recognizer with Fisher Matching, Hypotheses Propagation Network and Context Constraint Models," PhD Dissertation in Computer Science, New York University, 2001.