



Biometrics Writer Recognition for Arabic language: Analysis  
and Classification techniques using Subwords Features

---

By

Makki Jasim Radhi Maliki

Department of Applied Computing

University of Buckingham

United Kingdom

A Thesis Submitted for The Degree of Doctor of Philosophy in  
Computer Science to the School of Science in the University of  
Buckingham.

August 2015

Copyright © Makki Maliki and the University of Buckingham 2015

All Rights Reserved

## **Abstract**

Handwritten text in any language is believed to convey a great deal of information about writers' personality and identity. Indeed, handwritten signature has long been accepted as an authentication of the writer's physical stamp on financial and legal deals as well as official/personal documents and works of art. Handwritten documents are frequently used as evidences in forensic tasks. Handwriting skills is learnt and developed from the early schooling stages. Research interest in behavioral biometrics was the main driving force behind the growth in research into Writer Identification (WI) from handwritten text, but recent rise in terrorism associated with extreme religious ideologies spreading primarily, but not exclusively, from the middle-east has led to a surge of interest in WI from handwritten text in Arabic and similar languages.

This thesis is the main outcome of extensive research investigations conducted with the aim of developing an automatic identification of a person from handwritten Arabic text samples. My motivations and interests, as an Iraqi researcher, emanate from my multi-faceted desires to provide scientific support for my people in their fight against terrorism by providing forensic evidences, and as contribute to the ongoing digitization of the Iraqi National archive as well as the wealth of religious and historical archives in Iraq and the middle-east. Good knowledge of the underlying language is invaluable in this project.

Despite the rising interest in this recognition modality worldwide, Arabic writer identification has not been addressed as extensively as Latin writer identification. However, in recent years some new Arabic writer identification approaches have been proposed some of which are reviewed in this thesis. Arabic is a cursive language when handwritten. This means that each and every writer in this language develops some unique features that could demonstrate writer's habits and style. These habits and styles are considered as unique WI features and determining factors.

Existing dominating approaches to WI are based on recognizing handwriting habits/styles are embedded in certain parts/components of the written texts. Although the appearance of these components within long text contain rich information and clues to writer identity, the most common approaches to WI in Arabic in the literature are based on features extracted from paragraph(s), line(s), word(s), character(s), and/or a part of a character. Generally, Arabic words are made up of one or more subwords at the end of each; there is a connected stroke with a certain style of which seem to be most representative of writers habits. Another feature of Arabic writing is to do with diacritics that are added to

written words/subwords, to add meaning and pronunciation. Subwords are more frequent in written Arabic text and appear as part of several different words or as full individual words. Thus, we propose a new innovative approach based on a seemingly plausible hypothesis that subwords based WI yields significant increase in accuracy over existing approaches. The thesis most significant contributions can be summarized as follows:

- Developed a high performing segmentation of scanned text images, that combines threshold based binarisation, morphological operation and active shape model.
- Defined digital measures and formed a 15-dimensional feature vectors representations of subwords that implicitly cover its diacritics and strokes. A pilot study that incrementally added features according to writer discriminating power. This reduced subwords feature vector dimension to 8, two of which were modelled as time series.
- For the dependent 8-dimensional WI scheme, we identify the best performing set of subwords (best 22 subwords out of 49 then followed by best 11 out of these 22 subwords).
- We established the validity of our hypothesis for different versions of subwords based WI schemes by providing empirical evidence when testing on a number of existing text dependent and in text-dependent databases plus a simulated text-in text-dependent DB. The text-dependent scenario results exhibited possible present of the Doddington Zoo phenomena.
- The final optimal subword based WI scheme, not only removes the need to include diacritics as part of the subword but also demonstrating that including diacritics within subwords impairs the WI discriminating power of subwords. This should not be taken to discredit research that are based on diacritics based WI. Also in this subword body (without diacritics) base WI scheme, resulted in eliminating the presence of Doddington Zoo effect.
- Finally, a significant but un-intended consequence of using subwords for WI is that there is no difference between a text-independent scenario and text-dependent one. In fact, we shall demonstrate that the text-dependent database of the 27-words can be used to simulate the testing of the scheme for an in text-dependent database without the need to record such a DB.

Finally, we discussed ways of optimising the performance of our last scheme by considering possible ways of complementing our scheme using the addition of various image texture analysis features to be extracted from subwords, lines, paragraphs or entire

file of the scabbed image. These included LBP and Gabor Filter. We also suggested the possible addition of few more features.

In the loving memory of my beloved  
Father and Mother

## **Acknowledgment**

I would like to express my special appreciation and thanks to my supervisors Professor Dr. Sabah Jassim and Dr. Naseer Al-Jawad, you have been marvelous advisers for me. I would very much like to thank you for encouraging me and for helping me reach my potentials in my research and for allowing me to grow as a research scientist. Your advice, on both research and on my everyday livelihood when at Buckingham, have been priceless.

I would also like to thank the entire academic staff in the applied computing department for their tremendous help and effort.

I would especially like to thank my sponsors The Iraqi Higher Education Ministry, The University of Baghdad, and the Iraqi cultural attaché in London.

I would also like to thank the Middle East Culture Group in Milton Keynes especially the head of the community Mrs. Ayser Al-Jawad for their help and support.

I would also like to express my utter appreciation to my friend Wamedh Kereem for his help and support.

Special thanks go to my family. Words cannot express how grateful I am to my mother, for all of the sacrifices that she made on my behalf. Your prayers for me were what sustained me so far. I would also like to thank all of my friends who supported me in the writing, collecting Arabic database text, and motivating me to strive towards my goals.

At the end I strongly express my appreciation to my beloved family, Suham, my beloved sons (Ahmed and Baqer), and my lovely daughters (Lien and Sara) who spent sleepless nights with me and was always my biggest motivators to keep on going during difficult times.

## Abbreviations

<b>Abbreviation</b>	<b>Phrase</b>
2D	Two Dimension
ANN	Artificial Neural Networks
Ber	Bernoulli
CS	Connected Stroke
ComS	Compressive Sensing
DD	Data Base
DTW	Dynamic Time Warping
EHAA	Enhancing Histogram Analyses Approach
FLVQ	fuzzy learning vector quantization
GSC	Gradient, Structural and Concavity
GUI	Graphic User Interface
Gus	Gaussian
HPP	Horizontal Projection Peak
KNN	K nearest neighbours
LBP	Local Binary Pattern
LCC	Labelling Connected Components
MLP	Multi-Layer Perceptions
OCR	Optical character recognition
PCA	Principal Component Analysis
RIP	Restricted Isometry Property
SC	Shape Curvature
SCON	Shape Context contour shapes
SFT	Single Feature Test
SOM	self-organizing feature map
SVM	Support Vector Machine classifier
SIFT	Scale Invariant Feature Transform
SD	SIFT descriptors
SO	SIFT corresponding scales and orientations
WI	Writer identification
WMR	Word Model Recognizer

# Table of Contents

Abstract .....	i
Acknowledgment .....	v
Abbreviations .....	vi
Abbreviation.....	vi
Table of Contents .....	vii
List of Figures.....	x
List of Tables.....	xiv
<b>Chapter 1 : Introduction .....</b>	<b>1</b>
1.1 Introduction to Biometric Systems.....	1
1.2 Categorisation of WI Systems.....	4
1.2.1 Writer identification vs. Handwriting recognition.....	4
1.2.2 Writer identification vs. Writer verification .....	4
1.2.3 Offline vs. Online.....	5
1.2.4 Text-dependent vs. Text-independents .....	6
1.3 Arabic Scripts Structure .....	6
1.4 Thesis Motivations and Objectives .....	9
1.5 Contributions.....	11
1.6 Thesis Outline.....	12
1.7 Thesis Publications.....	12
<b>Chapter 2 : Literature Review .....</b>	<b>14</b>
2.1 Writer Identification – Background and Related Issue .....	14
2.2 A Brief Survey of Recent Research in WI from Handwritten Text .....	16
Work Based on Characters.....	16
Diacritics-based WI.....	19
Words based WI.....	20
Line based WI research.....	22



Work based on Paragraph (page or document) .....	23
Work Used a Combination of text parts.....	24
Our Approach.....	25
2.3 Databases we used.....	26
2.4 Latest Work .....	28
2.5 Summary of the Chapter.....	30
<b>Chapter 3 : Pre-processing and segmentation of Arabic Texts .....</b>	<b>31</b>
3.1 Arabic Language Text Analysis .....	31
3.1.1 Subwords Characteristic .....	33
3.1.2 Subwords with Stroke That Reflects Writer Habits.....	35
3.2 Automating Writer Identification .....	36
3.3 Common Challenges .....	36
Text noise – How does it appear? .....	38
3.4 The Developed Solution.....	39
3.4.2 Text Orientation (Skew) Problem .....	44
3.5 Text Segmentation.....	47
3.5.1 Line Segmentation .....	48
3.5.2 Segmenting Lines into Words, Subwords and Diacritics .....	50
3.5.3 Enhancing Segmentation Algorithm.....	52
3.5.4 The LCC Based Text Segmentation.....	56
3.6 Summary of the Chapter.....	60
3.7 Next Step .....	60
<b>Chapter 4 : Subword based Arabic Handwriting Analysis for WI.....</b>	<b>61</b>
4.1 Introduction .....	62
4.2 Components of WI Scheme for Arabic Language. ....	65
4.2.1 Handwritten Feature Extraction .....	66
4.3 WI Scheme(s) for Arabic Handwritten Text .....	73

4.3.1	Feature Selection- Experimental Design.....	73
4.3.2	Single Feature Test (SFT) .....	75
4.3.3	The Incremental WI Scheme.....	80
4.3.4	Writer Identification Rate (WIR).....	83
4.4	The Performance of Non-projection Features – Revisited.....	91
4.5	Summary of the Chapter.....	94
<b>Chapter 5 : WI based on Subwords without their diacritics .....</b>		<b>96</b>
5.1	Arabic Writer Identification based on Body of Subwords Alone .....	96
5.2	Diacritics Removal from Subword Boxes.....	98
5.3	The Subword Body based WI Scheme.....	100
5.4	Summary and Conclusions.....	107
<b>Chapter 6 : WI from Text-Independent Handwritten Arabic Texts .....</b>		<b>108</b>
6.1	WI from Text-Independent Data .....	108
6.1.1	The Adjusted Subword based WI Scheme for Independent Texts .....	109
6.1.2	The in-House in text-dependent database .....	109
6.1.3	WI Based on Subword with and without Their Diacritics .....	110
6.2	Simulating Subset Testing (Text-Independent from a Text-dependent database) 115	
6.3	How to Improve Writer Identification for text-Independent Scenario.....	118
6.4	Summary of the Chapter.....	124
<b>Chapter 7 : Conclusion .....</b>		<b>126</b>
<b>References .....</b>		<b>131</b>
<b>Appendix.....</b>		<b>140</b>

## List of Figures

Figure 1: Biometrics types .....	2
Figure 2: Block diagram for identification system .....	3
Figure 3: Individuality of handwriting system: a. Identification model, and b. Verification model.....	5
Figure 4: Diacritics. A: Compulsory Diacritical: B: Doubled Consonants Diacritical Marks C: Short Vowels Diacritical .....	7
Figure 5: Arabic Sentence divided into words subwords and diacritics.....	7
Figure 6: a. words and b. subwords .....	8
Figure 7: Ascenders and descenders are circled; horizontal lines are shown for reference (Amin, 1998).....	8
Figure 8: The best known Arabic calligraphic fonts/styles (translation of the sentence is: "Calligraphy is the tongue of hand") (Zoghbi, 2007, Accessed 13 April 2013).....	9
Figure 9: Typical OCR system.....	15
Figure 10: An example of IFN/ENIT database (Pechwitz, et al., 2002).....	26
Figure 11: An example of Al-Ma'adeed database (Al-Ma'adeed, et al., 2008).....	27
Figure 12: In-house database. a: text 1, b:text2 .....	27
Figure 13: In-house database, another example, a: text 1, b: text 2.....	28
Figure 14: Printed Arabic sentence and its structure .....	32
Figure 15: Examples of Connecting Stroke (CS) of lowercase Latin strokes (Huber & Headrick, 1999).....	35
Figure 16: Examples of Arabic Connecting Stroke (CS).....	36
Figure 17: Examples of subwords overlap. a: Arabic Printed text, b: Handwritten version with marked overlapping. ....	38
Figure 18: Binary image based on a Global threshold, a. Printed text image (for comparison), b. original handwriting text image, c. Binary image based on Global method .....	40
Figure 19: Binary image based on Local method .....	41
Figure 20: Image filtering by applying median filter, the input image was Figure 19 ...	42

Figure 21: Image filtering by applying Gaussian_filter_5_1, the input image was Figure 19.....	42
Figure 22: image cleaning by applying Morphological clean, the input image was Figure 19.....	43
Figure 23: image cleaning by applying Morphological Dilation, the input image was..	43
Figure 24: image cleaning by applying Morphological Erosion, the input image.....	44
Figure 25: Image cleaning by applying Morphological clean, the input image was Figure 18c (Binary image based on Global method).....	44
Figure 26: orientation Arabic paragraph .....	45
Figure 27: a. Orientation of an Arabic paragraph, b. horizontal projection.....	45
Figure 28: rotated (Figure 26) text, a. rotated text, b. horizontal histogram of (a).....	47
Figure 29: Line segmentation .....	48
Figure 30: line segmentation result .....	49
Figure 31: lost slope features demonstrates, a text line (taken from a text paragraph) before and after segmentation .....	49
Figure 32: re-rotate lines (Feature's recovery) .....	50
Figure 33: identify words and subwords gaps .....	51
Figure 34: words and subwords segmentation.....	51
Figure 35: Sub -words vertical segmentation (Berkani & Hammami, 2002) .....	51
Figure 36: a. Arabic printed sentence, b. handwritten text, and c. segmented word and subwords. ....	52
Figure 37: Identifying baseline of each and every segment.....	52
Figure 38: Re-positioning every segment on estimated baseline.....	53
Figure 39: Another example of re-positioning every segment on estimated baseline. a. original text line, b. re-positioning segments .....	53
Figure 40: segmentation results before applying further enhancement .....	54
Figure 41: segmentation after enhancement based on vertical projection threshold (=20) .....	54

Figure 42: segmentation after enhancement based on vertical projection threshold (=25) .....	54
Figure 43: Diacritics segmentation, (a) locating of the start points, (b) after clearing the text, (c) final diacritics segmentation. (Lutf, et al., 2010).....	56
Figure 44: connectivity diagrams (4 and 8-neighbor).....	57
Figure 45: Segmentation based on LCC. a. first word, b. second word .....	57
Figure 46: Text segmentation based on LCC algorithm .....	58
Figure 47: Proposed Segmentation system based on LCC .....	59
Figure 48: WI scheme- Block Diagram .....	64
Figure 49: Pre-processing, a: Original image, b: Binarised and cleaned image .....	65
Figure 50: Segmentation, a. LCCA image, b. re-attached subwords with their diacritics .....	65
Figure 51: Illustration of the Linear Regression Model.....	67
Figure 52: A. Subword slope, height, width, and area. ....	67
Figure 53: Subword distance from upper/lower phrase baseline .....	67
Figure 54: Invariant moment results for a subword.....	69
Figure 55: Subword Projections, a: Subword, b: Horizontal projection, c: Vertical projection .....	70
Figure 56: Illustration of DTW (Mathieu, 2009) .....	72
Figure 57: Warping Function (Mathieu, 2009).....	72
Figure 58: A. projections for (أ), B. projections for (بسم), a. subword, b. horizontal and c. vertical projection. ....	78
Figure 59: Incremental Features (I.F.) list of Experiments.....	80
Figure 60: Incremental Features Performance Chart .....	81
Figure 61: Structure of the testing system .....	84
Figure 62: WI system flow diagram .....	85
Figure 63: Variation in the position and Shape of diacritics of a subword written by the same writer.....	97

Figure 64: Projections of subword in of Figure 63, before and after diacritic removing	97
Figure 65: results that illustrate in table above (Table 5.4)	102
Figure 66: some of the possible problems associated with removing the diacritics from the subword هي	103
Figure 67: Faulty segmentation of the subword بخير for 2 different writers	104
Figure 68: image zigzag (Pantech, 2014)	119
Figure 69: example LBP algorithm (Ojala, et al., 2002)	119
Figure 70: LBP results of Left: writer 1 subword 1 and Right writer 2 subword 1	120
Figure 71: LBP results of writer 1: subword 2	120
Figure 72: Apply Gabor filter for text1 writer1	121
Figure 73: Apply Gabor filter for text2 writer1	122
Figure 74: Apply Gabor filter for text1 writer2	122
Figure 75: Apply Gabor filter for text2 writer2	123

## List of Tables

Table 1.1: The Arabic alphabet.....	6
Table 1.2: Variation in Handwritten letters (Amin, 1998).....	8
Table 2.1: (Aboul-Ela, et al., 2015) system accuracy .....	29
Table 3.1: Arabic vs. Other languages' subwords (Good morning phrase) .....	33
Table 3.2: Subwords repeated in dissimilar words for (subwords: في and قي) first example .....	33
Table 3.3: subwords (م) is repeated in a number of different words.....	34
Table 3.4: subwords (يم) is repeated in a number of different words.....	34
Table 3.5: subwords (سي) is repeated in a number of different words.....	34
Table 3.6: Subwords repeated in same word .....	34
Table 3.7: Example of High Subword repeated in dissimilar words if diacritics are not included.....	35
Table 3.8: comparison analysis between result of segmentation based on EHA and LCC .....	59
Table 4.1: Works that are based on word characteristic using text-dependent databases in different languages .....	62
Table 4.2: Repetition of the 49 subwords occurring the 27 words document .....	75
Table 4.3: Writer identification accuracy for each individual feature .....	77
Table 4.4: Subwords with errors way above the overall errors for each feature group plus the number of error places .....	79
Table 4.5: Feature weight calculated based on their accuracy.....	80
Table 4.6: Best WI discriminating Subwords with Feature vector ( $f_{14}+f_{15}+f_4+f_2+f_3+f_7+f_1+f_6$ ) .....	83
Table 4.7: Subwords success hits and accuracy in descending order of accuracy.....	86
Table 4.8: Success samples of all writers (out of 15 in each sample) in high to low order .....	87
Table 4.9: Success samples of all writers using 11 subwords (out of 15 in each sample) in high to low order.....	88

Table 4.10: Performance Summary of our Subwords-based WI schemes.....	89
Table 4.11: subwords success hit in order (from high to low) for all testing writers .....	90
Table 4.12: Success samples of all writers (out of 15 in each sample) in high to low order .....	91
Table 4.13: ComS based WI schemes (using Gaussian and Bernoulli methods) .....	94
Table 5.1: Example of higher number of subwords repeated in dissimilar words when diacritics are omitted.....	98
Table 5.2: The selected 14 subwords without their diacritics.....	99
Table 5.3: Performance of Segmentation algorithm .....	100
Table 5.4: subwords success hit in order (from high to low) using all writers.....	102
Table 5.5: subword هي with different writers before and after segmentation as presented in Figure 66 .....	103
Table 5.6: Success samples of all writers (out of 15 in each sample) in high to low order .....	105
Table 5.7: system accuracy in summary .....	106
Table 6.1: Subwords matching (within text) and between both texts.....	110
Table 6.2: Identification rate (%) for individual subword (with and without their diacritics).....	112
Table 6.3: Writer identification based on majority of subwords in text .....	113
Table 6.4: Writer identification based on (Average of samples algorithm).....	114
Table 6.5: Subwords weighting map.....	114
Table 6.6: Writer identification based on (Weighted subword algorithm) .....	115
Table 6.7: Simulated WI using 5 random subwords.....	116
Table 6.8: Simulated WI using 9 random subwords.....	117
Table 6.9: Examples of Arabic handwriting strokes for different writers. ....	124
Table 1: The Arabic alphabet.....	140



# **Chapter 1 : Introduction**

This thesis is concerned with Writer identification (WI) from the handwritten text in Arabic. Generally, WI from the handwritten text, in any language, is a behavioral biometrics due to the fact that personal style of writing is a habit that is learnt and refined from an early age. The identification of a person from his/her handwriting samples remains a useful biometric technique, with a variety of applications covering digitizing religious and historical archives, forensics, crime and terrorism fighting.

Technically, the automatic recognition of a person from his/her handwriting can be dealt with in the similar way that any biometric-based recognition. It is a pattern recognition problem that involve the extraction of digital feature vector representation, the availability of sufficient number of samples of such vectors from a number of users, the existence of a measure/distance defined between these feature vectors that could naturally reflect similarity between the persons from whom the samples are obtained, and an appropriate classification scheme. This problem can benefit from existing research into WI from handwriting in other languages such as English, but the structure and characteristics of the Arabic language will have to be taken into account when we attempt to tackle this specific challenge. There are many factors that influence the performance of any such schemes including differences in national and educational backgrounds of Arabic text writers

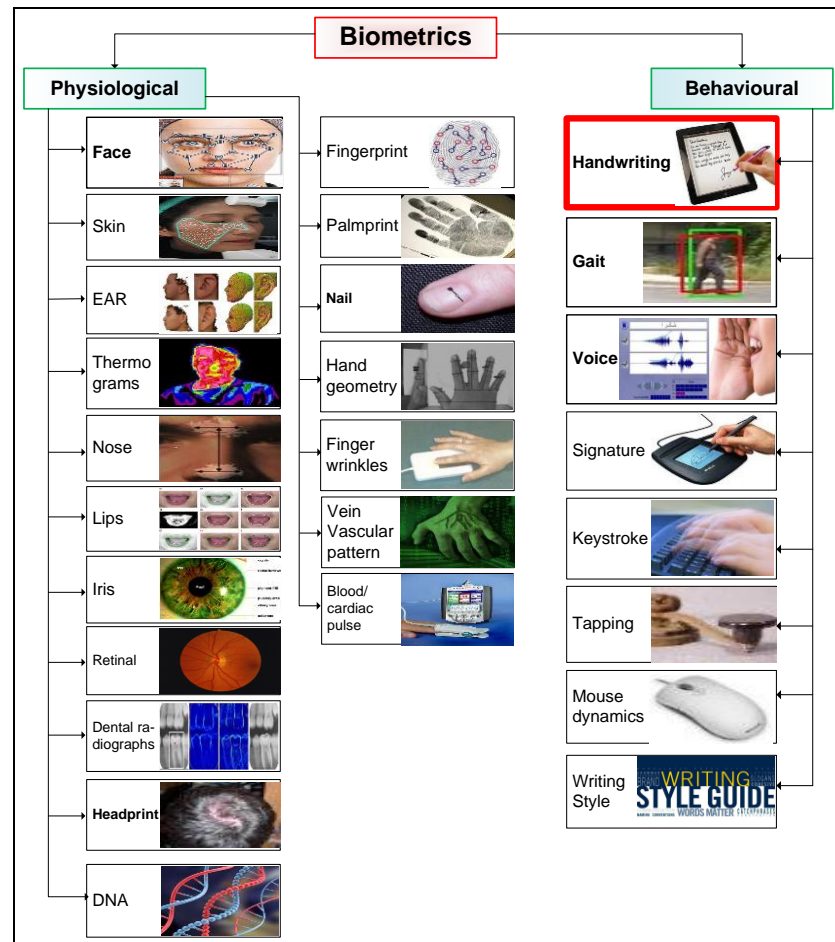
In this chapter, we describe the background materials and challenges in writer identification, briefly highlighting the approaches adopted in our investigations and the motivation behind them. In section 1.1, we give a brief description of biometric systems including WI, and in section 1.2 a categorization of handwritten text analysis is presented. In section 1.3, the structure and styles of Arabic scripts will be described. Section 1.4 is concerned with the motivations and objectives of this thesis while Section 1.5 outlines the contributions of this work while Section 1.6 outlines the organization of the rest of the thesis.

## **1.1 Introduction to Biometric Systems**

Biometrics is the automated recognition of individuals based on their physical (Physiological) and behavioral characteristics. Physical characteristics include human attributes like: face, skin, ear, thermograms, nose, lips, iris, retinal, dental radiogram,

head print, DNA, fingerprint, palm print, nail, hand geometry, finger wrinkles, vein vascular pattern and blood cardiac pulse.

Behavioral characteristics include handwriting (optical character recognition and writer identification), signature, gait, voice, keystroke, tapping, mouse dynamics, and writing style. See Figure 1

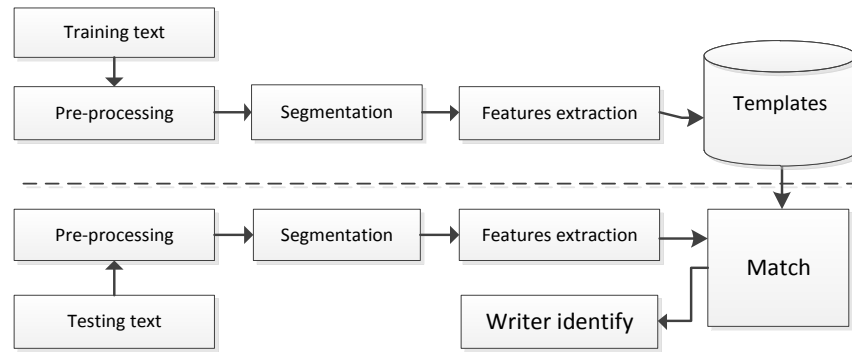


**Figure 1:** Biometrics types

From the physical body or the individual behavior properties, features are extracted to build biometrics templates as a digital representation of the chosen trait for the individual. At the core of any biometric system is the feature extraction procedure that creates the chosen trait template and store it in a database to be used for matching.

Biometric identification is performed by taking a new (fresh) sample given for investigation from an unknown person or a claimant, and comparing it with the templates of previously entered/enrolled persons in a biometric system’s database. Figure 2 is a block diagram for identification and enrolment. Matching is based on another essential component of the system, namely a similarity/distance function that measures the level of similarity between the fresh feature vector and each of those in the database. Exact matching is highly unlikely, and, in fact, should trigger an alarm when it happens.

Therefore, the tolerance of variation between the new and database feature vectors is controlled by a threshold that is normally determined through a training process. The person may or may not be already in the system, the outcome of identification should either confirm the claimed identity, identify the person in the system nearest to person of the fresh biometric sample or return “unknown”.



**Figure 2:** Block diagram for identification system

Physiological biometrics, especially iris, fingerprint, and DNA, are more accurate than Behavioral biometrics due to the low contrast and high complexity of the biometric templates. This is extensively used and performed in identification systems all over the world.

On the other hand, behavioral biometrics is less invasive as the performance achieved is less impressive due to the large contrast between the various behavior-derived biometric templates.

Even so, the identification of a person from his/her handwriting samples remains a useful biometric technique, mainly due to its applicability in the forensic field. Writer identification (WI) is kind of behavioral biometrics due to the fact that personal style of writing is a habit that is learnt and refined from an early age.

Our WI system is based on Arabic handwriting texts. Arabic language has a wide usage spectrum; statistically it is spoken by 347 million people (Lewis, 2009). In addition, over 1.2 billion Muslims all over the world use Arabic language daily when citing the Quran and in their prayers.

Some other languages use Arabic letters in their script (populations of about 700 million) or use the same letter shapes with minor differences. Examples of such languages include Hausa, Kashmiri, Kazak, Kurdish, Kyrghyz, Malay, Morisco, Pashto, Persian/Farsi,

Punjabi, Sindhi, Tatar, Ottoman Turkish, Uyghur, and Urdu. Moreover, Arabic, as one of the five languages widely spoken in the world (Chinese, English, Spanish, and Hindi/Urdu).

## **1.2 Categorisation of WI Systems**

Handwritten text analysis for recognition tasks depends on the purpose of the analysis, the way it is conducted and type of text. In this section, we describe the different categories of WI systems.

### **1.2.1 Writer identification vs. Handwriting recognition**

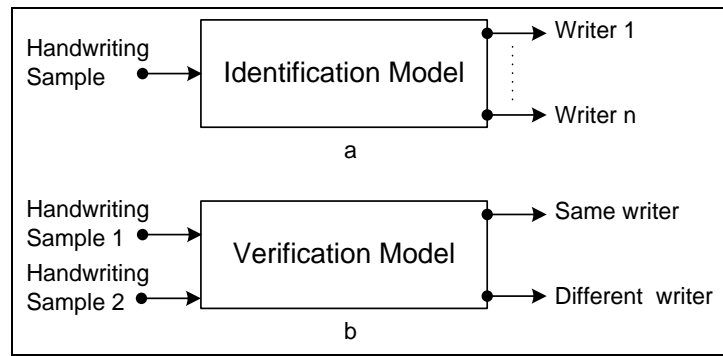
The aim of handwriting recognition that is also called Optical Character Recognition (OCR) is to classify optical patterns (often contained in a digital image) and convert them to alphanumerical or other characters in a number of languages, by finding out the variations between different handwritings for the purpose of correctly identifying the shapes of characters or words. In other words, the system will digitise the image of a handwriting text.

Writer handwriting analysis, however, is a special process designed to link text to a specific writer. It is assumed that no two people write alike. Some similarities may exist, but when inspected closely, handwriting varies from a person to a person (Kohn, et al., 2011). Each person's writing (habit and style) is expected to become unique eventually to that person and is the result of unconscious, automatic actions and interaction between the person's brain, eyes and hand.

WI is totally different from OCR, where WI is concerned with identifying the actual writer of a text while OCR is concerned with the recognition of the actual written text including letters, characters, numbers, words, and so on. It is not designed to identify the individual who wrote that text.

### **1.2.2 Writer identification vs. Writer verification**

A writer identification system executes a one-to-many test from a large database with handwriting samples of well-known authorships and it returns a possible list of candidates, while in Writer verification the process involves a one-to-one comparison with clarification as to whether or not the two samples are written by the same person as shown in Figure 3 below.



**Figure 3:** Individuality of handwriting system: a. Identification model, and b. Verification model.

In writer identification, the testing process is based on capturing the features of handwriting individuality turning them into templates in a database and put in order based on the distance between the actual templates and the sample.

On the other hand, in writer verification, the distance between two given samples is examined. If the distance is equal or smaller than a pre-defined threshold, then the samples are written by the same person. Otherwise, the samples are considered to be written by a different writer.

### 1.2.3 Offline vs. Online

Based on the input method of writing, writer identification has been classified into on-line and off-line. It is “on-line” if simulation of the tracking of the pen point is available. The writer will be asked to write on the screen using a specific instrument like a 'stylus' rather than using traditional pen and paper. On the other hand, it is “off-line” if it is applied to earlier written text, for instance, a traditionally written text image scanned by a scanner.

On-line problems are usually lesser than off-line problems as more information is available about the writing style of a person, such as speed, angle or pressure, which is not available in the off-line technique (Lorigo & Govindaraju, 2006), (Schlapbach, et al., 2008), (Schomaker, 2007).

**This thesis is restricted to off-line techniques only** because it is concerned mainly with samples that are important for forensic and other purposes and need to be matched and related to a specific writer among other writers.

### 1.2.4 Text-dependent vs. Text-independents

Based on text contents, there are two methods in writer identification approaches: text dependent and text-independent.

Text-dependent method only matches the same text (mostly letters or words) and accordingly requires the writer to write the same text more than one time. Such texts may need further pre-processing and segmentation. (Bulacu, 2007), (Sreeraj.M & Idicula, 2011).

Text-independent method is all about analysing a text made of a few written lines as a minimal amount of handwriting necessary to obtain a sample with sufficient attributes and features to identify the writer.

In our opinion text-dependent method, while used widely by most researchers, falls short of satisfying the requirement of identification a writer of the anonymous text.

**Our approach uses both text-dependent and text-independent methods** in processing and segmenting a text.

Most researchers use text-dependent DB only in their work. While we used text-dependent DB mainly to extract the best group of features and then the best group of subwords to be used when we examine texts entered in in text-dependent DB. Other reasons for using text-dependent DB is to compare our hypothesis against other tested systems that use entire words in their work.

### 1.3 Arabic Scripts Structure

Arabic is a cursive language written from right to left. The Arabic alphabet consists of 36 letters, of which 28 are primary, and 8 are modified. Each letter has between two to four shapes when written in a word: isolated, initial, medial, and final as shown in Table 1.1 (for full table see the Appendix)

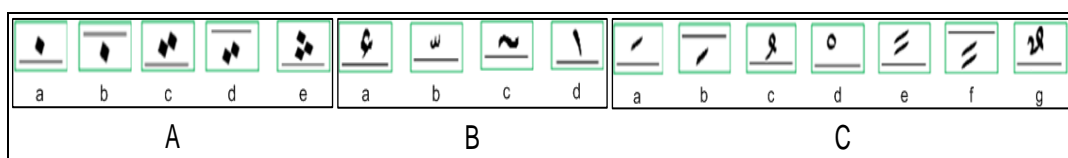
Table 1.1: The Arabic alphabet

No	Name of Letter in Arabic	Sound	Example in English	Isolated	Initial	Medial	Final
1.	Alif	Ā	'a' as in 'father'	ا	ا	ا	ا، اء
2.	Baa	B	'b' as in 'bed'	ب	بـ	بـ	ب، بب
3.	<b>for full table see the Appendix</b>						

Each word in Arabic language is constructed using different letter styles. Some letters can be connected to their neighbours on one or both sides like (بر, ثو, صبا) while other letters may be completely disconnected from their neighbours like (ورد).

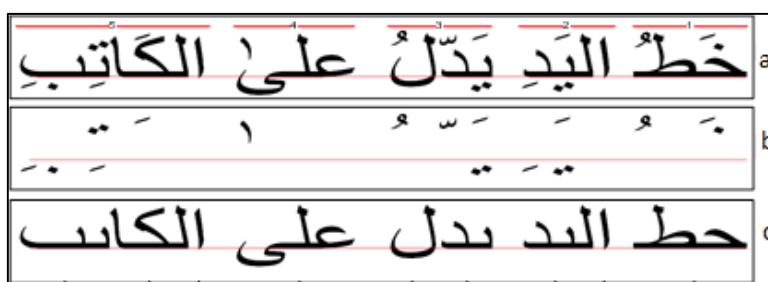
Most letters in Arabic words contain diacritics. They are used to reduce confusion between some similar letters in shapes like (ث, ت, ي, ن, ب). Also, diacritics are used to help pronunciation.

There are 16 types of diacritics in total. These diacritics can be classified as compulsory and optional. An example of compulsory diacritics is dots which are used to differentiate between letters, (one, two or three dots), as shown in Figure 4A. Optional diacritics are mainly short vowels, which are used to emphasize consonants; examples are shown in Figure 4C. Other optional diacritics indicate the pronunciation of doubled consonants or apply different sounds, as shown in Figure 4B.



**Figure 4:** Diacritics. A: Compulsory Diacritical: B: Doubled Consonants Diacritical Marks C: Short Vowels Diacritical

A typical printed Arabic sentence is shown in Figure 5 where (a) indicates the words highlighted by the red lines on top. Diacritics only are shown in (b) and the red line in the middle indicates the baseline, and (c) shows the body of the words without diacritics.



**Figure 5:** Arabic Sentence divided into words subwords and diacritics.

Words (Figure 6 ) in these languages are a combination of subwords that consist of one or more letters. These subwords are separated by small gaps.

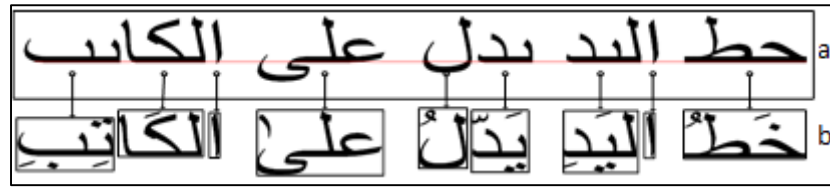


Figure 6: a. words and b. subwords

Some letters are “descenders” where their letters extend below the estimated baseline, and some are “ascenders” where their letters extend above baseline. See Figure 7.

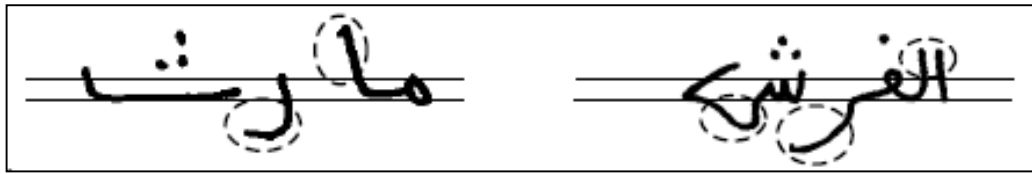


Figure 7: Ascenders and descenders are circled; horizontal lines are shown for reference (Amin, 1998)

**There are no upper or lower cases in Arabic scripts, but only one case.**

Unlike printed words there are many challenges in handwritten texts, we will list these and solve them in a later part of this study, however, some of the challenges that are regarded as “a problem” for an OCR system can be very useful for the process of WI as it will be considered as a distinctive feature. This is due to the enormous variation between writers according to their handwriting habits and styles. Examples of these are the shape of a handwritten diacritic and the specific way of ending a word which we call strokes. See Table 1.2

Table 1.2: Variation in Handwritten letters (Amin, 1998)

Printed char.	Handwritten Char.				Remarks
ا					Vertical line may be missing
ث					Dot pattern varies
ش					Dots and curve shape vary
ي					Curves' angles and letter sizes vary



Arabic calligraphic fonts and styles were developed over time in various Arabic countries, with different writing techniques and writing tools. The best known Arabic calligraphic fonts are shown in Figure 8.

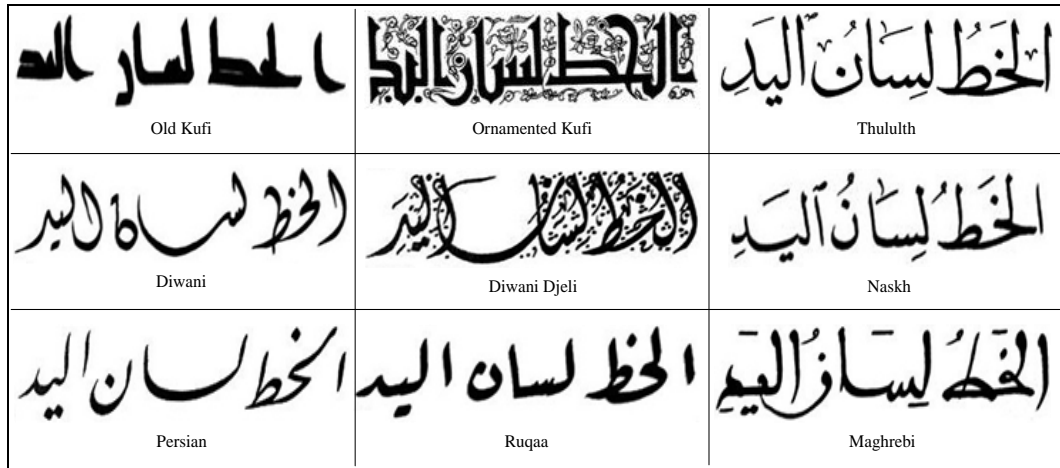


Figure 8: The best known Arabic calligraphic fonts/styles (translation of the sentence is: "Calligraphy is the tongue of hand") (Zoghbi, 2007, Accessed 13 April 2013)

Most of the Arabic scripts available are based on the Naskh or the Thuluth font Style. The other styles like the Kufi, Diwani and Maghrébi are mostly used in historical documents or established in arts or to exhibit typefaces (Zoghbi, 2007, Accessed 13 April 2013).

Common handwriting is based on Naskh and Ruqaa fonts. Therefore, we will avoid those which are used for other reasons because they lack writer's habits and generally have similar features. This fact makes the identification process very difficult. Consequently, our system is based on normal handwriting databases, which are written by non-professional calligraphers.

#### 1.4 Thesis Motivations and Objectives

Writer identification received renewed importance in the last few years, for various reasons:

1. It is increasingly used for forensic evidence and authenticating tool by courts all over.
2. The tremendous increase in crime rate and terrorist activities require vigilant counter activities by the authorities with the specific use of WI. WI from Arabic

handwritten text, after recent terrorist atrocities initiated by Middle Eastern groups, is becoming of great interest to intelligence agencies.

3. It is also a useful tool for historical research including the digitization of old national and religious archives and attributing old text to certain authors.

Items 2 and 3, in the above list, were the main, but by no mean the only, motivation for the research project of this thesis.

Identification of a writer of Arabic text can even be achieved from a small text regardless of any logical combination and sentence construction. This is the principal aim of this thesis. The main objectives of the thesis are to develop and test the performance of an automatic digital scheme of writer identification from the handwritten Arabic text. The system is aimed at capturing the habit/style of the writer, acquired over a period of training and education. We would follow the traditional WI systems, used for different languages, as a pattern and biometric recognition task.

Over the years, a number of important systems have been created based on entire word's features and in many languages. However, such systems have faced many challenges, such as the limitation of the reoccurrence of the same words in a single text, especially if the sample was very small. For example, the appearance of small gaps within a word, especially in cursive text, may lead to confusing features that could be misinterpreted as a separation between two words. This in turn will lead to faulty WI.

The main hypothesis of this thesis is that subwords are an essential reflection of an Arabic writer's habits that could be exploited for WI, thus considering solving the problems that come with the most handwritten text. Questions that arise in relation to the use of subwords and to be dealt with in this thesis include:

1. Is there a specific “relatively small” list of subwords that have more influence WI?
2. When using subwords for WI, should we include or exclude their diacritics?

The thesis objectives include answering these questions. In the coming chapters of this thesis concrete evidence that using our methodology of utilizing subwords, and in particular naked subword bodies (i.e. without diacritics), will produce higher rate of accuracy in the process of Arabic WI.

## 1.5 Contributions

Throughout the research work done for this thesis, we dealt with a number of challenging problems by developing some novel solutions that helped in achieving our thesis objectives. The implementation of the corresponding procedures and tools have led to improving the performance of our proposed WI system some of which are applicable to general handwriting analysis tasks other than WI. Here we list, the main novel contributions categorized as follows:

- **Resolving the problems of overlapping and orientation.** Many writers have developed habits in writing where subwords overlap horizontally or vertically and therefore automatic separation of such subwords become a challenge. On the other hand, most writers have difficulties in keeping the orientation of their text aligned along near-horizontal straight lines. We have developed two complementary solutions that helped enhancing the Pre-Processing and segmentation of handwritten text:
  - EHAA (Enhancing Histogram Analyses Approach) which simply such enhancement process is designed to solve the overlapping problem by aligning text image and then placing each segment on its original estimated baseline. This process is explained in details in 3.5.3 later on.
  - The Labelling Connected Components (LCC) segmentation strategy which is based on using Active Contour model segmentation to resolve the problems of overlapping and orientation without losing any text's attributes.
- Another task which resulted from these experiments is that we have successfully identified a very useful group of commonly used subword that we have incorporated into our system and which proved to be the best group of subwords to identify a writer even in the presence of a small sample of text.
- WI based on Subword features; after that we experimented using the body of the subword on its own stripped of its diacritics. These experiments were very fruitful in terms of WI accuracy rates and hence, we came to realise that basing our proposed system on the plain body of the subword is the best way forward. All of these efforts and experiments are dealt with in details later on in this thesis.
- For the purpose of enhancing the performance of, and the benefits from using, our proposed systems we have introduced and investigated a new and novel concept/algorithm which has significant advantages. :

- **Compressive sensing (ComS) for feature reduction.** We tested the applicability of the new emerging paradigm of ComS in order to obtain the smallest number of meta-features required for WI.

## 1.6 Thesis Outline

The outlines of this thesis are:

- **Chapter 2:** Literature Review: This chapter listed and described all historical efforts and systems in WI in a number of languages.
- **Chapter 3:** Pre-processing and segmentation of Arabic texts: here we have prepared the way for our experimentation where we solved a number of challenges that we predicted to encounter in our experiments like de-noising, binarisation, line segmentation, subword segmentation and especially the huge problem of text overlapping.
- **Chapter 4:** Subword based Arabic Handwriting Analysis for WI; This is a very important chapter in which we have experimented in finding WI using text-dependent DB from which we have extracted the best group of features and the best group of subwords and then applied these on complete subwords (subword included their diacritics)
- **Chapter 5:** WI based on Subwords without their diacritics. We carried out further experiments aiming to achieve the best overall results. We experimented with the body of the subwords on their own where we achieved the most accurate results in WI.
- **Chapter 6:** Investigating the suitability of subwords based WI of Arabic text in the in text-dependent DB scenario whereby one attempts to identify the writer of a given text document/paragraph which is different from stored template text files. Using in text-dependent DB is regarded a challenging system based on the word/subword patterns.
- **Chapter 7:** Conclusion and Proposed Future Work

## 1.7 Thesis Publications

- Maliki, Makki, Sabah Jassim, Naseer Al-Jawad, and Harin Sellahewa. ‘Arabic Handwritten: Pre-Processing and segmentation’, [Conference] // SPIE Defense, Security, and Sensing. - 2012. - pp. 84060D--84060D.

- Maliki, Makki, Naseer Al-Jawad, and Sabah A. Jassim. 'Arabic writer identification based on diacritic's features', [Conference] // SPIE Defense, Security, and Sensing. - 2012. - pp. 84060Y--84060Y.
- Maliki, Makki, Naseer Al-Jawad, and Sabah A. Jassim, 'Sub-word based Arabic Handwriting Analysis for Writer Identification, [Conference] // SPIE Defense, Security, and Sensing. - 2013. - pp. 87550M--87550M.

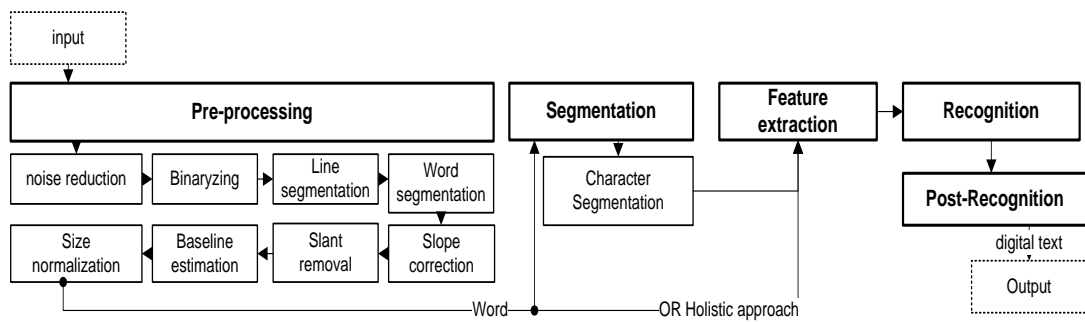
## **Chapter 2 : Literature Review**

Writer identification from handwritten Arabic text is not new area of research but it has attracted growing interest in recent years as a result of increasing terrorist acts that are, rightly or wrongly, associated with people of the Middle East and the rise of political Islam. This thesis does not attempt to prove or disprove this perception, but rather investigate the technical aspects of the problem and develop WI recognition schemes that take into account the distinguishing features of Arabic text writings. The ultimate beneficiaries of such a project include Historians and public organizations involved in the digitization of a wealth of religious/historic archives, but it is of interest to Forensics and terrorism fighting. In this chapter, we review the background material and the literature highlighting the main challenges in WI. We shall also critically discuss the various approaches adopted in the literature and briefly describe and justify our approaches and investigations conducted in this thesis. In section 2.1, we shall describe the concept of Writer identification, and then in section 2.2 we survey the literature on WI from Arabic handwritten text.

### **2.1 Writer Identification – Background and Related Issue**

Every part of a handwritten text reflects a certain writer's habit and style that can be a very effective tool to identify that particular writer. WI is based on analysing handwritten texts, which can be paragraph(s), line(s), word(s), character(s), and/or a part of a character (strokes). It is worth noting that Optical Character Recognition (OCR) has a close relation to WI from characters, but the two have different end objectives. (Bulacu, 2007)

OCR is designed mainly to recognise the letters and/or words within a given text. While WI targets the very special features, that reveal the habit and styles which in turn leads to the recognition of the writer. In other words, WI is not particularly affected by the text meaning/semantics. However, both applications have common tasks in their pre-processing and segmentation stages (see Figure 9 below). One of the main common task is to segment text into patterns (Lines, words, subwords letters, diacritics, and strokes). OCR system, as shown in Figure 9, is not concerned with specific patterns in their features like text slope and text slant which are considered as very important features for WI since they represent the most important writer's habits and style.



**Figure 9:** Typical OCR system

A number of references on OCR (handwriting recognition) are given in the bibliography section of this thesis ( (Lorigo & Govindaraju, 2006), (Kumar, et al., 2010), (Safabakhsh & Adibi, 2005), (AL-Shatnawi, et al., 2011), (AlKhateeb, et al., 2009), (Amin, 1998), (Baghshah, et al., 2006), (Berkani & Hammami, 2002), (Bar-Yosef, et al., 2009), and (Favata & Srikantan, 1996)).

In general, WI from handwritten text schemes are based on digital attributes that can be associated with words and letters/strokes are very popular among researchers, reflecting the knowledge acquired by existing research that individual writing habit/style is embedded in these parts of the text. Previously published literatures, seem to create the impression that a word's attributes result in a higher identification rate than attributes of characters or strokes (Awaida & Mahmoud, 2012) (Zhang & Srihari, 2003; Sreeraj.M & Idicula, 2011). The difficulty in segmenting and extracting letters and strokes from a script, because of the overlapping problem involved in Arabic handwriting, is probably another obstacle to investigating other than words-based WI schemes.

In fact, most existing research work have been based on separated letters' databases to overcome the overlapping problems, and researchers in this field revert to manually segmenting texts which are counterproductive to what happens in real life investigations. Comparing this to an automated system like ours which deals with the investigated text in its unity and then segment it automatically without corrupting it or its attributes. Automatic segmentation of handwritten text for the purpose of WI depends heavily on which part or elements of written texts are to be used for identification. In the next section, we critically review existing and relevant research on WI from handwriting in general and make simple arguments in support of our approach that deviates from the most common approaches by using subwords and diacritics as the most representative parts of writer's habit and style.

## 2.2 A Brief Survey of Recent Research in WI from Handwritten Text

There have been numerous researches conducted on writer identification for a variety of different languages. Approaches adopted by these researches are targeting a specific category of identification (i.e. on-line vs. off-line matching) or a particular text content and characteristics (text-dependent vs. text-independent) as described and explained in the previous chapter.

In most of these researches, writer discriminating features were extracted from entire pages, paragraphs, lines, words, or characters. Others were more interested in smaller parts than these features, like parts of characters or small strokes; these approaches have been used in different languages' like Latin, Arabic, Persian, etc.

In the following brief survey, we will explain the general characteristics of existing handwritten text analysis schemes. Bearing in mind that most of the literature presented below is based on off-line text approaches. The review is organised according to the choice of text component(s) that are deemed to be more writer discriminating and have been adopted for identification. Our review is not limited to Arabic text; due to the fact interest in WI from Arabic handwritten text is rather more recent. Moreover, one can benefit from many pre-processing procedures as well as classifiers that have been used in other languages.

### Work Based on Characters

Bensefia et al. (Bensefia, et al., 2005a), (Bensefia, et al., 2005b), (Bensefia, et al., 2002), (Bensefia, et al., 2003) proposed an identification and verification algorithm based on extracting Latin characters or part of a character's features. Connected components were extracted first, then segmented into possible strokes (*character or part of a character*) to generate graphemes (*grapheme is a letter of the alphabet, a mark of punctuation, or any other individual symbol in a writing system*). Grapheme k-means clustering was used to define a feature environment common to all documents in the database. Experiments were conducted on three intext-dependent databases that contained 88 books and 39 historical documents. These databases were written by 150 different writers. Writer identification was examined in an information retrieval framework while writer verification was based on the mutual information between the distributions of graphemes in the handwritings which were used for comparison. The results showed near 96% accurate verification. The same strategy used for grapheme clustering was also used for feature extracting (Schlapbach, et al., 2005).



Schomarker used similar approaches for analysing Latin fragments of text (characters or part of character) but based their work on Kohonen's self-organizing feature map (SOM). They (Schomaker, et al., 2007) (Bulacu & Schomaker, 2004a) (Schomaker, et al., 2004b) presented a writer identification algorithm by segmenting the text into fragments. The extracted features were based on connected component contours for these fragments after smoothing and binarising them. Then the Moore's contour was calculated. The Moore neighbourhood comprises the eight pixels neighbouring with a central pixel in a two-dimensional square matrix. The fragments 'connected component contour' training set was presented in relation to Kohonen self-organizing feature map (SOM). SOM is a type of Artificial Neural Network (ANN) learned to classify input vectors according to how they are clustered or grouped in the input space called a map or Kohonen map. SOM runs a technique of demonstrating multidimensional data in much lower dimensional spaces. (Kohonen, 1982).

The algorithm was tested on a text-independent western script database gathered from texts of 150 writers, and for any new sample it will return a ranking decision. K Nearest Neighbours (KNN) was used to find top1 writer at a 72% accuracy while the top10 writers yielded 93% accuracy rate. Also, Schomarker et al (Schomaker & Bulacu, 2004) presented the same strategy in (Schomaker, et al., 2007) (Bulacu & Schomaker, 2004a) (Schomaker, et al., 2004b) but for upper-case Western script.

Arabic and Latin writer identification and verification systems based on character extraction, textural, and allographic features were proposed by Bulacu et al (Bulacu, et al., 2007) , (Bulacu & Schomaker, 2006). Using allographic features as a writer style, they suggested first segmenting the text into characters based on the idea that there is no overlap between words and characters. The segmentation was done at the minima in the lower contour. Then using k-means clustering to generate a codebook. This codebook was considered as a training set of the graphemes extracted from the samples. Finally, this codebook was normalized by using Euclidian distance to produce one histogram for every similar character. This experiment was done by using in text-dependent DB written by 350 writers with 5 samples per writer. The best identification rate achieved here was 88%. They pointed out that the results obtained on Arabic are lower than the ones achieved in Latin texts.

Isolated Persian characters had been tested by Baghshah et al. (Baghshah, et al., 2006) for Persian writer identification. The images of these characters were pre-processed and then segmented into many strokes. Each stroke was described using a set of features like

stroke's direction, horizontal and vertical profile, and stork's measurement ratio. A combination of a fuzzy rule-based and the fuzzy learning vector quantization (FLVQ) had been used in order to identify the writer. Their proposed algorithm was tested on an in text-dependent DB which was written by 128 writers, and the results and accuracy rate were around 90% to 95% in different situations of testing.

Instead of working with alphabetic characters, Graham Leedham et al. (Leedham & Chachra, 2003) proposed an algorithm to identify Latin writers by extracting features from handwritten digits. These features included parameters such as height, width, the number of endpoints, the number of junctions, the degree of roundness, loop length, area, and centre of gravity, slant and number of loops. The system was tested on random strings of 0 to 9 written 10 times by 15 writers. Hamming distance was used for classification, with an accuracy rate of 95%.

The above algorithm seems to benefit from focusing on a smaller alphabet (10 numeral characters) that are visibly distinct, but it is of limited use. But the idea of extracting similar numerical features for a small writer discriminating subset of the alphabet for a WI is interesting, and research into determining such a subset of any language can provide improved accuracy. In fact, Maaten et al (Maaten & Postma, 2005) have analysed just two special Latin characters 'th', which it is written as one word but not separated characters, and developed a writer identification algorithm by combining statistical and model-based approaches. Their proposals were to extract directional features and codebook of graphemes. The method was examined on texts written by 150 writers, and the WI rate was 97%. In this thesis, we take such an approach later on but to select, through experimentation, the smallest set of subwords rather than characters for WR in Arabic.

The extracted features in the above publication are based on analysing the spatial domain of the scanned text documents. Gazzah et al. (Gazzah & Ben, 2006) presented an Arabic writer identification system using a combination of global features that are extracted from the frequency domain of the like: Wavelet transforms and entropy, and structural features like: Line height, spaces between subwords, inclination of the ascenders, and dot boldness and shapes. The performance of their proposed algorithm was tested on a DB of 180 handwriting texts including letters, numbers and punctuation marks. Scripts were written by 60 people who copied the same character 3 times. Multi-Layer Perceptions (MLP) classifier was applied to recognize the writer with an accuracy rate of 94.73%.

In another work of Gazzah et al (Gazzah & Amara, 2007) a two dimension (2D) discrete wavelet transforms DWT-Lifting scheme was used for feature extraction, along with the MLP classifier, achieving 95.68% accuracy rate. The authors had an interest in selecting a classifier that could get the best out of the wavelet-based features. They concluded in their latest paper (Gazzah & Amara, 2008) that MLP gets better results compared with Support Vector Machine classifier (SVM) which it had been examined as well.

Abdi and Khemakhem (Abdi & Khemakhem, 2010) introduced an algorithm to recognize writer identification through extracting six features from Arabic strokes. These features were based on length, direction, angle, and curvature. Before extracting these features, a number of pre-processing stages were conducted including binarising, removal of diacritics, morphological dilation, connected component extraction and contour extraction. The system was tested on a selection of texts collected from 82 writers chosen from IFN/ENIT DB. KNN was used for classification while several distance measures were examined:  $\chi^2$ , Euclidean, standardized Euclidean, Manhattan, Mahalanobis, Minkowski, Hamming and Chebechev. The best result the system recorded 90.2% accuracy rate for top 1, compared with 97.5% for top10. (Abdi & Khemakhem, 2010)

### **Diacritics-based WI**

To the best of my knowledge the only work that dealt with Arabic diacritics (smallest pattern in Arabic text) was carried out by Lutf et al (Lutf, et al., 2010), who proposed an identification algorithm by extracting the basic components (diacritics). They suggested applying pre-processing stages to the handwriting document like de-noising and thresholding. Their segmentation algorithm extracts the diacritics in a text by removing the main text from the image whilst keeping the diacritics untouched. They used the vertical and horizontal projection profile to extract the diacritics based on estimated baseline. For feature extraction process, Local Binary Pattern (LBP) histogram is calculated for each and every diacritic, then all these histograms concatenated as a single histogram feature. The LBP is the method used to extract a texture from an image. It has many different versions. In its simplest version, it replaces each image pixel value by an 8-bit byte formed by comparing the pixel values in its 3x3 neighbourhood in a clockwise manner. Starting from the top left corner, it sets 0 in the current bit position if the pixel value is less than the central pixel, 1 otherwise. The LBP image encapsulates the texture in the original image, but a compact version of the histogram of the LBP image that has

been used as a feature vector for face recognition and pattern recognition in general. The histogram of the LBP of an image has 59 bins corresponding to 58 uniform patterns (binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is traversed circularly) and 1 non-uniform for all other patterns. For more detail see (Ojala, et al., 1996), (Ojala & Pietikäinen, 1999), and (Ojala, et al., 2002). The authors did not follow this pattern recognition tradition on the number of histogram bins, but rather use the usual histogram with 256 bins.

For classification, they used KNN with  $X^2$  as a distance function. The system was tested on a selection of texts of 287 writers chosen from the IFN/ENIT DB. They claimed that the WI reached an accuracy rate of 97.56% (Lutf, et al., 2010). They also claimed that the WI rate increases if the LBP code is changed from 255 into 256, where the accuracy rate will jump from 51.22% to 97.56%.

### **Words based WI**

By far this is the most researched approach for many languages. In most cases, researchers attempt to work with a specific small set of selected words. WI based on the single word "characteristic" written in English and Greek languages was performed by Zois et al. (Zois & Anastassopoulos, 2000). They have tested their algorithm on a text-dependent DB of 50 writers, who had been asked to copy the same word 45 times. The images of the scanned text were binarised and morphologically thinned. Horizontal projection profiles were constructed, divided into 10 segments and processed using morphological operators on two scales to obtain 20-dimensional feature vectors. Bayesian and neural networks were used as classifiers. This system showed an accuracy rate of 95% for both English and Greek words.

Another system developed by Zhang et al., (Zhang & Srihari, 2003) used a group of Latin words. These words were: 'been', 'Cohen', 'Medical', and 'referred'. The words were extracted from a text copied three times by 1027 writers. The features extracted from these words were: Gradient, structural, and concavity. They used Knn for classification. The WI obtained an accuracy rate of 83% while the verification accuracy was 90.94%. The authors concluded that entire handwritten words achieve a better identification rate than using characters.

Tomai et al. (Tomai, et al., 2004), presented writer identification algorithm from English words' features. Twenty-five different English words were written by 1000 writers copied three times. This set of words were {"From", "Nov", "10", "1999", "Jim", "Elder",

“829”, “Loop”, “Street”, “Apt”, “300”, “Allentown”, “New”, “York”, “14707”, “To”, “Dr”, “Bob”, “Grant”, “602”, “Queensberry”, “Parkway”, “Omar”, “West”, “Virginia”}. From these words a number of features were extracted: Gradient, Structural and Concavity (GSC), Word Model Recognizer (WMR), Shape Curvature (SC), and Shape Context contour shapes (SCON). They concluded that longer words improved performance using their algorithm. Using KNN classifier, the algorithm achieved 66% accuracy rate for top5 writers. While, words containing the letters (G and F) got 67% for top5. Words based on Gradient features got an improved accuracy rate for top10: 82% for verification and 62% for identification.

Limiting the number of words for WI was also applied in other languages such as the Chinese language. In fact, Zuo et al. (Zuo, et al., 2002) have adopted the well-known dimension reduction scheme of Principal Component Analysis (PCA) for a Chinese handwrite identification scheme. Using a text-dependent DB consisting of 40 words, which were copied 10 times by 40 different writers. Half of the DB was used for training the system; the other half was used for testing. The best result achieved for a single word was 86.5% accuracy rate, while a combination of 10 words achieved a 97.5% accuracy rate.

Al-Ma'adeed et al. (Al-Ma'adeed, et al., 2008) proposed an Arabic identification algorithm based on features extracted from scanned images of words'. These features are multi-angled edge directions, moment invariants, and what are known as word measurements, also referred to as word structural features like: area, length, height, length from baseline to the upper edge, and length from baseline to the lower edge. Text text-dependent DB has been used in this work which contained 27 words copied 20 times by 100 writers. A quarter of this DB was used for testing while the rest was used for training. K-nearest neighbour was used as a classifier. Only the top10 identifications were presented obtaining an accuracy rate of 90% for specific words while for other words the accuracy rate results were between 53 to 75%. They did not mention the progress rate from top1 to top9. The words that obtained high rates were:

ولكم جزيل الشكر , تحية طيبة وبعد , السلام عليكم ورحمة الله وبركاته , بسم الله الرحمن الرحيم

Naturally, this last work is very important to our work due to the fact that it is about Arabic handwritten text and provides us with a benchmark for comparison. In fact, using their database provides the opportunity to compare the performance of our developed WI scheme.

### **Line based WI research**

The above schemes may seem naturally adding to the challenge of WI. Why should we focus on the small proportion of writer's text when we could benefit from a richer and longer written combination of words? Analysing an entire text line is a more habit revealing than working with a small subset of it. Moreover, some of the necessary but difficult segmentation procedures may become less demanding. In this subsection, we should focus on WI from a line of text. Marti et al. (Marti, et al., 2001), presented a writer identification algorithm based on English text line features. Twelve local features, which are derived from three global line features, were extracted. These global line features were zones, slant, and character width. A text-independent dataset consisting of 100 pages written by 20 different writers was involved to examine the proposed algorithm. Two classifiers were used to identify the writer, when applying KNN they managed to achieve a success rate of 87.8%, while by using Artificial Neural Networks (ANN) classifier they achieved 90.7%.

Another research of English text line was presented by Hertel and Bunke (Hertel & Bunke, 2003) to identify writers using an in text-dependent DB taken from a benchmark IAM DB (Marti & Bunke, 2002). This DB was collected from 50 writers who wrote 5 pages each and were chosen for this task.

Features extracted from the single line were: Distances between connected components, the blobs enclosed inside ink loops, the upper/lower contours and the thinned trace processed using dilation operations. Identification rates exceeded 90% using the KNN classifier.

Another example of line-based WI research was done by Rafiee et al. (Rafiee & Motavalli, 2007) for Persian language who proposed using an off-line text-in text-dependent DB. They managed to extract eight features from each line image. These features were derived from height and width of the text. The system was trained by using text written by 20 writers each writing 5 to 7 text lines. Neural networks had been used for classification to gain an accuracy rate of 86.5%. They conclude that the line text of unsteady writer was not suitable for their WI system.

### **Work based on Paragraph (page or document)**

The next natural focus of WI from written text is what some researchers have attempted by extracting features from paragraphs/pages. In other words, this could be considered as a fusion at the feature level of a number of the same line-based WI schemes but with a number of lines.

Shahabi et al. (Shahabi & Rahmati, 2006) presented an Arabic/Farsi off-line identification system based on text-independent page. This system used one A4 page which was written by 25 writers. Every page was segmented into 4 blocks with three blocks used for training and one for testing. The pages were pre-processed then they extracted features using Gabor filters. Euclidean, Weighted Euclidean, and  $X^2$  distance were used as distance functions. Their latest work, using a text-dependent DB written by 40 writers, (Shahabi & Rahmati, 2007) reported that top1 got 82.5% identification rate accuracy.

Al-Dmour et al. (Al-Dmour & Zitar, 2007) proposed a page-based Arabic writer identification scheme by using different feature extraction methods such as hybrid spectral-statistical measures (SSMs), multiple-channel (Gabor) filters, and the grey-level co-occurrence matrix GLCM. These features were used to find the best subset of features. Many classifiers were used in their experiments such as Liner Combined distance (LDC), Support Vector Machine (SVM), and KNN. All of these classifiers produced accuracy rates between: 57.0%, 47.0%, 69.0% and 90.0% respectively. Their system used a database collected from 20 writers, who were asked to write 2 copies of an A4 document, one was used for training the system and the other for testing

Helli et al (Helli & Moghaddam, 2008a) (Helli & Moghadam, 2008b) (Helli & Moghaddam, 2009) (Helli & Moghaddam, 2010) developed another page-based but “text- independent” writer identification system for both Farsi and Latin languages. They utilised Gabor filters for feature extraction. To test the system they asked 100 writers to write 5 different pages in Farsi while they selected handwritten texts of 30 other people in English from the IAM database (Marti & Bunke, 2002) where each writer was asked to write 7 different pages. For the purpose of experimenting they used 60% of the material for training their system, and the rest was used for testing it. They reported that top1 got 98% accuracy rate for the Farsi test, while 94.4% accuracy rate was reported for the English test.

Other researchers who have developed an off-line Farsi text-independent WI system are Ram et al. (Ram & Moghaddam, 2009). Their work is based on gradient features which were used in the past with Latin script (Tomai, et al., 2004) (Zhang & Srihari, 2003). Their system was tested on 250 handwritten samples, which was gathered from 50 writers who wrote 5 different sheets each. They reported that they had achieved 94% accuracy rate when using neural networks as a classifier, against 90% when using Fuzzy clustering classifier (Ram & Moghaddam, 2009).

The last few researches in this section as well as some works in the previous sections achieved higher WI accuracy rates, confirming the expectation that the availability of more samples and/or longer text from the same writers capture more of writer learnt habits and style in writing in any language. However, WI from shorter text and/or fewer samples is by far the more challenging and is in more demand these days for applications like crime/terrorism fighting and Forensics.

### **Work Used a Combination of text parts**

Little research was conducted using mixed parts of the text to gain better identification performance. To some extent, this can be classified as attempts to apply fusion at the feature level from different numbers of the above problems. One of the very few researchers who delved in this is Said et al. (Said, et al., 1998) (Said, et al., 1998) who proposed an English text-independent algorithm using multi-channel Gabor filter and grey-scale co-occurrence matrices as writer features based on text line and word attributes. Gabor filters are commonly used to capture/highlight directional texture in images. They used in their experiments frequencies of 4, 8, 16 and 32 cycles/degrees. For each central frequency  $f$ , filtering is performed at  $\theta = 0^\circ, 45^\circ, 90^\circ, \text{ and } 135^\circ$ . Then the distance between text lines/words and text padding were extracted. Finally the mean and the standard deviation of each output image are calculated. For the purpose of experimenting they employed a DB made out of handwritten texts of 20 different writers and 25 samples per each writer. For classification, they used nearest centroid method using weighted Euclidean distance and Gabor features. This scheme achieved 96% accuracy rate. The same algorithm has been applied to printed documents for script recognition (Tan, 1998), and font identification (Zhu, et al., 2001).

The Srihari et al (Srihari, et al., 2001), work which was funded by National Institute of Justice in USA, presented an identification algorithm by using ANN in WI through



extracting a number of features from pages of text, They suggested extracting 21 features to help increase the accuracy rate.

These features can be classified into two categories: global (macro) and local (micro) features. The macro features, which work with a document, paragraph and word level, the parameters used were: grey level at the document level, the number of pixels, ink, a number of inside and outside features, a number of components of the slope four trends, higher average / italics, paragraph aspect ratio and indentation, the length of the floor and upper / lower zone ratio.

In the micro-features, which work at the word and character level, the parameters comprise of Gradient, Structural, and Concavity (GSC) attributes. These features were used originally for handwritten digit recognition (Favata & Srikanthan, 1996). The evaluation test was carried out on texts written by 1500 writers who wrote 3 copies of a pre-determined text (Text-dependent dataset) of 156 words (the CEDAR letter database).

The micro-features outperformed the macro features in WI tests with an accuracy exceeding 80%. A multilayer preceptor or parametric distribution was used for writer verification with an accuracy of about 96 %. They later developed a dependent Arabic writer verification system based on the above technique, For the Arabic language system macro and micro features were extracted from 10 different handwritten pages, written by 10 different writers. They reported that the Arabic system can verify the writer with an average of 86% accuracy rate.

### **Our Approach**

Most of the above research works seem to ignore or pay little attention to, specific distinguishing characteristics of Arabic text in relation to structures of words. Words in the Arabic language and few other similar script languages are constructed from one or more subwords and may have multiple diacritics associated with them. Each subword and diacritic may have different recognizable attributes that reflect different writer's habits and styles which may also contradict each other to produce fake features. On the other hand, many subwords occur in different words. This implies that there would be more repeated versions of subwords than words in a text which is expected to provide better opportunities for writer identification. Moreover, many single letters appear as a separate subword. In chapter 3 we present a detailed analysis of the structure of Arabic writing which support this claim about subwords plus other characteristics that motivate placing more emphasis on subwords for WI from Arabic text. In fact, our investigations will be

automating and exploiting all properties of subwords to obtain small subsets of the very large set of subwords sufficient to achieve desirable accuracy level that outperform the state-of-the-art. Our investigations will also investigate the role of diacritics in WI for Arabic text and in particular to answer whether their inclusion within the subwords improves or impair WI accuracy.

### 2.3 Databases we used

We have used three databases in the experiments we have conducted for this thesis. Two are “publicly available” text-dependent DBs, namely IFN/ENIT and Al-Ma'adeed and one is our own in text-dependent DB, which was gathered in-house by us at the University of Buckingham.

The IFN/ENIT DB, with a resolution of 300 dpi binary handwritten words, contains 26,459 images of single words representing Tunisian town names, written by 411 different writers. In total IFN/ENIT contains more than 212,211 letters (Pechwitz, et al., 2002). An example is shown in Figure 10.

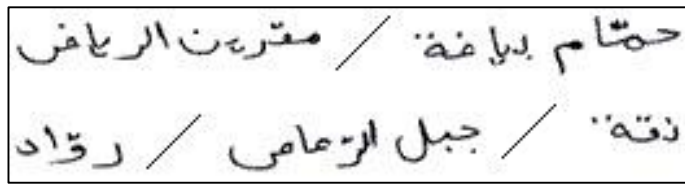


Figure 10: An example of IFN/ENIT database (Pechwitz, et al., 2002)

The second text-dependent DB we employed was collected by Al-Ma'adeed et al. (Al-Ma'adeed, et al., 2008). This DB was made of a group of 27 words derived from 16 commonly used phrases/sentences written by 100 writers who were asked to write repeatedly these words for about 20 times. Examples of these words include:

على، عن، بخير، في، هي، هو، لك، المحترم، التوقيع، من، ولكم، جزيل، الشكر

An example is shown in Figure 11

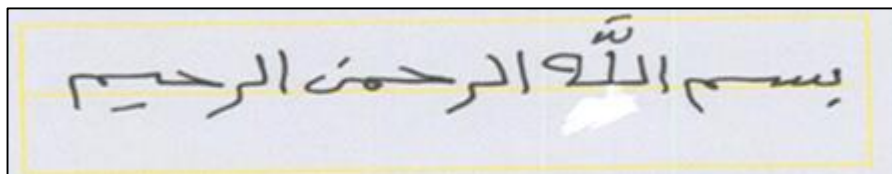


Figure 11: An example of Al-Ma'adeed database (Al-Ma'adeed, et al., 2008)

Unlike the two text-dependent DBs mentioned above, our in-house DB contains 120 multi-sentence documents written by 50, randomly selected writers aged between 8 and 85 years. Each writer wrote 2 different pages; each page is made of an average of 2 paragraphs.

This DB has two given texts, on average:

- Text1 has 6 lines consist of (50 Words and 120 subwords) and
- Text2 has 5 lines consist of (35 Words and 100 subwords).

Examples are shown in Figure 12 and Figure 13

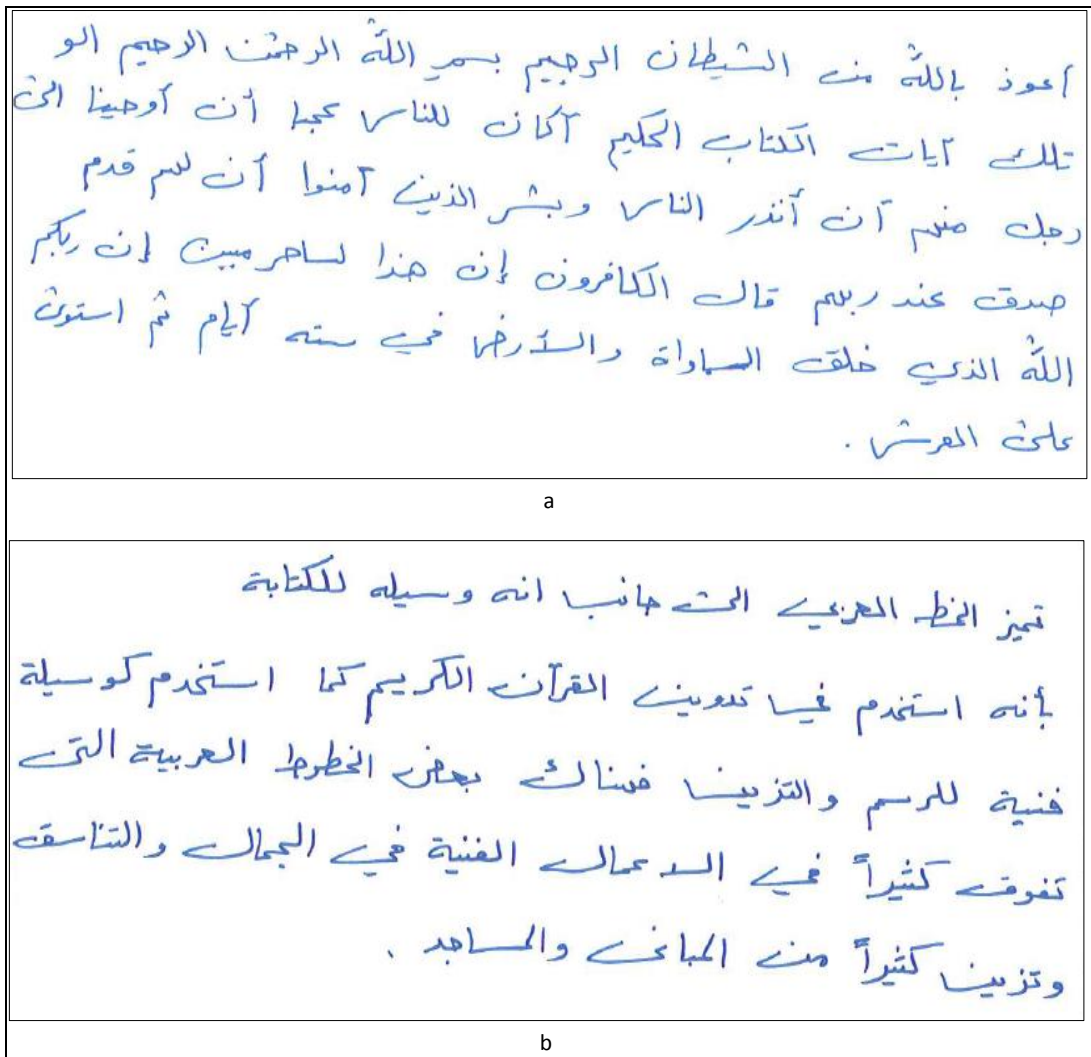


Figure 12: In-house database. a: text 1, b: text2

تميز الخط العربي - الى جانب انه وسيلة للكتابة - بأنه  
استخدم في تدوين القرآن الكريم استخد كوسيلة فنية  
للرسم والتزيين .  
فهناك بعض الخطوط العربية التي تفوق كثيرا من  
الاعمال الفنية في الجمال والناسق ، وتزين كثيرا  
من المباني والمساجد .

a

أعوذ بالله من الشيطان الرجيم بسم الله الرحمن  
بسم الرحيم المر تلك آيات الكتاب الحكيم  
أكان للناس عجباً أن أوحينا إلى رجل منهم  
أن أنذر الناس وبشر الذين آمنوا أن لهم  
قدم صدقٍ عند ربهم قال الكافرون أن هذا  
لساحرٌ مبين ان ربكم الله الذي خلق السموات  
والارض في ستة ايام ثم استوى على العرش

b

Figure 13: In-house database, another example, a: text 1, b: text 2

## 2.4 Latest Work

Aboul-Ela et al. (Aboul-Ela, et al., 2015) presented WI based on English and Arabic phrase, word, and character level. Their proposed algorithm tested an off-line text-dependent DB which constricted by 50 writers who had been asked to copy five times three different English and four Arabic text lines. An automatic segmentation system was implemented at the line-level while semi-automatic was done at the word-level. But at character-level a manual segmentation system was performed. Therefore, a total of 31

words (14 English and 17 Arabic words) which segmented as well into 37 characters (18 English and 19 Arabic characters).

Many features were used which are derived from three main features to create the features vectors. These main features are Geometric moments, Features that used with signature matching (like slant, width, height feature), and Features that derived from fractal analysis.

As classification, Knn technique was used to find the WI rate, Table 2.1 shows their system accuracy got based on phrase, word, and character level both in English and Arabic languages.

**Table 2.1:** (Aboul-Ela, et al., 2015) system accuracy

Language	Phrase-level	Word-level	Character-level
English	Top10=86.8	Top5=92.4	Top1=92.8
Arabic	Top10=85.6	Top5=96.4	Top1=86.8
English&Arabic	Top10=94.8	Top5=98.4	Top1=99.6

They also concluded that Arabic text might get better accuracy rate than English due to the variety of Arabic script shapes which cause in most cases into the individual writer habit and style. In addition, exclude specific handwriting samples lead increasing WI.

Moreover, they tested their algorithm using different benchmark English DB (IAM DB) for 470 writers. The best top10 result they obtained was between 40.43, while using 66 writers for specific features got in top10 between 51.09% to 80% WI rate.

Wu et al (Wu, et al., 2014) presented WI based on word level. Six bench DBs were uses to test their algorithm centred on five different languages, (English, Germany, French, Greek, and Chines). An automated segmentation process was used to segment the handwriting image texts into words using an isotropic LoG filter. Then, the scale invariant feature transform (SIFT) descriptors (SDs) and the corresponding scales and orientations (SOs) are extracted for each word image text.

The words therefore, categorized in the training stage by constructing a codebook based on an algorithm to detect a number of key points and extract their descriptors, scales, and orientations.

They concluded that the word-level features of handwriting are much more suitable for WI than page-level and allograph-level features

Knn was used for classification where obtain in top1 between 80.8% to 98.5 using different DBs.

## **2.5 Summary of the Chapter**

In this chapter, we tried to search for and list most of the past efforts which were concerned with WI. We also came to notice that that WI was historically based on analysing paragraph(s), line(s), word(s), letter(s), and/or a part of a letter.

In general, WI systems that are based on words and letters are very popular among researchers in this field. We, however, have adopted a totally diverse way of investigation basing our work on the part of a word which we term as ‘subword’ with/without ‘diacritics’. And contrary to the trend, we have also set our work to be carried out on intext-dependent DBs rather than text-dependent DBs like others did. The reason behind that is the fact which we believe in that in real life investigation it is not practical to obtain repeated samples from the same writer.

## Chapter 3 : Pre-processing and segmentation of Arabic Texts

Pre-processing and segmentation are essential steps to extract the important and relevant feature vector from the available handwritten text sample to be used to identify text writer. Pre-processing in this respect aims to prepare the digital input scan of the Arabic handwritten text for the sought after segmentation of the written text. It consists of steps that are particular to (1) the process of scanning which may introduce artefacts or noise, (2) the nature of handwriting in general such as the misalignments of the written lines, and (3) the structure of Arabic text such as the presence of overlaps between text line, words, subwords, letters, and between diacritics. The last two types of pre-processing challenges derive from the way the candidates write their text, the type of the pen used, the pen pressure applied, the font size, etc.

Segmentation is the process of partitioning the written text in the scanned image into the components that are to be used in the identification procedure. It is necessary to isolate the very important patterns in the text from which feature vectors are to be extracted, compared and matched with the existing template vectors. In this chapter, the pre-processing and segmentation challenges and the proposed algorithms to solve them will be explained in details. But, we first need to review the structure of Arabic text components in order to provide further justification for our decision and to guide our work on the later section on segmentation.

### 3.1 Arabic Language Text Analysis

Words in Arabic language and other similar script languages consist of different types of letters that differ in connectivity characteristics. There are 6 letters [أ، د، ذ، ر، ز، و]، out of 28, in Arabic language that do not connect with the letter that follow them in a word, the part of the word that ends with one of these 6 letters will create a subword (see Figure 14.e). So a single word might consist of one or more subwords. Moreover, the subword might also be a single letter. The other 22 letters tend to connect with each other through their horizontal line (see Figure 14.a & b). On the other hand, some letters may have diacritics that distinguish them from other letters of the same shape (as demonstrated in Figure 14c & d). The shape of the diacritic can either be a small dot, a number of dots or minor symbols that can help with pronunciation of the words as presented in Figure 14c.

Note that Kurdish, Persian, Urdu, and other languages similar to Arabic have common features with Arabic writing structure, as shown in Table 3.1 and all these languages are read from right to left. One of the main features of these languages is that the word might consist of many subwords as in Arabic.

In Arabic texts, a word consisting of multiple subwords separated by narrow gaps, as shown in Figure 14.b (in the figure words are indicated by overlies (numbers 1 to 5 are words)). The presence of these narrow gaps between subwords within words is compulsory contrary to Latin languages. Note that these narrow gaps are smaller than the usual space symbol which is used by the writer to separate words. This can be useful in the segmentation of subwords in that subwords of a given longer word would have two gaps on either side of it in the text at least one of which is narrower than the average space between words. However, there are two problems with this. First of all, some of the patterns that appear as subwords can also appear as separator words (see section 3.1.1). Secondly, this depends on estimating the average size of the writer space that he/she uses in his writing. Thus, we will not pursue this any further but we need to use other known and easy to use characteristics of subwords in segmenting Arabic text.



Figure 14: Printed Arabic sentence and its structure



**Table 3.1:** Arabic vs. Other languages' subwords (Good morning phrase)

Language	Phrase (Good Morning)	Disconnected Patterns
English (Cursive)	<i>Good Morning</i>	G ood M orn ing
Arabic	صباح الخير	صباح   ا   خير
Kurdish	به یانیت باش	به   یا   نیت   باش
Persian	صبح به خیر	صبح   به   خیر
Urdu	گڈ مارننگ	گڈ   ما   ر   ننگ

### 3.1.1 Subwords Characteristic

By using subwords rather than full words, it is expected that the process of WI will benefit from having more samples to work with. The probability of repeating a particular word in the same text sample is relatively low while subwords can be found many times even within dissimilar words. Furthermore, the probability of finding subwords will increase dramatically if diacritics are omitted. This is due to the fact that are many similar subwords that only differ in the way their diacritics (if any) appears. Besides, subwords usually have connected strokes which reflect writer's habit and style in a precise manner. These specific characteristics will be discussed in details in the coming sections.

#### 3.1.1.1 Repeated Subwords in a Text

A specific subword can repeatedly be found in dissimilar words as shown in Table 3.2 (We use colours to indicate repeated subwords in the different text). Subword (في) in Table 3.2 column (1) is repeated in column (2) that has different words. In the same table, subword (في) in column (3) is exactly similar to the one in column (4). Interestingly, in this table, if we ignore/omit two different diacritics appearing in the upper parts of the 4 different words we have only one subword in common.

**Table 3.2:** Subwords repeated in dissimilar words for (subwords: في and في) first example

1	2	3	4
وفي	وافي	رقي	راقفي

Moreover, some subwords that are similar in shape can be found embedded in many different words with different meaning: example of these subwords (م), (يم), and (سي) when they are separated from the original word as shown in Table 3.3, Table 3.4, and Table 3.5 respectively. Subword (م) is repeated 22 times once separated from the original words. But when using a full word, it is rare to find similar word repeated in the same text.

Table 3.3: subwords (م) is repeated in a number of different words

م					
الهام	علام	محزوم	مثنوم	ملوم	يلوم
برشام	غلام	مرسوم	معلوم	مهضوم	يوم
محكوم	متموم	مسموم	مفهوم	مهموم	
هشام	محروم	مسنوم	مكتوم	نجوم	

Table 3.4: subwords (يم) is repeated in a number of different words

يم					
تمايم	صايم	تقديم	قديم	قسايم	كريم
مريم	نديم	صائم			

Table 3.5: subwords (سي) is repeated in a number of different words

سي					
اختلاسي	اساسي	حماسي	خماسي	راسي	سداسي
قاسي	كاسي	كراسي	ناسي	مرسي	راشي

Examples of subwords that are similar in shape which can also be found embedded in the same word see Table 3.6 in columns 1: as (و), 2: as (مر), 3: as (بر), and 4: as (ود).

Table 3.6: Subwords repeated in same word

1	2	3	4
و	مر	بر	ود

### 3.1.1.2 High Level of Subword Repetition

Another kind of subwords that might bring higher repetition in a text is found in dissimilar words (as well as within a word) if diacritics are excluded. For example subwords (في) is found 9 times in Table 3.7 if the diacritics are not included with their subwords while it might be found 2 to 3 times if the diacritics are included. Such approach will be discussed in chapter 5.

**Table 3.7:** Example of High Subword repeated in dissimilar words if diacritics are not included

في				
وفي	فيء	رقي	ساقى	في
وافى	قيء	رُقي	راقى	

### 3.1.2 Subwords with Stroke That Reflects Writer Habits

A very important subword's characteristic, which is also very important for our research, is the stroke that subwords may end with. Strokes are connected patterns that are in many cases predicted to reflect the habit and style of the writer, and therefore it is advisable to be taken into consideration for WI from Arabic texts.

Huber et al. (Huber & Headrick, 1999) considered the Connected Stroke (CS) as a writer habit. In their book, 'Handwriting Identification: Facts and Fundamentals', they mentioned that Osborn says (Huber & Headrick, 1999) that specific features like slant and slope of CS are among the most important variations in Latin handwriting. Moreover, Osborn, Harrison, and Ellen (Huber & Headrick, 1999) illustrate that the strokes in the English language that have a bowl or circular component such as (a, c, d, g, etc.) and are of considerable value for WI, as shown in Figure 15 below.



**Figure 15:** Examples of Connecting Stroke (CS) of lowercase Latin strokes (Huber & Headrick, 1999)

The following Arabic subwords: (لك، لى، على، هو، من, etc.) that have a bowl or circular components that can be investigated to see if they reflect writer habit and style.

In Arabic and similar script languages, the end parts of subwords are CSs too and is expected to reflect the writer habits and style in a similar way to Latin strokes. Examples of Arabic CS are shown in Figure 16. These parts of subwords are reflected implicitly without segmenting them from their subwords. However, these strokes (tails) can be used for identification individually, and we are not going to use them in this thesis explicitly.



**Figure 16:** Examples of Arabic Connecting Stroke (CS)

In summary, all these facts justify our claim that subwords are rich with a vital source of information that should be investigated for their writer discriminating feature to be employed in the process of Arabic handwriting WI.

### **3.2 Automating Writer Identification**

Automating writer identification is a tough challenge due to many factors including the behavioral nature of such a biometrics that obviously change over time and/or through training. Experts are working on writer identification usually develop very complex skills that help them identify the writer of a written text. Transferring and automating these skills need a great deal of considerations, starting from extracting the most writers discriminating features to use these features to train the system for writer identification.

### **3.3 Common Challenges**

In order to extract the handwriting features of a writer from a text, different challenging processes have to be followed and applied. Our entire thesis is based on using subwords with/without diacritics (disconnected patterns) in WI. In order to extract these disconnected patterns, a segmentation process should be followed and applied to the scanned samples of the given Arabic text. In most cases, this text is provided as a page, paragraph(s), or line(s). The most obvious, but difficult, challenges that complicate segmentation tasks can be listed as follows:

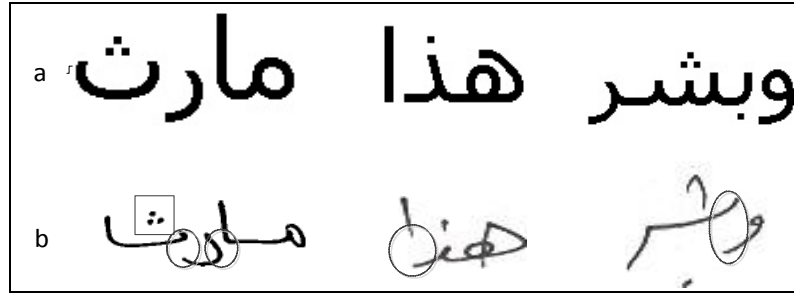
1. Text overlapping that occurs in handwriting, especially in Arabic language, between the lines, words, and in particular between subwords and diacritics.

2. Text orientation which may be variable throughout the scanned written text between different line(s), word(s), subword(s), and diacritic(s).
3. Different types of image noise that may corrupt the scanned text due to use different paper quality, variation in ink quality, and the quality of the used scanners.

Our experience, which can be sensed from various publications, indicates that text overlapping is the most common occurrence in Arabic handwriting. The fact that text overlapping may complicate other factors such as confusing diacritics with each other and deciding to which part of the overlapped text a diacritics belong. Moreover, the orientation problem and the presence of noise both contribute to the difficulty of removing/resolving overlaps. Adding to that, procedures that could be used for resolving overlaps may introduce other cases of artefacts. Therefore, text overlap resolving must deal with the two problems as part of the task of resolving text overlap. Recall that our intention is to introduce an algorithm that will segment text into subwords with/without diacritics successfully while preserving the required text component(s) intact.

### **How does overlapping appear in Arabic handwriting?**

An answer to this question is essential for overcoming this challenge. Text overlapping appears when the writer unintentionally places two or more characters so near each other so that their assumed spaces intersect. In most cases, the writer is unaware and/or doesn't pay too much attention. For a human, this may not affect their reading of the text, but automatic text detection may mistake the part of one of the characters as diacritics of the other. Figure 17.b shows examples of overlapped text found between subwords (highlighted by the circular area) and between diacritics (highlighted by the rectangle area). It is worth noting that this is not a serious problem for the human experts who would normally use their knowledge of the language in disentangling overlapped subwords.



**Figure 17:** Examples of subwords overlap. a: Arabic Printed text, b: Handwritten version with marked overlapping.

Unless, writers use lined papers and carefully adhere to write within the printed lines, it would be somewhat difficult to keep to a baseline. Entire text lines or part of a line (consisting of a single word or multiple words) may become misaligned with each other as the writer proceeds causing the overlap of subwords within a word or across lines. Note that, writers often attempt to frequently re-align their written text. Consequently, the scanned text often suffers from variable orientation within each text lines and the entire page. Therefore, our intended algorithm must address the problem of variation in orientation of a line of text and across lines. An orientation related word/subwords feature used in WI is the slope of such components which an indicator of writing habit and style. It is, therefore, essential that resolving the variable orientation problem must not destroy subwords slope feature. In fact, the entire process of text re-orientation and line segmentation will not, ultimately, be necessary for the purpose of our hypothesis as we are concerned only with segmenting connected component patterns for subwords bodies and their diacritics.

### **Text noise – How does it appear?**

Noise in any image relates to the appearance of pixels that has different intensity significantly compared to its neighbourhood. Noise is usually more noticeable in smooth areas of the image, but it corrupts the entire image including significant features such as edges. By text noise, we mean all pixels scattered around the text that do not belong to any of the genuine text components. It is produced as a result of scanning a text written on different kind of paper using different kind of ink quality. The resolution level of the scanner itself can also produce undesired noises in and around the text. However, during the text image for segmentation our attempt to resolve text overlapping and other pre-processing steps may introduce image artefacts that could have the same effect of text noise in that it does not belong to the genuine text components. Therefore, our de-noising procedures should be applied after the pre-processing stage.

### 3.4 The Developed Solution

To resolve of the above challenges, we apply a number of commonly available pre-processing procedures in the scanned text image preparation and prior to the segmentation stages, including de-noising, and text orientation.

Pre-processing aims at producing data that makes it easy for the system to operate accurately in WI. The first pre-processing task would be the binarisation of the text image to distinguish genuine text pixels from non-text ones.

#### 3.4.1.1 Binarisation

Unlike written text analysis from electronic tablets, paper written text cannot benefit from pressure information due to the difficulty of extracting such features. However, variation in pressure contributes to creating different shades of pixel gray values. Moreover, digital scanners output greyscale images of different shades of intensity throughout the text and non-text pixels. The banrisation task refers to the conversion of scanned greyscale text image into a binary image by turning all pixels below a selected threshold to zero and all other pixels to one as shown in equation ((3.1) below

If  $g(x, y)$  is a threshold version of  $f(x, y)$  at some global threshold  $T$ ,

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

The process of converting text image into the binary is used to remove and clean background noises as it has different tones, which may affect the segmentation process.

There are many techniques to select a threshold. These techniques can be classified into global and local threshold, see (Morse, 2000).

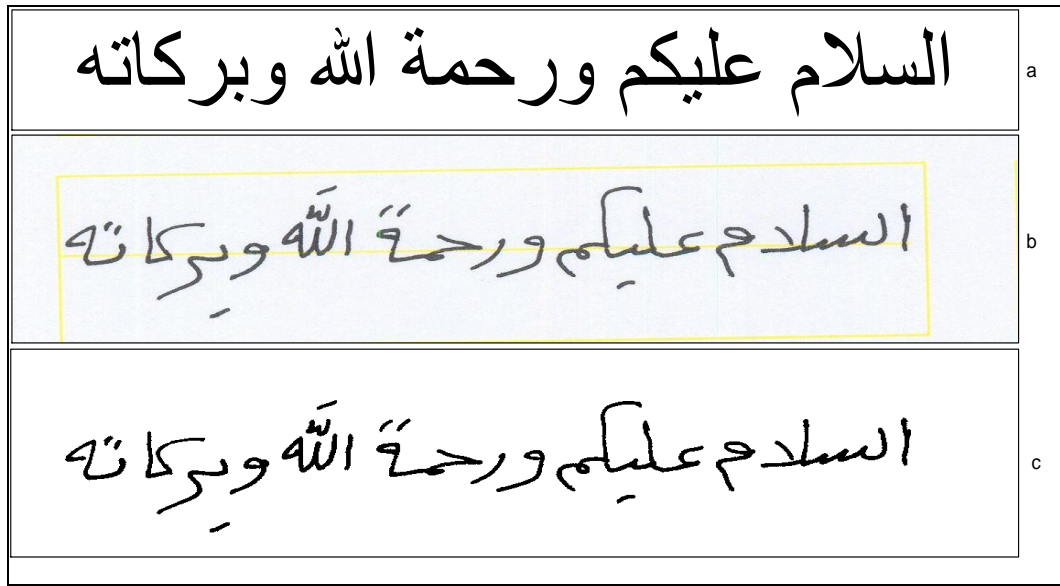
The choice between Local and global thresholding was determined by testing their observed effect on several text image samples that are chosen randomly from a database which has been collected especially for this work.

#### **Global threshold:**

A global threshold is a single threshold value obtained from and applied to, the entire text image. This threshold is based on estimating the background level from the intensity histogram of the image. Otsu's method, which is a well-known global threshold (Otsu,

1979) is used to accomplish objectively clustering-based image thresholding. The algorithm automatically estimates a single global threshold value for a bimodal image (bimodal image is an image whose histogram has two peaks) by finding out an average value of these peaks.

Figure 18.c shows a binary image based on global (Otsu, 1979) threshold. For reliable comparison, we use a printed text which is a clear binary image. As one can see, the binarised image of the scanned handwritten version of the same text has similar image characteristics of the binary printed text image.



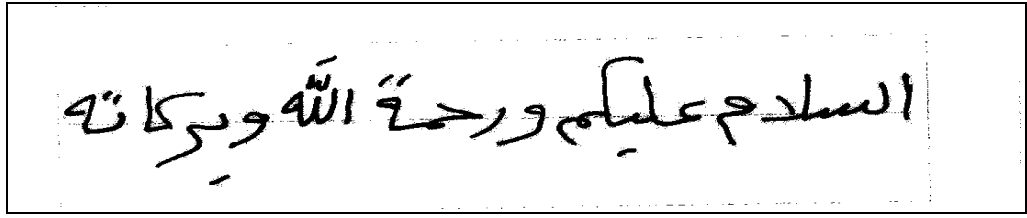
**Figure 18:** Binary image based on a Global threshold, a. Printed text image (for comparison), b. original handwriting text image, c. Binary image based on Global method

### **Local (Adaptive) threshold:**

Instead of having a single global threshold, this is based on selecting a different threshold value for each local area in an image. The local algorithm is based on calculating a threshold value for a small area or region of an image. Therefore, many different thresholds for many regions will be created in the same image (Gonzalez, et al., 2009).

Figure 19 shows a binary image based on local (mean) threshold. Unlike the output image from the above global thresholding procedure, the local thresholding based binarised image contains some artefacts in different locations throughout the image. This is due to the fact that the presence of a different level of dirt/shadows in different locations affects the local distribution of pixel values accordingly.





**Figure 19:** Binary image based on Local method

Our experiments confirmed the above observation and showed that the global threshold produces a more accurate clean image while the local method produces lots of noises which may need extensive processes for de-noising

### **3.4.1.2 Developing the de-noising procedure**

As mentioned earlier, the presence of image noise and artefacts have adverse effects on the segmentation process and added artefacts and undefined patterns resulting in the faulty classification process and low accuracy. Image de-noising is an extensively researched area of image processing, and several filtering techniques have been developed for the spatial domain as well as the frequency domain. For binary images, morphological operations are also used for de-noising and the removal of image artefacts. The use of morphological operations in our work for de-noising and artefact removal stems from the fact that we are dealing with binary images. For the same reason, we shall not consider the use of frequency domain de-noising schemes. Therefore, we shall examine a number of commonly used spatial de-noising filters like the median and the Gaussian filters, and morphological operations like, erosion, dilation, thinning, etc. (Gonzalez, et al., 2009). Note that, the spatial domain filters can be used for de-noising prior to the binarisation procedure. However, we have observed in the above section that binarisation with local thresholds introduces artefacts. Hence, in the following we shall test the use of filtering on locally binarised text images.

To compare the performance of these different operations we examine their effect on text images selected from our in-house scanned DB. For this DB, writers were asked to write texts in Arabic on white sheets of paper, and they were given the freedom to use different kinds of pens. These pages were scanned by a digital scanner of 150 dpi (dot per inch) resolutions. As a result of this scanning, different types of noises were produced everywhere around the text.

### Median filter (with local binary)

The median filter replaces the central pixel value of the filter window by the median value of the all the pixels in the window. Figure 20 below shows the output result of applying local binary image then applying a median filter of size 3x3. Some noises can still be noticed in the resulting image.

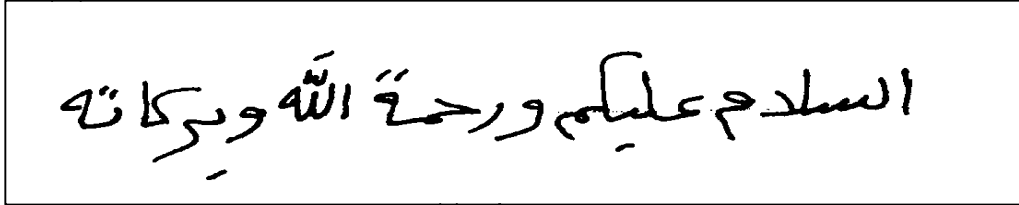


Figure 20: Image filtering by applying median filter, the input image was Figure 19

### Gaussian filter (with local binary)

This filter replaces the central pixel value of the filter window by output of taking the inner product of the all the pixels in the window with the following matrix:

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Figure 21 below shows the output result of applying local binary image followed by Gaussian filter. Some noises can still be noticed in the resulting image, but less than the noises produced when using median filter

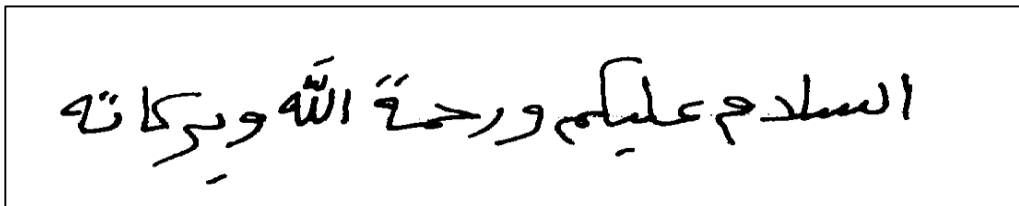


Figure 21: Image filtering by applying Gaussian\_filter\_5\_1, the input image was Figure 19

### Morphological clean filter (with local binary)

The resulting image from the previous section of the binarisation process contains isolated noise which can be confused mainly with the text diacritics. One de-noising way which is used with binary images that help to reduce the noise is applying Morphological filters. Morphological filters mainly used for thinning (Erosion) or thickening (Dilation). These filters are presented in Figure 22 Figure 23, and Figure 24 shown below:

Figure 22 below shows the output result of applying local binary image followed by the Morphological clean filter. More noises can be noticed in the resulting image.

Morphological clean removes isolated pixels (individual 1's that are surrounded by 0's), such as the centre pixel in this pattern.

0	0	0
0	1	0
0	0	0

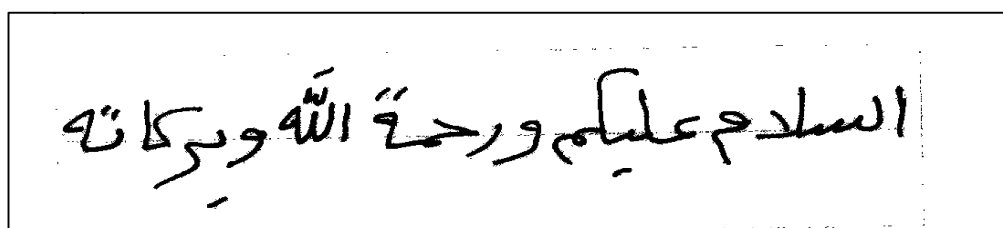


Figure 22: image cleaning by applying Morphological clean, the input image was Figure 19

#### **Morphological dilation filter (with local binary)**

Figure 23 below shows the output result of applying local binary image followed by the Morphological Dilation. A considerably more noises can be noticed in the resulting image.

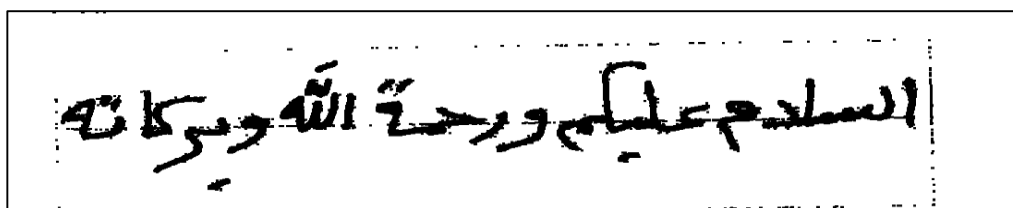
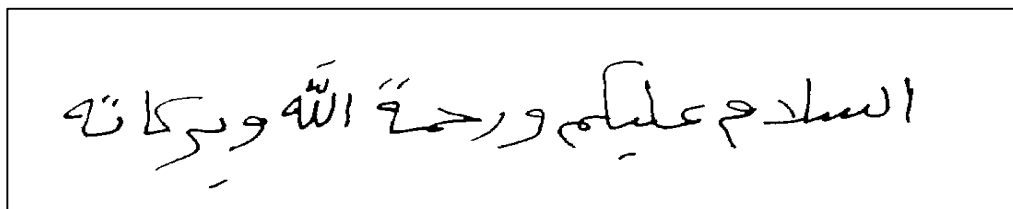


Figure 23: image cleaning by applying Morphological Dilation, the input image was

#### **Morphological erosion filter (with local binary)**

Even after applying different de-noising filters or morphological operations, isolated pixels can still be noticed out in and around the image, as shown in Figure 20 to Figure 23. Such pixels will have an effect on the segmentation accuracy as they still present noises and not regarded as text objects. This faulty filtering is due to the input image being binarised using local thresholding. However, morphological Erosion may produce

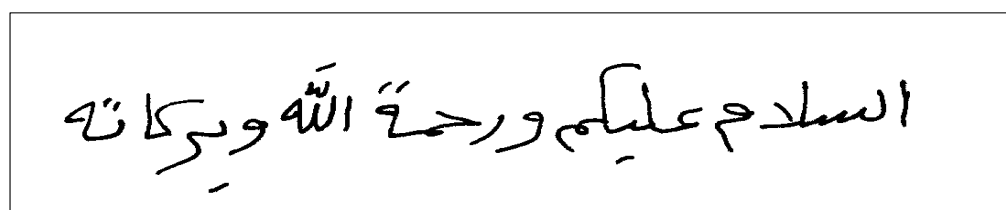
a relatively cleaner image but at the expense of losing some writing attributes. , as shown below in Figure 24. This due to the fact that this operation erodes away the boundaries of areas of the foreground pixels causing the pixels to shrink in size, and holes within those areas become wider, as a result the text image will be modified and will in turn corrupt the features.



**Figure 24:** image cleaning by applying Morphological Erosion, the input image

### 3.4.1.3 Suggested Methods: Global Binarising and Morphological Clean

We, therefore, came to the conclusion that using both global thresholding (binarising) followed by the morphological clean will produce higher accuracy than using local binarisation on its own, as clearly demonstrated in Figure 25 below



**Figure 25:** Image cleaning by applying Morphological clean, the input image was Figure 18c (Binary image based on Global method)

### 3.4.2 Text Orientation (Skew) Problem

Most handwritten texts are exhibits a certain orientation away from the horizontal text baseline named “text skews” as shown below in Figure 26

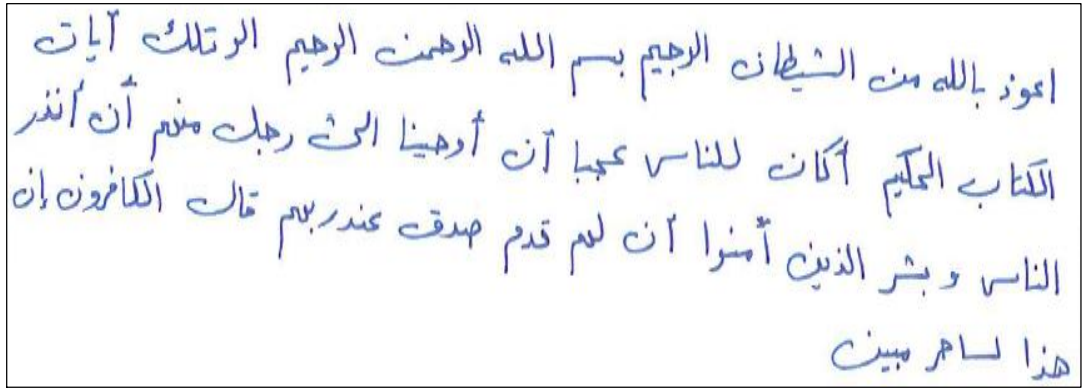


Figure 26: orientation Arabic paragraph

Skew detection is an important pre-processing task to the text image because it has a direct effect on the reliability and efficiency of the segmentation and feature extraction stages. The main approaches used to correct the skew are: projection, smearing, grouping, Hough-based, correlation, and graph-based, which are discussed in details in (Likforman-Sulem, et al., 2007) and (Bar-Yosef, et al., 2009). Here we give brief description of these approaches:

#### Projection profile:

The projection profile is calculated by summing up intensities from all pixels found at each scan line as shown in Figure 27. The corresponding profile is smoothed, and the produced valleys are identified. These valleys indicate the space between the lines of the text. (Manmatha & Rothfeder, 2005), (Bruzzone & Coffetti, 1999), and (Arivazhagan, et al., 2007).

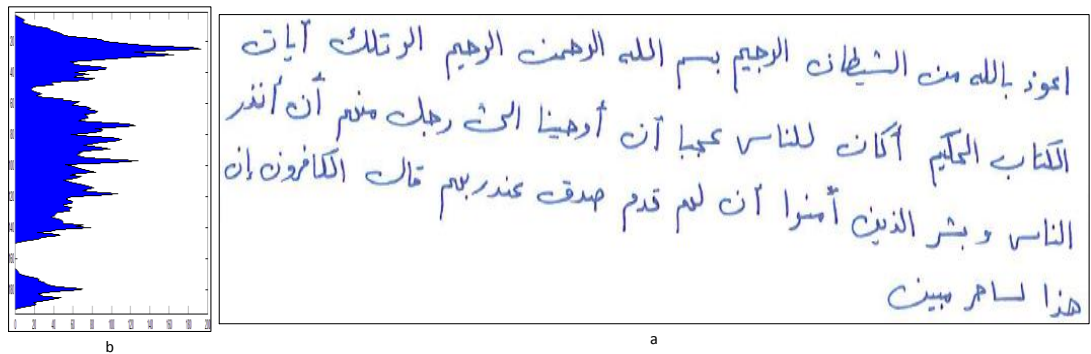


Figure 27: a. Orientation of an Arabic paragraph, b. horizontal projection

**Smearing methods:**

In this technique, sequential black pixels in the horizontal profile are smeared. If the distance between the white space is within a specific threshold, it is filled with black pixels. The bounding boxes of the connected components in the smeared image are considered as text lines. (Li, et al., 2006).

**Grouping:**

This method builds alignments lines by grouping units. Units may be pixels, connected components, or blocks. Then these units join together to extract alignments lines. (Likforman-Sulem & Faure, 1994), and (Feldbach & Tönnies, 2001)

**Hough transforms:**

Hough transform is also used for skew detection. The points in the Cartesian coordinate system are described as a summation of sinusoidal distribution as shown in equation (3.2):

$$p = x\cos\theta + y\sin\theta \quad (3.2)$$

The skew angle is calculated on the basis that at the skew angle the density of transform spaces is maximum. After mapping  $(x, y)$  into  $(p, \theta)$ , the count of points where a sinusoidal curve intersects another sinusoidal curve with a different  $(p, \theta)$  value increases the probability that a line determining the skew angle (Fletcher & Kasturi, 1988) and (Likforman-Sulem, et al., 1995).

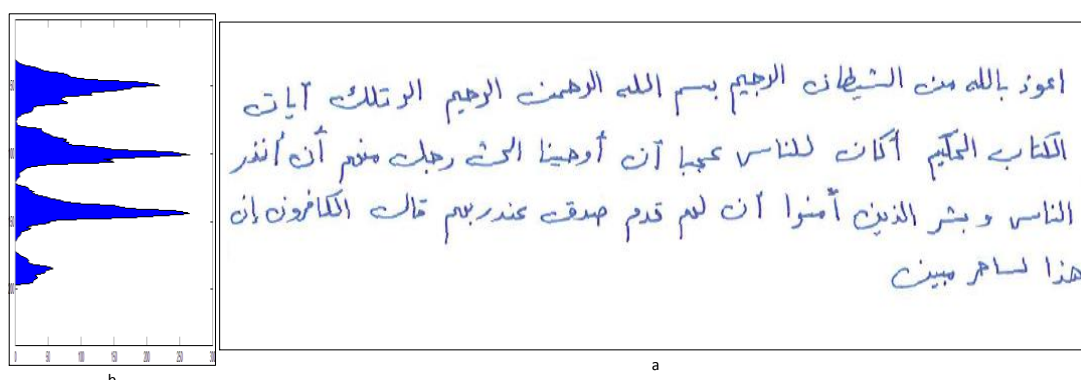
**Graph-based approach:**

This method consists of building a graph of main strokes of the document image and searching for the shortest space in this graph. This method assumes that the space distance between the words in a text line is less than the distance between two adjacent text lines. (Kumar, et al., 2006).

We propose enhancing projection algorithm by estimating the skew rotation angle. In order to estimate the right angle of the text page a multi rotation is done by the proposed algorithm below:

1. The multi rotating procedure which was applied by our publication in (Maliki, et al., 2012) to find the maximal Horizontal Projection Peak (HPP) which defines the skew of the page.
2. Find the horizontal projection at:
  - a) The current page with zero rotation.
  - b) The one degree rotated subword in a clockwise direction.
  - c) The one degree rotated subword in the anti-clockwise direction.
3. From the three points above, if the highest of the HPP is (a) then no extra rotation is needed. If the highest of the HPP is in the clockwise direction (b) then a clockwise increase of three degrees will be performed four times, the degree related to the highest of the HPP will be considered as the skew degree of that page. And the same will apply to (c) but in an anti-clockwise direction.

The results prove that the above-proposed algorithm (using our DB) for estimating and correcting the text skew were successful and gives good lines separations based on the projection, see the results in Figure 27 compared with Figure 28.



**Figure 28:** rotated (Figure 26) text, a. rotated text, b. horizontal histogram of (a)

Another challenge is to keep text attributes untouched while fixing these problems or at least recover them back after fixing. Line overlapping makes this challenge even harder to solve.

Therefore, in the next section we will be discussing line segmentation and the recovering of the original angle that comes with unprocessed text to keep the most important feature which is the slope of the text.

### 3.5 Text Segmentation

Having pre-processed the binarised scanned text images and corrected the text skew, we are now in a position to discuss the text segmentation needed to achieve our objective of

WI for Arabic handwritten text. This process works in a number of steps, starting with the segmentation of a text page into its separate lines and segmenting each into its connected components ending with segmenting subwords. However, in each step we may need to improve the outcome by recovering features that may get lost through each step.

### 3.5.1 Line Segmentation

Line segmentation aims to extract text lines from pages or paragraph(s). Page and text lines have individual characteristics (features), but, these features are of no importance to us as we shall focus on the smaller patterns which are subwords and their diacritics. However, some line attributes help preserve subwords and diacritics features intact. In particular, the slope feature may become corrupted as a result of the process of de-skewing the page for line segmentation. Therefore, we suggest re rotating the lines of the image text based on the original page angle.

Segmenting pages into lines then into words, and/or into characters is widely used by OCR and WI researchers. We followed the same strategy they proposed while developing the proposed algorithm which will help in identifying the writer. After de-skewing the text as discussed in the previous section, we search to find the minima points in the horizontal projection histogram in de-skew page text to be regarded as line segment points, i.e. scan the histograms at each and every image rows to identify the minima points (valleys points). Minima point is a valley point in a horizontal projection text image to be used as segmentation points and provides automation of the process. These valleys indicate the space between the lines of the text to regard as segmented line points as shown in Figure 29

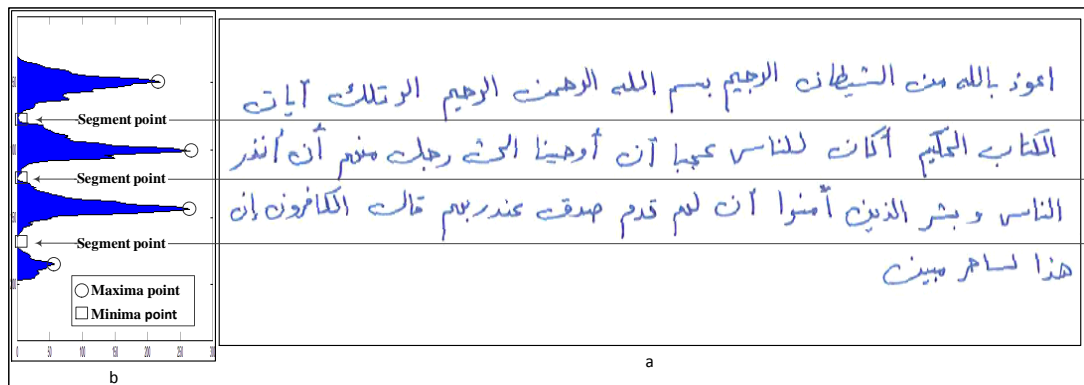


Figure 29: Line segmentation



As result of using projection technique, text image is segmented into separate lines as illustrates in Figure 30

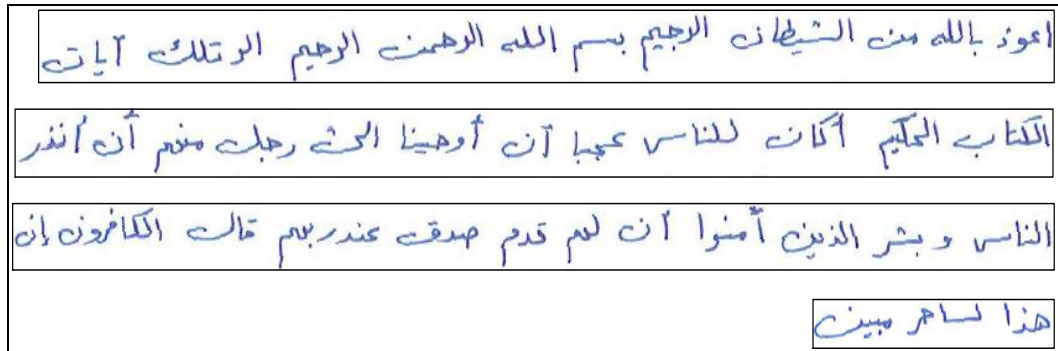


Figure 30: line segmentation result

### Recovering lost features

The result of de-skewing the text and then segmenting it into lines may corrupt the slope features of each line which in turn leads to corrupting subwords' and diacritics' slope features.

Figure 31a shows the text line slope features (original slope) before segmentation while Figure 31b shows the text line slope (but corrupted slope feature) after line segmentation.

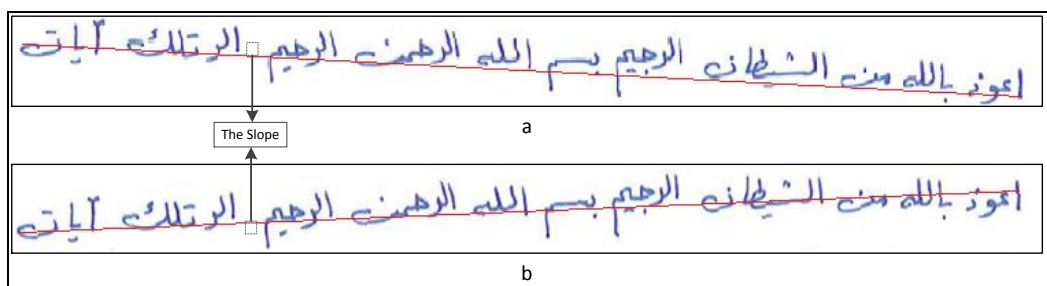


Figure 31: lost slope features demonstrates, a text line (taken from a text paragraph) before and after segmentation

To overcome this problem, the saved line image we re-rotated the image by an inverse angle to revert it to the original handwriting text line. Sample of the result is shown below in Figure 32



**Figure 32:** re-rotate lines (Feature's recovery)

In summary, line segmentation works by the following 3-step procedure:

1. De-skew the text image using the maximum HPP procedure describe above in section 3.4.2
2. Construct the line projection histogram of the image, and use the rows corresponding to the minimal points of the histogram values as line separators and split the text page into the different text lines.
3. Restore the original line slope by reversing the operation in step 1 to recover the lost slope features.

### 3.5.2 Segmenting Lines into Words, Subwords and Diacritics

Two major approaches are used to segment Arabic printed and handwritten lines into words, subwords and other patterns. Again these approaches are based on histogram projection analysis and labelling/re-grouping the connected components (Amin, 1998), (Lorigo & Govindaraju, 2006), and (AlKhateeb, et al., 2009). However, in this case, the projection histograms are determined by the number of pixels vertically along each text line image columns. In fact, the interest is in detecting vertical gaps to correspond to successive columns that have no text pixels.

#### Words and subwords segmentation

As mentioned before in chapter 1, words in Arabic and in similar script languages are constructed from different types of characters, some of these characters can be connected with the previous and the following character and others can be connected to the previous letter only or be completely disconnected. This particular structure will mean that a word

can contain multiple subwords, which are separated by small gaps (Lorigo & Govindaraju, 2006), as shown in Figure 33.

Based on these spaces/gaps words and subwords can be segmented if the text is written perfectly (no-overlap), like the case of printed text as shown is Figure 33 and Figure 34 or some handwritings that are well written.



Figure 33: identify words and subwords gaps

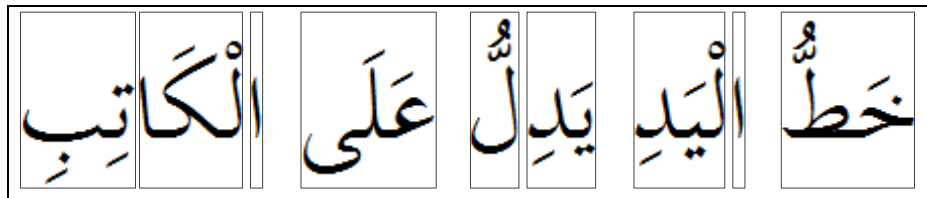


Figure 34: words and subwords segmentation

Such approach was implemented by Berkani et al. (Berkani & Hammami, 2002), as shown in Figure 35

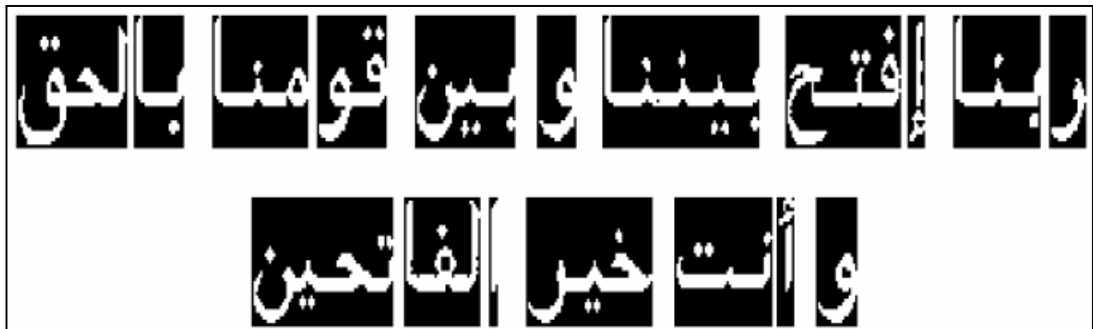


Figure 35: Sub -words vertical segmentation (Berkani & Hammami, 2002)

These examples show that the gaps that separate subwords within a word are significantly smaller than those separating complete words. This is normally expected to be the case for handwritten text, but different writers may have different habits in the size of gaps they leave in both cases.

However, most words, subwords, characters, and diacritics (even between text lines) in the handwritten text are usually overlapped especially with cursive languages. These overlapping problems are found clearly in Arabic and in similar scripts languages. This problem renders writer attributes extraction difficult as it may produce incorrect patterns for some if not all writers. Figure 36c shows a sample of overlapping between words and subwords. The encircled areas are faulty segmentation because the subwords are overlapping. The overlapping problem will be discussed in the next section.

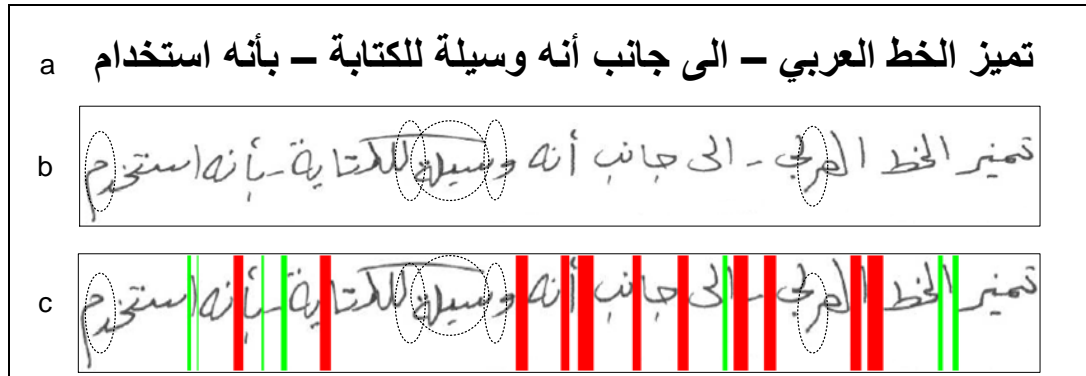


Figure 36: a. Arabic printed sentence, b. handwritten text, and c. segmented word and subwords.

### 3.5.3 Enhancing Segmentation Algorithm

The overlapping problem means that some of the horizontal gaps between subwords/words are not detected. Here we propose the Enhanced Histogram Analyses Approach (EHAA) algorithm for segmenting line text into words and Subwords which is meant to solve the overlapping problem.

Such enhancement process is meant to solve the overlapping problem.

The proposed algorithm is shown below:

- Align text line image by:
  - Determining the gaps correctly found between words and/or subwords in the original line/text image without repositioning them.
  - Determine the baseline of each and every segmented text. See the indicated short red lines in Figure 37. Note that in this example not all short red lines are aligned along the same horizontal baseline.

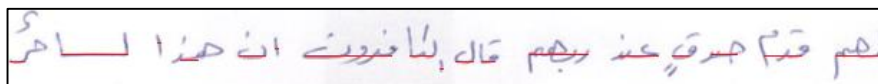
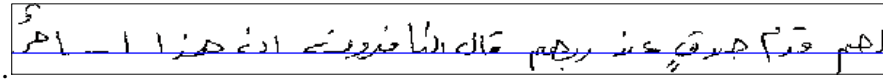
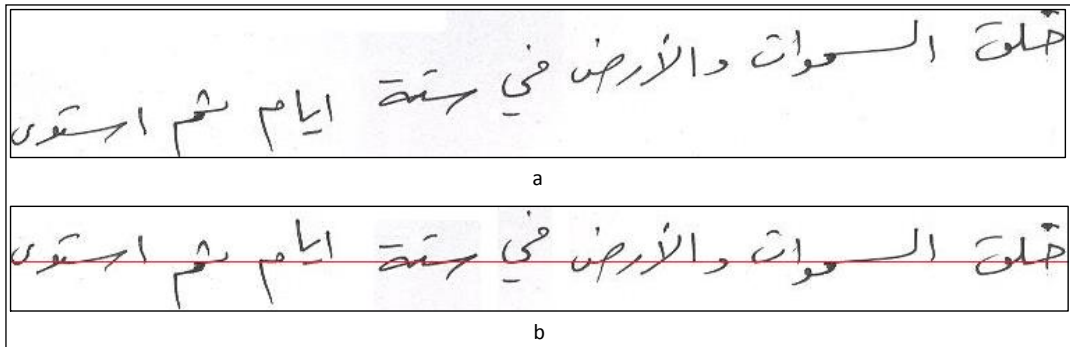


Figure 37: Identifying baseline of each and every segment

Re-position each and every segment by placing it according to its own baseline on the estimated baseline of the entire text, as shown in Figure 38 and



**Figure 38:** Re-positioning every segment on estimated baseline



**Figure 39:** Another example of re-positioning every segment on estimated baseline. a. original text line, b. re-positioning segments

- Normalizing these gaps (between 0 to 1) based on the following equation:

$$space\ Normalizing = \frac{\max\ space\ value - segment\ space\ vlaue}{\max\ space\ value - min\ space\ value} \quad (3.3)$$

- Select a threshold to be used to distinguish words' gap separators from subwords' gap separators. Where Norm (gap) equal to or above (3.3) indicates word's gap (Figure 36 indicated by red colour) while gaps below that value indicate the presence of a subword (Figure 36 indicated by green colour). This Segmentation is based on a vertical projection of the entire line zone which results in determining the gaps between words or subwords.

Unfortunately, this procedure may not succeed in detecting subwords and will require additional enhancement before getting a reliable subwords segmentation. To illustrate this, consider the results of the above segmentation, before applying the further

enhancement. We observe 7 errors in total, 2 of which are words' gaps and 5 are subwords' gaps. See Figure 40

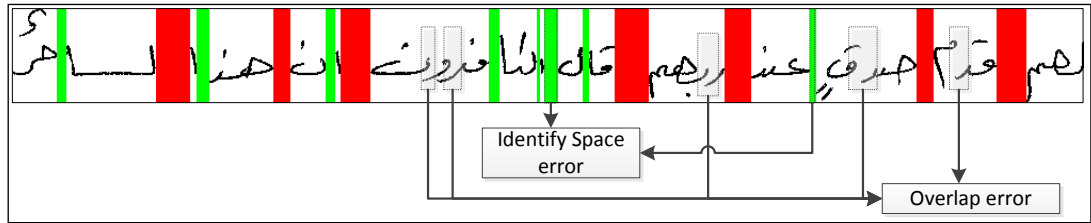


Figure 40: segmentation results before applying further enhancement

We then attempted to enhance the segmentation process by using a **partial vertical projection** that starts from the baseline zone and continues either upwards or downwards. This gave us higher segmentation accuracy, as shown in our experiment in Figure 40. Here, we have used two different partial vertical projection thresholds to distinguish between words and subwords.

When we applied vertical space threshold=30, 6 errors in total have been found. And when determining words' gaps 2 errors had occurred, while when determining subwords' gaps 4 errors occurred. As shown in Figure 41

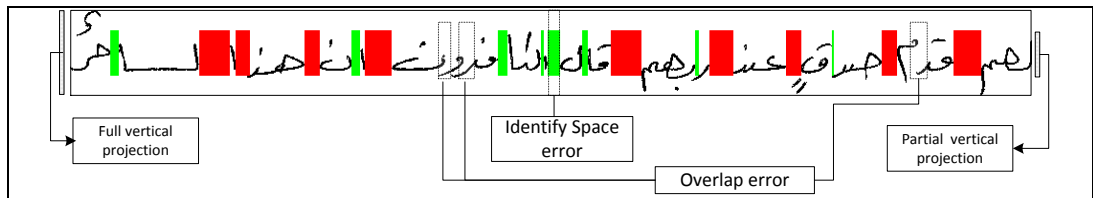


Figure 41: segmentation after enhancement based on vertical projection threshold (=20)

When we applied threshold=25, five errors were found. And when determining words' gaps 3 errors had occurred, while when determining subwords' gaps 2 errors occurred. As shown in Figure 42

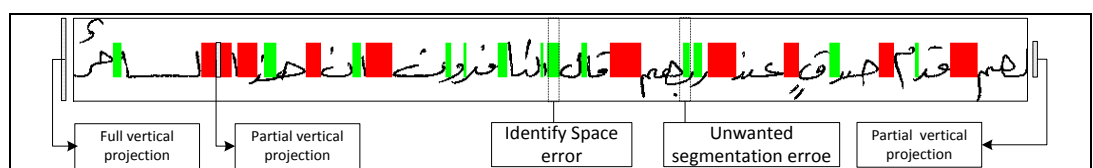


Figure 42: segmentation after enhancement based on vertical projection threshold (=25)

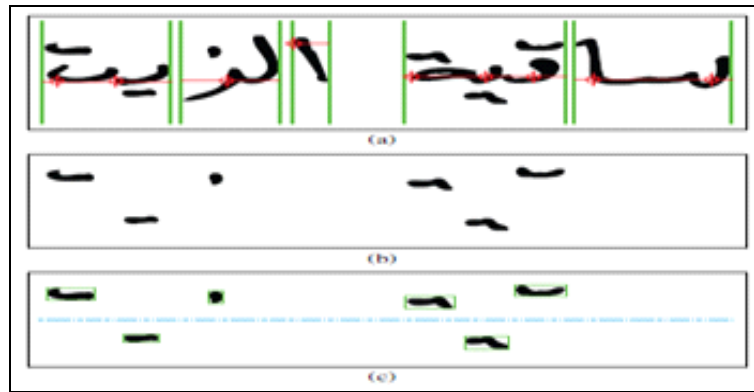
We conclude from the above experiment that applying the enhancement method (EHAA) described above can reduce the rejected segmentation errors, but it may not be enough.

However, while rotating segmented parts to remove slope and slant will lead to increased segmentation accuracy, it will corrupt important attributes which form an important part of writer's habit/styles dependent features. Therefore segmenting text lines without rotation is more desirable and could lead to an improved writer identification accuracy. Hence, we went ahead to investigate further other segmentation methods with the aim to segment the text in a better way while preserving the attributes that reflect writer's habits and styles. Our preference would be to use a Labelled Connected Components (LCC) approach to segment text lines into their components. However, for Arabic writing this approach would detect words/subwords without their diacritics. Accordingly we first discuss existing work on diacritics segmentation first before proposing our proposed LCC based segmentation.

### **Diacritics segmentation**

To the best of our knowledge, very few researchers have conducted WI experiments based on diacritics only. The only work that uses diacritics for the purpose of WI that I am aware of is by Lutf et. al (Lutf, et al., 2010).

Lutf et. al segmented the diacritics by removing the main text from the input image while keeping the diacritics untouched as shown in Figure 43. The way they had done this is by removing all the pixels found on the baseline as well as the pixels that are in their connected components (i.e. pixels that can be reached from the baseline through a finite sequence of neighbouring linked text pixels). This shows that their approach is also LCC based.



**Figure 43:** Diacritics segmentation, (a) locating of the start points, (b) after clearing the text, (c) final diacritics segmentation. (Lutf, et al., 2010)

Then they used the vertical and horizontal projection profile to extract the diacritics. Depending on their position they had categorized diacritics as either above or below the estimated baseline.

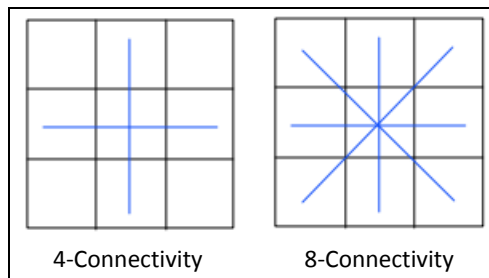
### 3.5.4 The LCC Based Text Segmentation

Segmenting Arabic text into subwords or diacritics using projection information has been shown above to faces difficulties that mostly arise as a result of text overlapping. Hence, we propose to follow a different segmentation strategy based on using Active Contour model (also known as Labelling Connected Components LCC), which is used for many automated text analysis applications Such as writer identification.

LCC algorithm scans a binary image pixel by pixel from top to bottom and left to right, and then clusters them into patterns based on pixel connectivity. Every cluster (group) is given a unique label. The values of the labels are positive integers; start with 0 (as image background). Pixels labelled 1 make up one object (first pattern); those labelled 2 make up a second object (second pattern); and so on.

LCC works on binary or grey images checking the first pixel that it encounters and all neighbouring pixels in a window around it that is also connected to it. There are different bases for considering connectivity, the most common ones are based on 4, or 8 connectivity, see Figure 44 which shows the connectivity patterns below:





**Figure 44:** connectivity diagrams (4 and 8-neighbor)

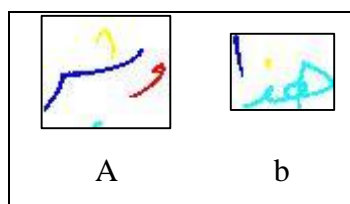
Our system works on binary images and 8-connectivity and scans the text from right to left as written in Arabic.

The system, in this case, will associate (0) to background pixels and (1) to a pixel in a text pattern. If all eight neighbours are (0), the system will keep scanning until it encounters a (1). At this point, it will assign a new label to a new pattern.

Once it encounters a (0), it decides the end of the pattern and so on until it reaches the end of the text image altogether.

### **What benefits we gain from applying LCC on our investigations**

1. With LCC, we solve the problem of text overlapping because the segmentation process now will recognise patterns as labels only regardless of its position whether it is above or below any other label. See Figure 45 where (a) shows 4 different patterns extracted from one word and (b) shows 3 different patterns produced by the second word



**Figure 45:** Segmentation based on LCC. a. first word, b. second word

2. With LCC, also, we will not be concerned with text orientation as the algorithm will base its analysis on the connectivity of pixels resulting in a pattern rather than its position on a text baseline. However, if we need to re-connect diacritics to its original subwords (as for our experiments in chapter 4), we will have to resort to baseline estimation (algorithm of this process is shown in Figure 47).
3. The most important benefit of all is the fact that we do not lose any of the features that are vital to our investigation like the pattern slope of the writer.

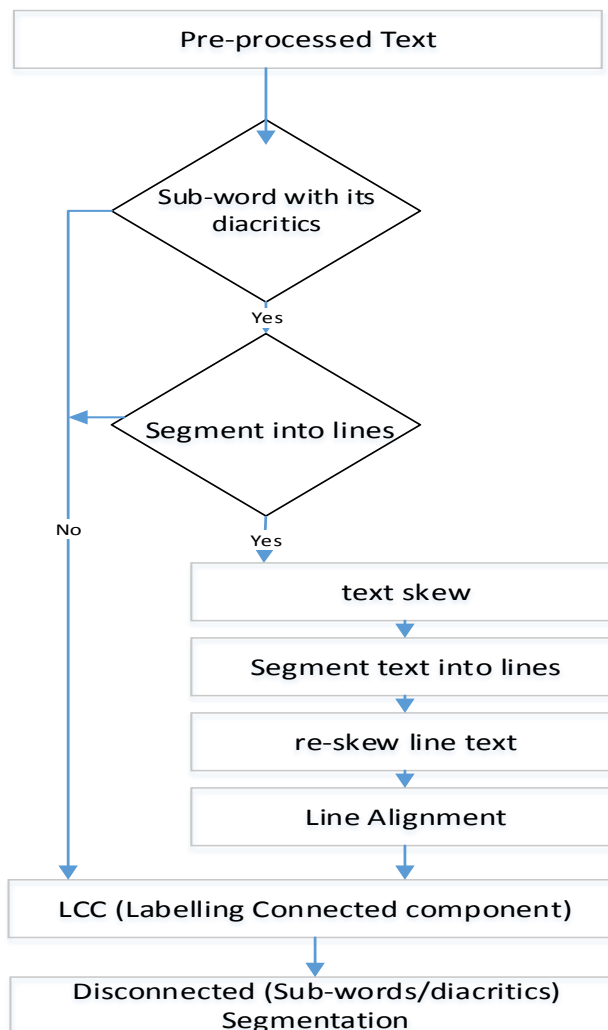
Results obtained when applying LCC shown in Figure 46



**Figure 46:** Text segmentation based on LCC algorithm

Using LCC will return patterns (as labels). At this stage, the system will classify these patterns into two types: subword bodies (without diacritics) and diacritics only. This classification process is based on the pattern's size, where smaller patterns will be regarded as diacritics otherwise they are subword bodies. This will make it possible to conduct identification experiments using either diacritic on their own, subword bodies without diacritics, or subwords with their diacritics.

The flowchart diagram Figure 47 below summarizes the entire pre-processing and segmentation scheme that will be adopted in the rest of the thesis. Note that this algorithm implicitly determines page orientation, line segmentation, baseline estimation, and finally the application of LCC.



**Figure 47:** Proposed Segmentation system based on LCC

To the best of my knowledge, there are no existing methods for checking the accuracy of extracting subwords and diacritics from a text, especially in Arabic handwritten texts.

Here, we have developed a **Graphic User Interface (GUI)** system, in Matlab, to randomly examine 200 text lines that consist of (as an average) 1400 subwords from IFN/ENIT DB, and another 200 text lines that consist of (as an average) 3800 subwords from our in-house DB. We found out that the above LCC method produced higher accuracy results in both DBs than the EHAA projection method as illustrated in Table 3.8.

**Table 3.8:** comparison analysis between result of segmentation based on EHA and LCC

Methods	Accuracy	
	In-Home DB (Buckingham)	IFN/ENIT DB
EHAA	76.191 %	76.398 %
LCC	98.45%	98.96 %

We notice from Table 3.8 above that the error rate when applying LCC is less than 1.5%, this is due to some undesired connectivity between patterns.

### **3.6 Summary of the Chapter**

In this chapter, we investigated various approaches to segment Arabic handwritten texts into patterns. We first developed a pre-processing scheme to extract a binary image from the scanned images of handwritten paragraph/page text in Arabic which successfully yields a visibly clear binary image from which noise and other scanner-caused artefacts were removed. Having identified the main cause of text overlapping, we developed a horizontal projection frequency based scheme to reduce the effect of this problem by correcting text orientation. The success of this horizontal projection scheme was further utilized to segment successfully the lines of text in the scanned documents. This has motivated our next investigation into using vertical projection to segment the text lines into its words subwords and diacritics component. Although the algorithm was successful in most cases, but we found some errors, that are mainly due to sever text overlapping cases. We finally used the active shape LCC approach to segment all the required text components and demonstrated its superior performance compared to the projection-based algorithm. We have shown that this procedure increases segmentation accuracy significantly to more than 98%, and most importantly the LCC algorithm maintains the features that are most important to our investigation like pattern slope.

### **3.7 Next Step**

In the next chapter, we shall use the segmented subwords with diacritics to begin our work to test our hypothesis that subwords are suitable for WI from Arabic handwritten text.

We will conduct experiments to test the performance of the developed scheme on a certain benchmark text-dependent DB. Our experiments also aim to identify the most essential features and a group of subwords that are sufficient for achieving high accuracy on an in text-dependent DB.

## **Chapter 4 : Subword based Arabic Handwriting Analysis for WI**

The earlier part of last chapter was devoted to make arguments that subwords are the text components that encapsulate writer handwriting habits and styles. This argument was mainly based on the fact that subwords can be repeatedly found in a variety of text configurations as part of a specific group of words of different meanings or even as complete words. A very understandable, but yet to be substantiated, is that these observations lead to believe that automatic WI systems have a better chance of success when based on subwords than on words. Accordingly, our main hypothesis in this chapter and throughout the thesis dictates that subwords are the most suitable text components could be exploited for WI from handwritten Arabic text

Therefore, we aim to investigate and test the credibility of our hypothesis by developing subwords based WI from Arabic handwritten text and empirically demonstrating that such schemes will achieve higher accuracy than those WI schemes that use complete words as done by other researchers. The performance of our proposed subwords WI scheme(s) will be tested on a publically available text-text-dependent database that is used WI from 27 Handwritten Arabic words, derived from 16 commonly used phrases/sentences. In total, there would be 49 subwords that can be extracted from the 27 words.

The work in this chapter builds on the pre-processing procedures and the LCC texts segmentation scheme designed and tested in last chapter. In section 4.2, we shall describe 15 relevant digital features that could represent the discriminating power of subwords, we shall develop the corresponding feature extraction scheme to be used for writer identification. In section 4.3 we develop and refine our proposed subwords based WI scheme. We first conduct initial pilot experiments to select the most performing features and an incrementally designed WI scheme that uses the most writers discriminating feature sub-vector of the original 15-dimensional feature vectors. We shall then conduct extensive experiments to test the performance of the designed scheme, aiming at achieving the best performance with the smallest number of the most writer discriminating subwords. The results will be compared with the performance of an existing word based WI scheme. Finally, in section 4.4, we shall focus on the nature of

the final set of features. Two of them are projections (in the form of time-series) whereas the others are single valued real numbers and the initial pilot experiment has indicated that these do not contribute significantly to the performance of our WI scheme. We shall consider the use of compressive sensing approach for dimension reduction to replace the original 13 features with a smaller number of meta-features formed from linear combinations of all the 13 features as an alternative to feature selection.

## 4.1 Introduction

The handwriting style is learnt and mastered over time, and people usually develop habits that influence their style of writing. These habits are discernible from and embedded in, certain parts of their handwritten texts.

As mentioned earlier in chapter two, writer identification is trendily based on analysing paragraph(s), line(s), word(s), character(s), and/or a part of a character, but more often is based on words and characters. Past Arabic texts WI researches used entire words in their analyses rather than parts of a word. In fact, this follows the trend in WI research in many other languages. Table 4.1 displays different works using a variety of languages were presented. All of these works have adopted a word-based approach and tests carried out on text-dependent DBs only while using subwords and in text-dependent DBs in WI is not very common.

**Table 4.1:** Works that are based on word characteristic using text-dependent databases in different languages

no	Paper	# Word	# Copy	# Writer	Total words	Language
1.	Zois et al. (1999) (Zois & Anastassopoulos, 2000)	1	45	50	2250	English/Greek
2.	Tomai et al (2002) (Tomai, et al., 2004)	25	3	1000	75000	English
3.	Zou et al. (2002) (Zuo, et al., 2002)	40	10	40	16000	Chinese
4.	Zhang et al (2003) (Zhang & Srihari, 2003)	4	3	1027	12324	English
5.	Al-Ma'adeed et al(2008) (Al-Ma'adeed, et al., 2008)	27	20	100	54000	Arabic
6.	Wu et al (Wu, et al., 2014)	Six benchmark DBs				English, Germany, French,

no	Paper	# Word	# Copy	# Writer	Total words	Language
						Greek, and Chines
7.	Aboul-Ela et al. (Aboul-Ela, et al., 2015)	31	5	50	165850	Arabic and English

In chapter 2 we pointed out that words in Arabic writing consists of one or more subwords and at the end of each subword there is a special ending we called it “connect stroke”. In handwritten texts, these strokes constitute feature, unique only to a specific writer which we will detect and utilize for the purpose of identifying the writer of a certain text. However, in this thesis we will not explicitly use a representation of such feature, but some of our extracted attributes will implicitly be influenced by strokes (see section 4.2.1, below). Furthermore, subwords can be repeatedly found in a variety of text configurations as part of a specific group of words of different meanings or even as complete words.

Our aim is to develop a well performing WI system from Arabic handwritten text that will use subwords’ attributes. Initially, we test the performance of our scheme on a well-known benchmark text-dependent DB to extract features and subwords and then to move forward to use intext-dependent DBs to test investigated texts accordingly. Text-dependent DB is also used to compare all the results we obtained with their results.

In this chapter, many experiments conducted to demonstrate the validity of our hypothesis, we will consider the entire subword in question as an atomic unit (i.e. body of subword together with its diacritics) as a tool to identify the writer’s habits. However, in chapter 5 we shall investigate the splitting of this unit.

The main question of the entire hypothesis is; **can subwords in an Arabic text be a writer discriminating factor to identify reliably the writer of that text better than using whole words?**

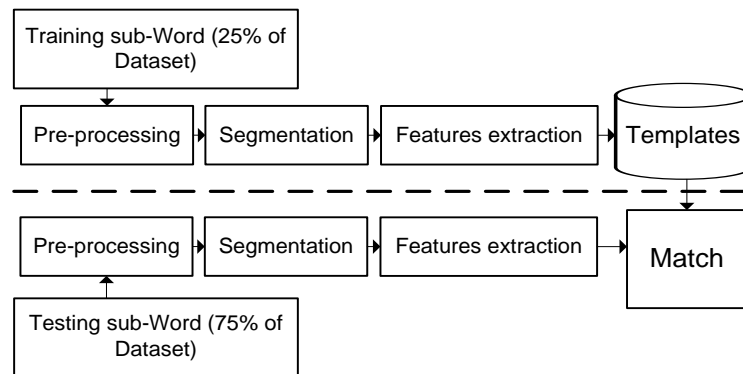
To substantiate such a profound claim, we chose a DB, which was tested before to identify writers based on entire word’s features. This DB is a text-dependent database, which contains some of the most commonly used Arabic words there is which was collected by Al-Ma'adeed et al (Al-Ma'adeed, et al., 2008) This DB was made of a group of 27 specific words written by 100 writers who were asked to write repeatedly these words for about 20 times.

Below is a list of some examples of these words:

(على، عن، بخير، في، هي، هو، لك، المحترم، التوقيع، من، ولكم، جزيل، الشكر)

One of the reasons behind this choice is to enable a distinct comparison between the two different methods (subwords-based identification as opposed to word-based identification). Another reason is to extract the best group of features and the best group of subwords that will be incorporated the in text-dependent DB, which have gathered to help us in mapping writer's habits and styles.

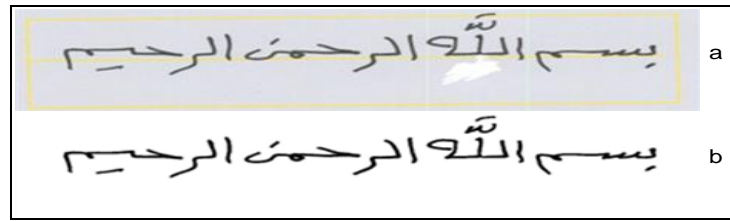
In order to develop our intended WI scheme, we follow the traditional approach used in designing and testing pattern/biometric recognition schemes. Such an approach consists of multiple steps the initial part of which covers the preparation step that begins with the row signal and ends with producing what is known as feature vector representation, also known as template, of the object of recognition. For any machine learning system, the initial step will be conducted during both the training stage as well as the testing step, although in the training stage many parameters are fine-tuned. The final step of validation deals with decision making that are based on comparisons using distance/similarity function to be used with an appropriately selected classifier. The specific steps of our WI scheme are pre-processing, segmentation, feature extraction, classification, as shown in Figure 48



**Figure 48:** WI scheme- Block Diagram

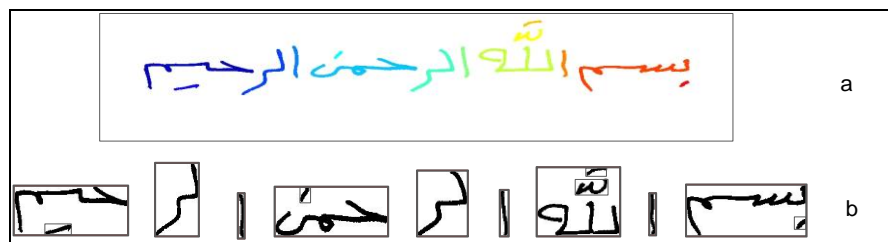
The pre-processing and segmentation stages have already been fully developed and presented in chapter 3. In order to remind the reader about the effect these various steps, we display below a scan of a short one-line text string in Arabic followed by the output from the various steps of preparation. Figure 49 shows a sample of the input and output of the pre-processing binarisation and de-noising procedures:





**Figure 49:** Pre-processing, a: Original image, b: Binarised and cleaned image

Figure 50, below, shows output from the LCCA segmentation algorithm, described in chapter 3, of the binarised and cleaned text in Figure 49. The top line shows the colour labelled subwords and their diacritics while second line encloses each connected component in a box ready for the next step of feature extraction.



**Figure 50:** Segmentation, a. LCCA image, b. re-attached subwords with their diacritics

## 4.2 Components of WI Scheme for Arabic Language.

A WI scheme is characterised by the structure of the feature vector representation of a handwritten text components/document, a measure to determine the level of similarity between two such vectors, and decision criteria that is based in some way on the similarity value to accept or reject the claimed identity.

Following the approach depicted in Figure 4.1, the most crucial element of any identification scheme is the feature selection and extraction step which aims to represent any Arabic handwritten text by a digital feature vector. How many and what type of features (i.e. coordinates) are to be used when trying to identify the writer of a given input handwritten text, are to be determined experimentally. In this section we shall first discuss the extraction of what we consider as a sufficiently large number of features from which we can select all or a subset of handwritten texts templates/representations to be used for testing the performance of any proposed WI scheme in order to determine the optimal feature vector that achieves the best WI accuracy rate. In the last part of this section, we shall describe a set of WI schemes each corresponding to the chosen feature vector representation of Arabic written text.

### 4.2.1 Handwritten Feature Extraction

The procedures developed in this section are meant to work on each of the text component boxes output from the preparation step. In general, there is a potential for a large number of digital features that can be extracted from the handwritten text that may tell us about writers style and may even have writer discriminating power. However, here we shall begin our investigations by listing three main types of features that we propose to extract from each subword box, namely: statistical, boundaries, and projections. For each text component we extract 15 different features: Slope, height, width, area, distance from word's baseline (upper and lower pixel distance), image moment (7 features), and image projections (horizontal and vertical). Although, we have mentioned that strokes (i.e. the shape of the end character in a subword) is a reliable indicator of writing style/habit that human experts use in identifying writers of Arabic text, the list below doesn't explicitly include this for our schemes. In fact, some of the features below are affected by some aspects of strokes shapes, and we believe that this implicit use of strokes features is possibly sufficient for our purpose.

#### 4.2.1.1 Statistical Features (Subword's Slope, Height, Width, Area)

The following features; Slope, Height, Width, and Area are considered to be very important in reflecting the writer's habits and style as mentioned in (Huber & Headrick, 1999).

The slope feature ( $f1$ ) is the angle of the shape which we proposed, in (Maliki, et al., (2012)), to be calculated after extracting regression equation, i.e. the best linear fit of the shape (Linear Regression Model) calculated by equations (4.1), (4.2), and (4.3) below.

$$\bar{y} = \text{slope} * x + \text{intercept} \quad (4.1)$$

Where:

$$\text{slope} = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \quad (4.2)$$

$$\text{intercept} = \bar{y} - \text{slope} . \bar{x} \quad (4.3)$$

Where:  $\bar{y}$  is the mean of y, and  $\bar{x}$  is the mean of x.

Figure 51 illustrates the Linear Regression Model in general while Figure 52 shows the slope of a subword.

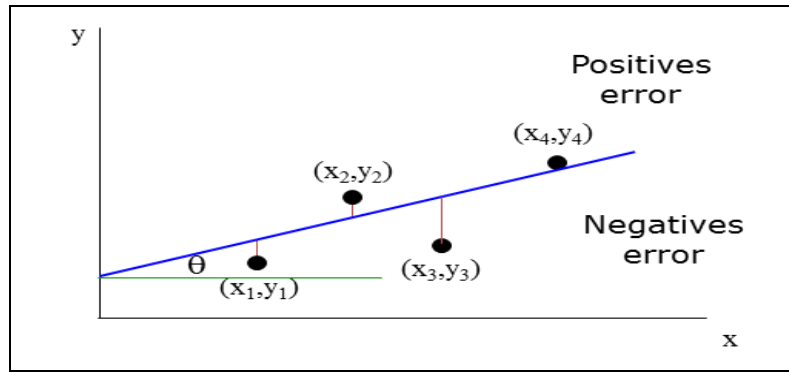


Figure 51: Illustration of the Linear Regression Model

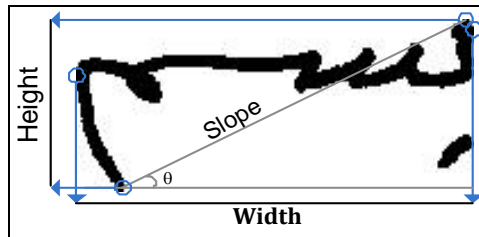


Figure 52: A. Subword slope, height, width, and area.

The Height and Width features ( $f_2$ ) and ( $f_3$ ) are separately calculated by finding the first and the last black (i.e. text) pixels in the image by projecting horizontally and vertically, as illustrated in Figure 52 above.

The area feature ( $f_4$ ) represents the size of the text and is calculated by counting the number of black pixels of a subword image.

#### 4.2.1.2 Boundary Features

The boundaries of text lines or sentences are made of two features. The first is the upper zone ( $f_5$ ) and the second is the lower zone ( $f_6$ ). The upper zone is calculated starting from the baseline upwards to the first detected pixel of the horizontal projection of the subword. While the lower zone is calculated starting from the baseline downwards to the last detected pixel of the horizontal projection of the subword, as shown in Figure 53 below.

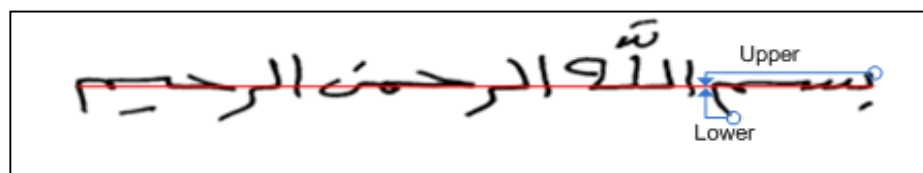


Figure 53: Subword distance from upper/lower phrase baseline

### 4.2.1.3 Image Moment (Invariant):

Image moment features (invariant moments (f7:f13)) are special weighted averages of the intensity of the image pixels. Image Moments may be used to describe objects after segmentation. For full details see (Flusser, 2000)

For a 2D continuous function  $f(x,y)$  the moment of order  $(p + q)$  is defined by equation (4.4)

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad (4.4)$$

For  $p,q = 0,1,2,\dots$  Adapting this to scalar image with pixel intensities  $I(x,y)$ , raw image moments  $M_{ij}$  are calculated by equation 4.5)

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \quad (4.5)$$

Based on Hu's theorem (HU, 1962) which states that if  $f(x,y)$  is piecewise continuous and has nonzero values only in a finite part of the  $xy$  level, moments of all orders exist, and the moment sequence  $(M_{pq})$  is uniquely firm by  $f(x,y)$ . On the other hand,  $(M_{pq})$  uniquely determines  $f(x,y)$ . Moreover, the image is summarized with functions of a few lower order moments

Hu employed the results of the theory of algebraic invariants and derived his seven famous invariants to the rotation of 2-D objects:

We suggest to find the intensity of the subword pixels using the 7 equations below which performed by (HU, 1962) [see equations (4.7) to (4.13)]. In each of these 7 equations, the value of  $\mu_{m,n}$  is calculated using equation (4.6), for  $n=0,\dots, 3$  and  $m=0,\dots,3$ .

$$\mu_{pq} = \sum_x^p \sum_y^q (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (4.6)$$

$$I_1 = n_{20} + n_{02} \quad (4.7)$$

$$I_2 = (n_{20} - n_{02})^2 + (2n_{11})^2 \quad (4.8)$$

$$I_3 = (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2 \quad (4.9)$$

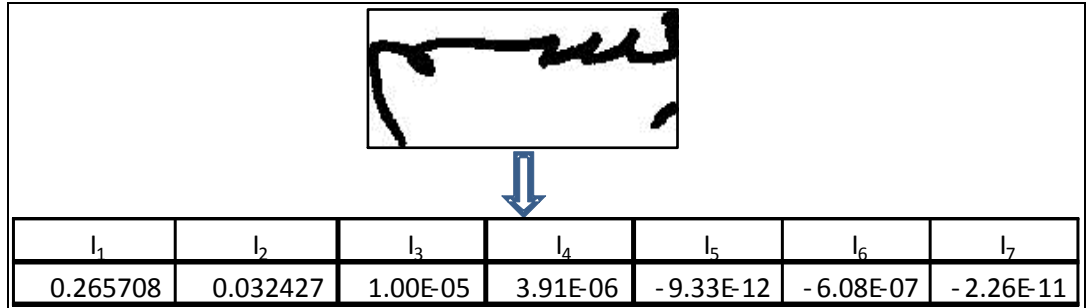
$$I_4 = (n_{30} - n_{12})^2 + (n_{21} - n_{03})^2 \quad (4.10)$$

$$I_5 = (n_{30} - 3n_{12})(n_{30} - n_{12})[(n_{30} - n_{12})^2 - 3(n_{21} - n_{03})^2] + (3n_{21} - n_{03})(n_{21} - n_{03})[3(n_{30} - n_{12})^2 - (n_{21} - n_{03})^2] \quad (4.11)$$

$$I_6 = (n_{20} - n_{02})[(n_{30} - n_{12})^2 - 3(n_{21} - n_{03})^2] + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}) \quad (4.12)$$

$$I_7 = (3n_{21} - n_{03})(n_{30} - n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} - n_{03})^2] - (n_{30} - 3n_{12})(n_{21} - n_{03})[3(n_{30} - n_{12})^2 - (n_{21} - n_{03})^2] \quad (4.13)$$

Figure 54, below, illustrates the invariant moments for the displayed subword.

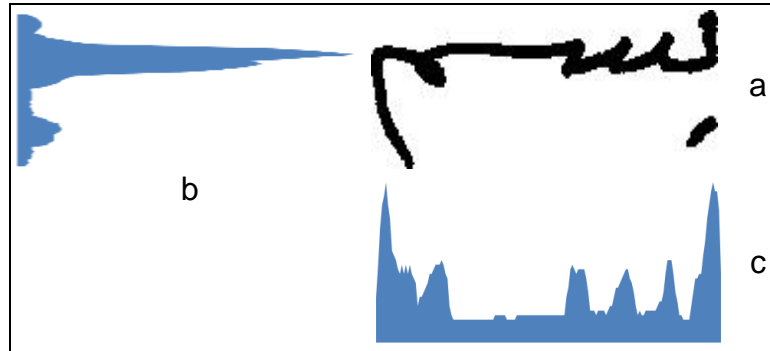


**Figure 54:** Invariant moment results for a subword

#### 4.2.1.4 Projection Features (Horizontal and Vertical Projection):

The image (subword) projection is found by calculating the number of black pixels when scanning a certain binary image horizontally ( $f_{14}$ ) and vertically ( $f_{15}$ ), as shown in Figure 55 below. These features are different from the previous ones in that the previous features are single valued whereas these two are vectors each coordinate of which represent the number of written pixels in horizontal/vertical direction. For most subwords, these features are influenced greatly by the connected strokes at the end of subwords, and these projections encapsulate such information implicitly.

Although, the visible part of this projection look like histograms, but they may not cover a similar number of coordinates unless we append by 0's. This will make it difficult to define a similarity/distance function in the same way done for the other features. In our experiments, we shall test two different similarity functions.



**Figure 55:** Subword Projections, a: Subword, b: Horizontal projection, c: Vertical projection

#### 4.2.2 Handwritten Text Distance/Similarity Functions

Throughout this thesis, identification decision will be based on the nearest neighbour classifier determined by an appropriate distance/similarity function. In some cases, our results allow the determination of accuracy rates at different ranks. Recall that positive identification at rank  $k$  is accepted as long as the identity of the named writer matches the identity of one of the nearest  $k$  neighbours. Note that the distance/similarity function depends on the nature of various feature attributes defined above. For the last two projection features, similarity functions have to be different from the other single-valued feature attributes. Histogram intersection is a commonly used similarity function between histograms defined over the same range of indices. However, the horizontal/vertical projection features defined above may have different range of indices, and therefore we shall experiment with the following two options:

1. The **City Block distance**: Extend the shorter projection by appending sufficient 0 values to have the same range of the long one and sum up the differences between the projection values over the full range.
2. The **Dynamic Time Warp (DTW)**: This is based on treating projections as time series that are possibly sampled at different rates over a fixed time/space. (See the definition below).

For other  $k$ -dimensional real feature sub-vectors of the 13-dimensional feature vector, defined above, that exclude the last two, one can use the Euclidian or city-block distance functions.

### **Dynamic Time Warping (DTW) distance**

DTW is a well-known algorithm commonly used as a similarity function between time series where the data may not have been sampled at the same rate. A time series is a real-valued function defined by successive data/measurements taken over a period of time or along any space line/arc. Suppose that we place two humidity sensors in different parts of a long beach and wish to conduct a daily comparison of the level of humidity in those two places. Assume further that one sensor takes a measurement every 20 minutes and the other is slower and takes a measurement every 30 minutes. In this case, the number of daily measurements collected by the first sensor is 72 but for the second is 48. The obtained data are two vectors but of different lengths, and comparing them cannot be done by usual distance functions. DWT is a measure of similarity used for comparing time series of different lengths. The two projection features,  $f_{14}$  and  $f_{15}$ , can be considered as time series, and the number of projection scans is affected by the style of writing at the time of recording. The DTW similarity function has been used for speech recognition (see (Brik, et al., 2013)) reflecting the fact that two speakers of the same word/sentence may utter the speech at different speeds. DTW has also been applied to many other fields like handwriting recognition. (Brik, et al., 2013), (Güler & Meghdadi, 2008), (Kohonen, 1982), and (Niels, et al., 2005)

Let  $A=[a_1 \ a_2 \ \dots \ a_i \ \dots \ a_n ]$  and  $B=[b_1 \ b_2 \ \dots \ b_i, \ \dots \ b_m]$  be two sequences/vectors of dimension  $n$  and  $m$  respectively. The DTW works by aligning both ends of the two sequences and warping the time/space axis iteratively until an optimal match between the two sequences is found, see Figure 56 for illustration. The optimal criteria for warping are based on the trend of temporal change in the two data sequences. This can be illustrated on an  $m \times n$  grid where the horizontal axis represent the domain of  $A$  and the vertical axis represents the domain of  $B$ , see Figure 57

The time axis is warped so that each data point in the green sequence ( $A$ ) is optimally aligned to a point in the blue sequence ( $B$ ). The best alignment between  $A$  and  $B$  is determined by a path  $P = (p_1, \dots, p_s, \dots, p_k)$  through the grid, called the warping function, from the common, where  $p_s = (i_s, j_s)$  Which minimizes the total distance between them.

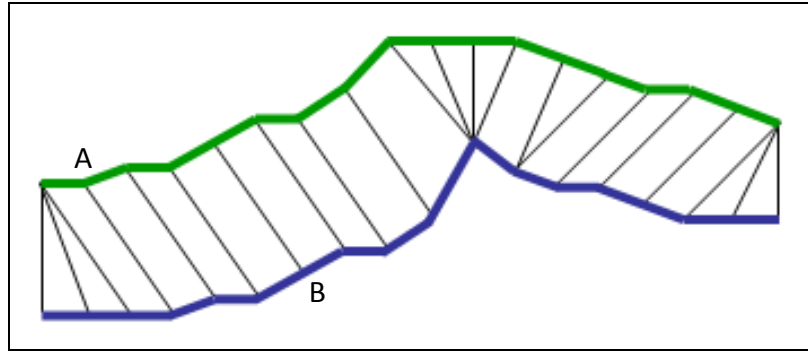


Figure 56: Illustration of DTW (Mathieu, 2009)

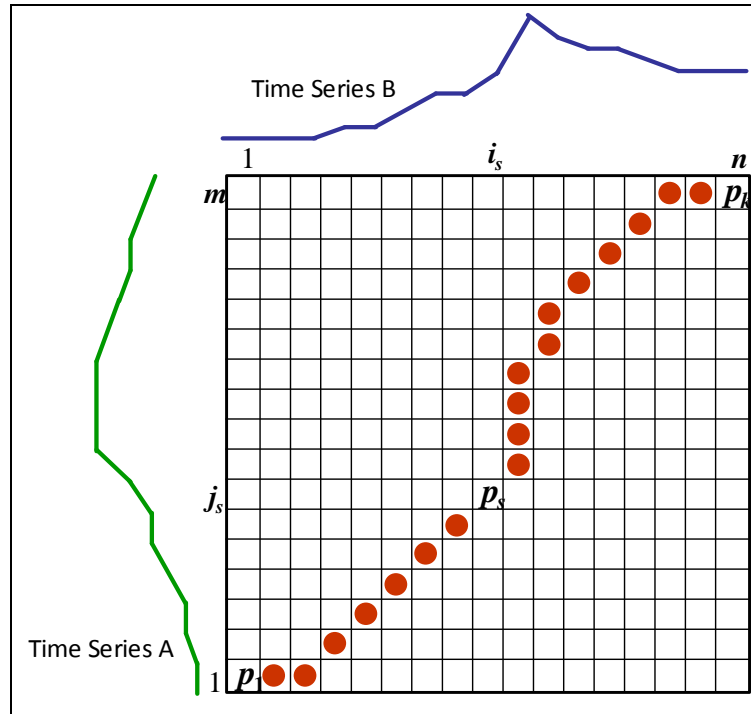


Figure 57: Warping Function (Mathieu, 2009)

Time-normalized distance between A and B is then defined by the formula:

$$D(A, B) = \left[ \frac{\sum_{s=1}^k d(p_s) - w_s}{\sum_{s=1}^k w_s} \right] \quad (4.14)$$

Where  $d(p_s)$  = distance between  $i_s$  and  $j_s$ , and  $w_s > 0$  is the weighting coefficient.

Best alignment path between A and B is defined as:

$$P_0 = \arg \min_p (D(A, B)).$$

And the Weighting Coefficient at position  $p_s$  is defined as:

$$w_s = (i_s - i_{s-1}) + (j_s - j_{s-1}).$$



### 4.3 WI Scheme(s) for Arabic Handwritten Text

Having identified 15 features ( $f_1, \dots, f_{15}$ ) to be extracted from handwritten Arabic text for use as feature vectors representing a writer of such text component (i.e. subword), and identified possible distance and similarity functions we can envisage a number of WI schemes each using all or a sub-vector of the 15 dimensional feature vectors space. Note that the last two coordinates are fundamentally different from the others, and hence they may need different considerations depending on the choice of distance function: City-block or DTW.

#### 4.3.1 Feature Selection- Experimental Design

Although 15-dimensional vector representation of biometric traits is by no mean inhibiting, but there are obvious correlations between some of the above features. We shall demonstrate that including all the 15 feature attributes reduces, rather than increases, the WI accuracy rate. Hence, we need to either use a feature selection technique to obtain the smallest set of feature attributes sufficient to attain the best accuracy rate, or apply some dimension reduction procedure to obtain a shorter meta-feature vector whose attributes are linear combinations of the 15 coordinates but with significantly reduced inter-correlation.

A suitable feature selection scheme have been used in face recognition which was based on ranking the feature attributes, in terms of their discriminating power, and iteratively fusing these features one by one according to their ranks until no further improvement in accuracy was possible, (Abboud & Jassim, 2012) and (AL-ASSAM, 2013). Accordingly, the experimental work in this section is done in steps:

1. We test the accuracy achieved by each single feature. We call this **Single Feature Test (SFT)**.
2. Weights are associated with each feature attribute in the STF.
3. Multiple features are selected using the incremental feature addition technique, to determine the best set of features required for the optimal identification accuracy.

Since, the number of possible subwords in Arabic text is large, and many subwords appear in different words then it would practically desirable to limit the number of subwords to be used for WI. Hence, once the shortest optimal handwritten text feature vector is determined by the above mentioned incremental method we should seek to

determine a relatively small set of subwords for which the selected feature vector obtained in step 3 yields the optimal WI accuracy rate.

To be able to achieve this task, we need to use a text-dependent dataset of handwritten Arabic texts that includes repeated writings of sufficiently long texts written by a reasonably large number of Arabic speaking persons. The database developed by Al-Ma'adeed et al. (see (Al-Ma'adeed, et al., 2008)) is adequate for this purpose and throughout this chapter, we will be using it as the performance testing database. The identification of small list(s) of subwords that are known for their writer discriminating property would be useful for use with text-independent scenarios and related terrorism fighting and for forensic applications.

This will be the subject of the chapter 6, where we use the outcome of the experiments in this chapter to test the performance of WI our in-house in text-dependent DB that consists of the handwriting of 50 writers who wrote two different texts each.

An alternative to the above incremental feature selection approach is the use of dimension reduction tools that are designed to remove the effect of dependencies between some of these attributes. The use of the most common tool of PCA (Principle Component Analyses) for dimension reduction is not suitable due to the small size of the feature vectors. Instead, we shall investigate, in section 4.4, the use of the recently emerging paradigm of compressive sensing using random projection matrices that are widely known for simplicity and for their desirable effect on classification problems.

In this section, we implement the above 3-steps process and conduct experiments to select the best performing set of features selected from the 15 feature vectors described in the previous section. Initially, we test the performance of each feature singularly, and then we adopt an incremental approach whereby we start with the best subwords based discriminating feature, and add the next best discriminating feature. At each stage of this incremental approach, we will have a new proposed WI scheme that is based on the selected feature sub-vector.

The experiments in this section *are not designed to prove the validity of our main hypothesis* on subwords based WI scheme. But these experiments are designed as a pilot to identify a subset of the 15 features and relatively small set of subwords that have good WI discriminating characteristics. These experiments test the performance of any combination of feature subvectors and subwords over a benchmark database of specific handwritten text documents, each consisting of 27 words and are segmented into 49

subwords. The 27 words documents were collected in a database for testing a WI scheme, by Al-Ma'adeed, see (Al-Ma'adeed, et al., 2008). This text text-dependent database consists of handwritings of 100 writers who were asked to write the 27 words into pre-set text boxes in one document, and the process is repeated 20 times per person. In fact, the text boxes were coloured, and a baseline for text was included. Since the words are written inside prepared textboxes, then there was no need to segment the documents and extract the words for the WI experiments conducted by Al-Ma'adeed et. al. However, in our experiments we must apply automatic subwords segmentation of each document before extracting the 15 feature vectors, which could result in errors with adverse impact on WI accuracy.

Although the database includes the recorded handwritten texts for 100 persons, the records for 5 persons are removed due to various errors. In total there are  $20 \times 95 = 1900$  handwritten text documents in the database, each consisting 27 selected Arabic words. In terms of subwords, each of the documents consists of 49 different subwords of different lengths. Three of these 49 subwords are of length 1, and one might wonder whether such subwords can have writer discriminating powers. The following table (Table 4.2) shows the different 49 subwords together with the number of times each is repeated within the document.

**Table 4.2:** Repetition of the 49 subwords occurring the 27 words document

No of repetition	9	4	2	1
Subword	أ	و	لر - لله - م	على - عن - بخير - حيم - حمن - بسم - لي - في - هي - هو - لك - لشكر - يل - جز - لكم - من - لمحتر - ته - كا - بر - حمه - ر - عليكم - لسلا - لسيد - بعد - طيبه - تحيه - قيع - لتو

Here we present the results of one of the experimental protocols which are based on using 25% of the DB documents for training and 75% for testing. Other experiments in which we used 50%-50%, and 75%-25% training to testing ratios all had similar results. This is most likely to be due to the fact that the writing of the 20 documents was done in a single recording session for each writer included in the Al-Ma'adeed et. al database.

#### 4.3.2 Single Feature Test (SFT)

In this section, we estimate the writer is discriminating power of each single feature when tested using the 49 subwords that constitute the various written samples included in the

Al-Ma'adeed et. al database. Although the database contains handwritten samples for 100 writers, the data are reliable for only 95 writers.

Although each writer contributed to the writing of the 27-words document 20 times, but very little variation in style can be expected between the 20 samples. Accordingly in this section, 5 documents for each writer were selected randomly for inclusion in the gallery. Since this is just a pilot experiment, for each writer and each of the 49 written subwords, we average each of the 15 feature attributes over the 5 training samples. We shall also average each of the 15 feature attributes, for each of the 49 subwords, over the 15 remaining samples of documents to be used for testing. Hence, in the gallery, each writer is represented by one list, labelled by the identity of the writer, of the corresponding averaged 15-dimensional feature vectors, one for each of the 49 subwords in the written documents. Moreover, in the testing set each writer is also represented by one list of 49 averaged 15-dimensional feature vectors, one for each subword. Once we analyse the results of the SFT experiments and picked the best performing feature sub-vector in the incremental scheme, in the subsequent experiments we should not use the averaging of feature attributes but use each written document as a separate written sample.

The accuracy rate for each feature  $f_i$  is calculated using the procedure, below. For the two projection features, we normalised the length of indices by adding 0's as necessary, and we use the city-block as the similarity function for matching.

```

Initialise an array of integers  $Match[1..95]=0$ ;
Total_Matches=0; Total_Tests=95*49;
For p = 1 to 95
  For sb = 1 to 49
     $fp = f_i(p, sb)$ ; //Feature value from the testing sample
    nearest=1;  $d = \text{dist}(fp, f_i(1, sb))$ 
    For j= 2 to 95
      If  $\text{dist}(fp, f_i(j, sb)) < d$  then {  $d = \text{dist}(fp, f_i(j, sb))$ ; nearest=j;}
    End;
    If (nearest =p) then  $Match(p, sb)++$ ;
  End;
  Total_Matches += Match(p);
End;
Accuracy( $f_i$ ) =  $100 * \text{Total\_Matches} / \text{Total\_Tests}$ .

```

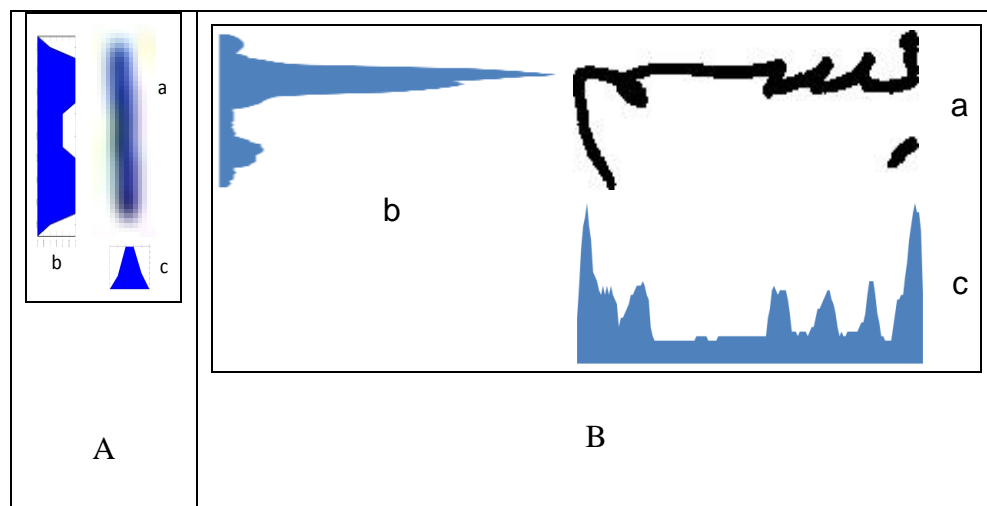
The final accuracy rates for all the 15 features are shown in Table 4.3 below.

**Table 4.3:** Writer identification accuracy for each individual feature

Feature	Accuracy %	Feature	Accuracy %
$f_1$	20.168	$f_9$	16.447
$f_2$	27.491	$f_{10}$	19.088
$f_3$	24.850	$f_{11}$	15.726
$f_4$	27.731	$f_{12}$	15.606
$f_5$	19.328	$f_{13}$	16.567
$f_6$	19.928	$f_{14}$	61.945
$f_7$	20.408	$f_{15}$	57.143
$f_8$	16.327		

Surprisingly, at least two of the features  $f14$  (horizontal projection) and  $f15$  (vertical projection), yield relatively high WI accuracy. In fact, these results split the 15 features in terms of their discriminating characteristics into 4 distinct groups. The *high discriminating group* of features are  $f14$ , and  $f15$  achieve high identification rates of 61.94% and 57.143% respectively, more than double the accuracy of the next nearest group of features. The second group of features are the ( $f4$ ,  $f2$ ,  $f3$ ,  $f7$ , and  $f1$ ) have achieved reasonable writer discriminating accuracy between 27.73% and 20.17%, and the third group of features ( $f6$ ,  $f5$ , and  $f10$ ) have scored between 19.93 and 19.09. Other features  $f13$ ,  $f9$ ,  $f8$ ,  $f11$ , and  $f12$  have scored the lowest rate (weak features).

Note that not every subword produces such significant accuracy value in any of the above feature groups. For example, the ( $f14$  and  $f15$ ) projection features result in high accuracy in identifying writers of the subword (بسم) but doesn't help discriminate between writers of the subword (إ), see Figure 58



**Figure 58:** A. projections for (إ), B. projections for (بسم), a. subword, b. horizontal and c. vertical projection.

A close look at the number, per feature, of subwords that contribute to most identification errors, reveal interesting patterns. In the table below (Table 4.4), we list for each groups of feature of the 4 groups of features, the set of subwords that have a negative impact on accuracy achieved by the groups.

**Table 4.4:** Subwords with errors way above the overall errors for each feature group plus the number of error places

Sub word	Group 1 (> 50% error)		Group2 (> 85% error)					Group 3 (> 90% error)			Group 4 (>96% error)				
	F14	F15	F1	F2	F3	F4	F7	F5	F6	F10	F8	F9	F11	F12	F13
أ	2	7	8		7	7	7	4	4	1	3	5	4	5	4
ر		1	1		1		1				1				1
و	2	3	4	1	2	1	2	2	1		1			1	1
م	1	2	1				1				1	1	1	1	
بر		1			1	1				1	1	1	1		
هو				1			1				1				
ته														1	1
يل		1									1		1		
كا	1	1				1				1			1		
لسلا	1	1	1	1	1		1								
لشكر			1	1		1				1				1	
طيبه	1		1									1	1	1	
عليكم	1	1				1	1						1	1	1
الله						1									
لكم							1								
حمن															1

For many features, it is evident that subwords of length one (constructed by just one letter) brings higher error rates than longer subwords. In fact, the best writer discriminating subwords are of length 2-4 for all feature attributes. This seems to confirm a common expectation that little information if any, about the writer is conveyed by length 1 subwords. Understandably “أ”, is the least writer discriminating subword due to the fact it is only present in a vertical line. *In addition, the weak features group bring up fewer subwords in comparison with the others.*

The significant differences in accuracy achieved by the 4 different groups of features indicate that we may not need all the 15 features or the 49 subwords to achieve good WI accuracy rates. We shall first settle the question of selecting a smaller subset(s) of WI discriminating features. For this, we have designed an incremental method to build a reliable WI scheme that uses a small subset of features, identified according to what we call “writer discriminating weight” defined, below, for each in terms of the values achieved in Table 4.4 above. The question of selecting a subset of the 49 subwords will be settled later in terms of the developed incremental scheme.

### Features' WI-discriminating weights

Based on the individual feature result as shown in Table 4.4 weights can be associated with individual features, as illustrated in Equation (4.15), which we would use as a ranking of these features in term of subwords based WI discriminating strength.

$$F_w = \frac{IFA}{\sum F_s} \quad (4.15)$$

Where:  $F_w$ =Feature Weight,  $IFA$ =Individual feature accuracy, and  $F_s$ = Features scores,

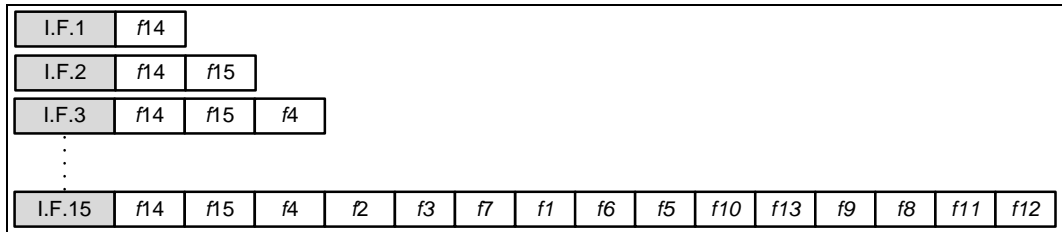
Table 4.5 shows the features in descending order based on weights deriving from the above Equation (4.15).

**Table 4.5:** Feature weight calculated based on their accuracy

Feature	Accuracy%	Feature Weight	Feature	Accuracy%	Feature weight
<i>f14</i>	61.94	0.1636	<i>f5</i>	19.33	0.051
<i>f15</i>	57.14	0.1509	<i>f10</i>	19.09	0.0504
<i>f4</i>	27.73	0.0732	<i>f13</i>	16.57	0.0437
<i>f2</i>	27.49	0.0726	<i>f9</i>	16.45	0.0434
<i>f3</i>	24.85	0.0656	<i>f8</i>	16.33	0.0431
<i>f7</i>	20.41	0.0539	<i>f11</i>	15.73	0.0415
<i>f1</i>	20.17	0.0532	<i>f12</i>	15.61	0.0412
<i>f6</i>	19.93	0.0526			

### 4.3.3 The Incremental WI Scheme

In this section, we determine the optimal number of features that is necessary to achieve the highest accuracy by incrementally adding all features, one by one, in the descending order of discriminating characteristics. This process is called **Incremental Features (I.F.)**. Our experiments, start with *f14* as the highest discriminating feature, and add the rest as depicted in Figure 59, below. This order is in accordance with results of **Error!**  
**Reference source not found.**



**Figure 59:** Incremental Features (I.F.) list of Experiments

In these experiments, we followed the same protocol that was used in the SFT in terms of the ratio of training to testing samples. Again the feature sub-vectors were represented by the average value of the training/testing sets. Note that for *f14* and *f15* we normalised the length of the indices by appending with sufficient number of zeros. Therefore, the



feature vectors used for any scheme that included  $f_{14}$  and  $f_{15}$  has dimension  $(n_h + n_v + m)$ , where  $n_h$  is the normalised length of the horizontal projections,  $n_v$  is the normalised length of the vertical projections, and  $m$  is the number of other features. As for the distance/similarity function, we used Euclidean distance in the resulting  $(n_h + n_v + m)$ -dimensional space.

Figure 60 shows the results of these experiments. It is clear, that the performance our incremental schemes continue to improve at every step while we add the 8 features  $\{f_{14}, f_{15}, f_4, f_2, f_3, f_7, f_1, f_6\}$  reaching an optimal accuracy of slightly above 82% but the performance deteriorates afterward i.e. adding more features will lead to reduced accuracy.

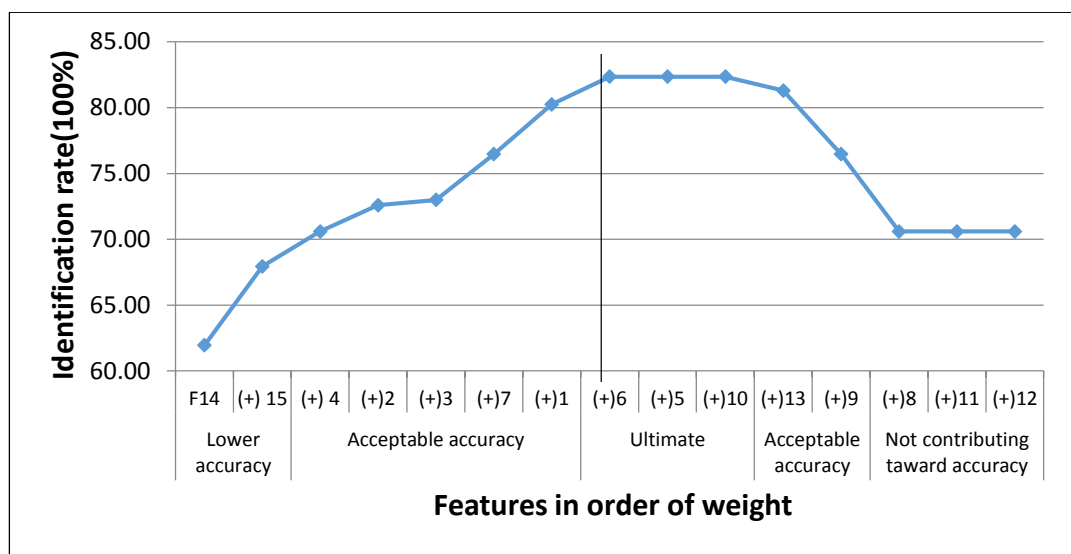


Figure 60: Incremental Features Performance Chart

From Table 4.4 we observe that removing the 3 length 1 subwords from the tests is expected to improve the accuracy well beyond the current 82.5%. Moreover, in these experiments we used automatic subword segmentation which cannot be perfect and segmentation errors undoubtedly cause reduced WI accuracy.

### The most effective combination of features and subwords

In order to understand the message the above chart conveys, we need to analyse the results further to determine the contribution of different groups of subwords to this pattern of accuracy. Here we could use the results of **Error! Reference source not found.** to guide our discussion, but that table was based on using each and all the single features rather than groups of features. First of all, we identify three distinct groups of features in terms of the effect of their addition to the feature vectors:

1. **Significant features** consisting of the  $\{f_{14}, f_{15}, f_4, f_2, f_3, f_7, f_1, f_6\}$  features whose addition in that order to the incremental procedure monotonically increases WI accuracy.
2. **Neutral Features** consisting of the  $\{f_5, f_{10}\}$  features whose addition in the incremental procedure, after the previous list, have no effect on WI accuracy.
3. **Bad features** consisting of the remaining set  $\{f_{13}, f_9, f_8, f_{11}, f_{12}\}$  of feature whose addition in the incremental procedure, after the above two sets, lead to a reduction in WI accuracy.

While incrementing the number of significant features, we note that only using the first 3 features ( $f_{14}+f_{15}+f_4$ ), the top group of 8 subwords, together with the words within which they appear, contribute most to the WI accuracy rate.

Subwords	حمن	لي	في	بعد	هو	طيبة	تحية	لمحتر
Parent words	الرحمن	الى	في	وبعد	هو	طيبة	تحية	المحترم

On adding the next two significant features ( $f_2+f_3$ ) to the above combination, has led to even better results with an expanded list of 13 useful subwords, the additional subwords being:

Subword	من	حيم	عن	لشكر	لسيد	لله
Parent words	من	الرحيم	عن	الشكر	السيد	الله

The third step of adding ( $f_7+f_1$ ) also improved the accuracy and expanded the list to 17 useful subwords, the additional subwords being:

Subword	لر	بخير	ته	بسم
Parent words	الرحمن	بخير	وبركاته	بسم

Finally, when added ( $f_6$ ) to above list feature combination the accuracy improved further and the list of best WI discriminating subwords expanded to 22.

**Table 4.6:** Best WI discriminating Subwords with Feature vector ( $f_{14}+f_{15}+f_4+f_2+f_3 +f_7+f_1+f_6$ )

no	Subword	no	Subword	no	Subword
1.	على	2.	عن	3.	بخير
4.	حيم	5.	لر	6.	حمن
7.	لر	8.	لله	9.	بسم
10.	لي	11.	في	12.	هي
13.	لك	14.	جز	15.	لكم
16.	من	17.	لمحتر	18.	لسيد
19.	بعد	20.	تحية	21.	قيع
22.	لتو				

#### 4.3.4 Writer Identification Rate (WIR)

In the above section, the pilot experiments identified the best group of features that also enabled us to extract the best group of subwords to be used in writer identification. Achieving around 82.5% accuracy rate of WI at the top rank nearest neighbour can be seen as outperforming existing WI from Arabic handwritten words. In fact the word-based WI algorithm of Al-Ma'adeed et al (see (Al-Ma'adeed, et al., 2008)) only achieves 90% accuracy at rank 10 nearest neighbour, i.e. a positive decision is made as long as it is matched to the correct person within the top 10 nearest neighbours. Admittedly, this claim may have been boosted by the fact that in the pilot experiments we averaged the feature vectors' attributes over the 15 testing samples and over the 5 training samples. However, to some extent these results indicate the validity of our hypothesis of using subwords rather than words for WI from Arabic handwritten text. But substantiating our claim we need to avoid *averaging* samples whether for testing or for gallery and instead we use the above identified 8-dimensional feature vector for each sample of written subwords as testing or a training feature vector.

In what follows we report the results of 2 similar experiments, the only difference between them is that in the first we use the DTW distance functions for the two projection scheme while we used the Euclidean distance to all the 8 features including the projection ones where we append by sufficient 0's if necessary as above. In the first case, the distance between test subword sample and a gallery template sample is:

$$\text{Dist} = \text{Euclidean (6 features } [f_4, f_2, f_3, f_7, f_1, \text{ and } f_6]) + \text{DTW } ([f_{14}]) + \text{DTW } ([f_{15}])$$

In both experiments, for each of the 95 writers and each of 22 subwords, listed in Table 4.6, we input to:

**The Gallery:** 5 samples/subword/writer of the 8-dimensional feature vectors (Total = $5 \times 22 \times 95 = 10450$ ), and

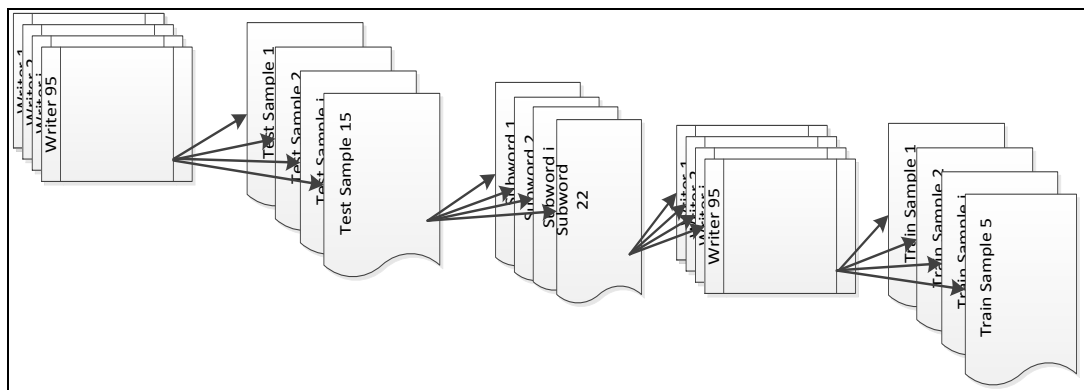
**The Test set:** 15 samples/subword/writer of the 8-dimensional feature vectors (Total = $15 \times 22 \times 95 = 31350$ ).

In these experiments, we tested the performance of our proposed scheme for 2 different purposes:

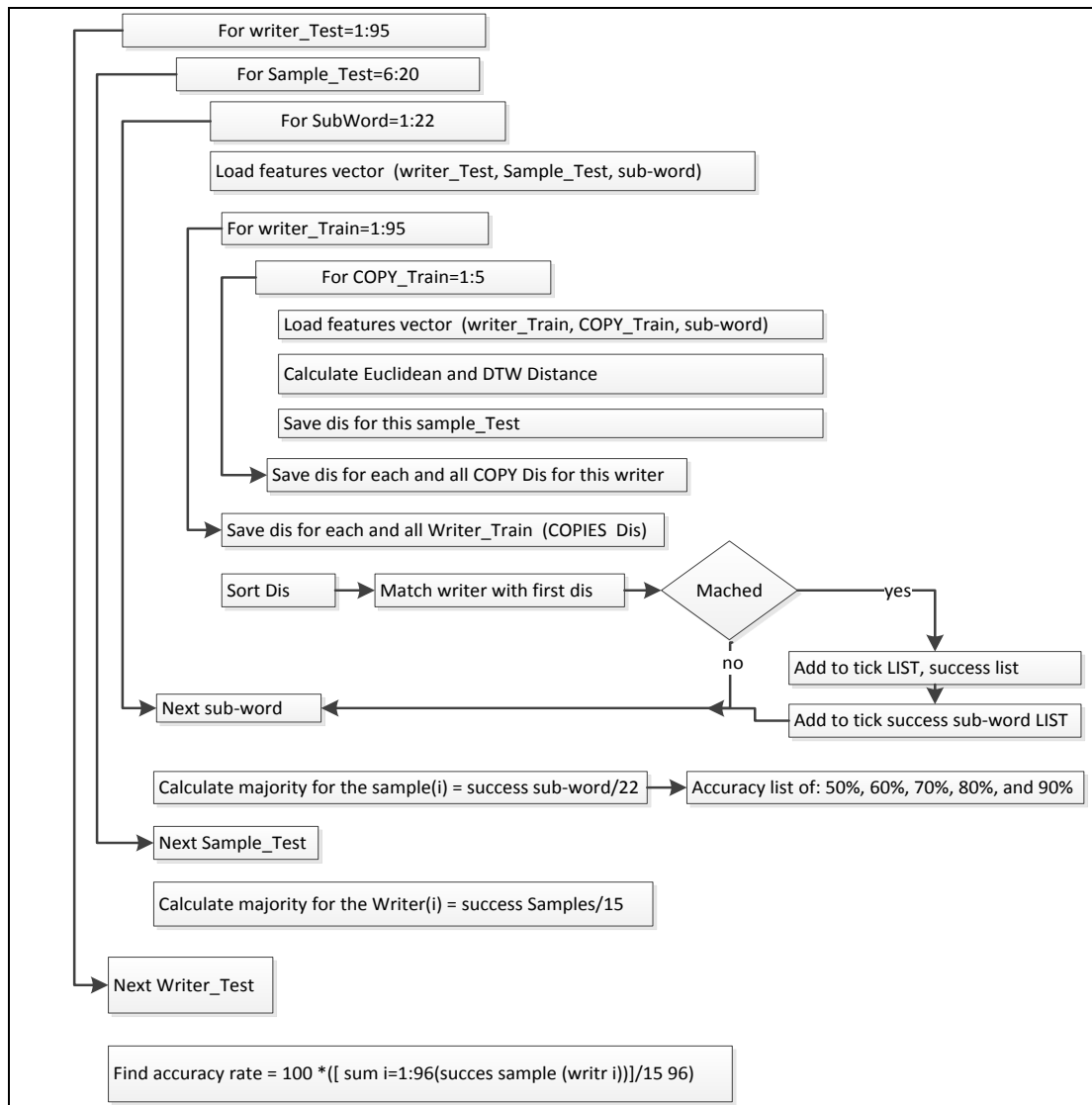
**Case 1:** Identification is based on one subword. In this case for each writer success is declared if the majority of the 15 testing samples return the accurate writer.

**Case 2:** Identification is based on a full document of the 22 subwords. In this case for each writer success is declared if the majority of the 15 testing document return the accurate writer.

The diagram in Figure 61 illustrates the structure of the testing. Moreover, the flowchart given in Figure 62 helps calculate the number of successful test and accuracy for all the three cases above.



**Figure 61:** Structure of the testing system



**Figure 62:** WI system flow diagram

### Results for experiment 1

Table 4.7, below, gives the identification accuracy based on each single subword of the list of 22 subwords. For each subword, a writer is identified if the majority of his/her 15 written samples of subword have been matched at rank 1. These results provide a reasonable evidence of the validity of our hypothesis that subwords in the handwritten Arabic text have WI discriminating power. Indeed, a single subword has achieved over 80% accuracy in identifying the writer, 3 other subwords achieving accuracy in access to 70% identification accuracy, and only one subword has just missed the 50% accuracy. Moreover, the overall identification accuracy achieved when using all subwords list is

$$100 * (19948/31350) = \mathbf{63.63\%}.$$

**Table 4.7:** Subwords success hits and accuracy in descending order of accuracy

no	subword	success hit	Wr Ident %	no	subword	success hit	Wr Ident %
1	لكم	1144	80.2807	12	على	908	63.7193
2	هي	1031	72.35088	13	لسيد	901	63.22807
3	حيم	1014	71.15789	14	الله	884	62.03509
4	حمن	1007	70.66667	15	جز	882	61.89474
5	لمحتر	997	69.96491	16	قيع	874	61.33333
6	بخير	983	68.98246	17	لك	855	60
7	بسم	968	67.92982	18	لر	775	54.38596
8	لى	959	67.29825	19	لتو	767	53.82456
9	من	947	66.45614	20	طبيه	755	52.98246
10	عن	931	65.33333	21	لر	754	52.91228
11	في	921	64.63158	22	بعد	691	48.49123
<b>SUM of Success Subwords</b>						<b>19948</b>	

In the next set of results obtained from this experiment, we take each of the 15 documents containing all the 22 subwords as one testing trial and matching is declared if the majority of the subwords were matched. Table 4.8 lists the number of successfully match documents for the writers descending order of success. The results in this table first give the accuracy rate for writer identification is given by:

$$\begin{aligned}
 & 100 * \left( \frac{\sum_{wr=1}^n \text{success samples}}{\text{no of writers} * \text{no of testing samples}} \right) \\
 & = 100 * (1177 / (95 * 15)) \\
 & = \mathbf{82.59649\%}
 \end{aligned}$$

This level of accuracy achieved with 22 subwords and at rank 1 matching is more favourable, by a long way, to what was achieved by the results of word-based WI algorithm of Al-Ma'adeed et al (see (Al-Ma'adeed, et al., 2008)) which achieves around 90% accuracy but only at rank 10. We do not see any difficulty in claiming that we can achieve near optimal accuracy if we to consider matching at a slightly higher than rank1. Here, we are more interested in pointing out that these results also provide further evidences to the validity of our hypothesis that subwords have much stronger WI discriminating characteristics in Arabic handwritten text.

A closer look at the table reveals few interesting pattern and a hint of the presence of a Doddington Zoo phenomena (see (Poh, et al., 2006)). Table 4.8 shows that there are 34

writers (i.e., 36%) have all their documents were correctly identified. The number of correctly identified has increased to 56%, 65%, 73% writers when we tolerate 1, 2, or 3 error documents, respectively. Although demonstrating the appearance of a Doddington Zoo phenomenon is outside the scope of this thesis, we shall nevertheless follow up at the performance of the next refined versions of our current algorithm for these different groups of writers. However, it is only prudent to refrain at this stage from using the traditional Doddington Zoo terms of Sheep, Goats, Lamb and Wolves. Instead and for ease of discussion and performance comparison throughout the thesis, we shall for each WI scheme categorise the group of tested writers as

1. **Professional writers:** if they are identified in  $\geq 80\%$  of their tested document
2. **Normal writers:** if they are identified in more than 50% but  $<80\%$  of their tested document
3. **Unknown/Amateur Writers:** otherwise.

Another worthy observation is that the above overall accuracy is only 2.4% higher than the accuracy of identification that was achieved when the single subword “لكم” was used as the basis for writer identification. Consequently, the addition of the other 21 subwords did not help as one expect. Perhaps, we can speculate that higher improvement in accuracy can be achieved by using fewer than 22 subwords. Although an incremental addition of the subwords according to their WI discriminating power, as shown in Table 4.6, one can identify the smallest subset that achieve highest accuracy so that addition of any other one will result in deteriorated accuracy. However, this may be true only in this database. Instead, we shall next simply consider the performance when we only use the best 11 performing subwords, all of whom yield above the average accuracy.

**Table 4.8:** Success samples of all writers (out of 15 in each sample) in high to low order

no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample
1	1	15	20	52	15	39	11	14	58	30	13	77	37	10
2	4	15	21	58	15	40	12	14	59	32	13	78	54	10
3	7	15	22	60	15	41	14	14	60	55	13	79	76	10
4	9	15	23	63	15	42	21	14	61	66	13	80	78	10
5	17	15	24	64	15	43	24	14	62	98	13	81	31	9
6	19	15	25	65	15	44	38	14	63	13	12	82	77	9
7	26	15	26	67	15	45	41	14	64	28	12	83	22	8
8	29	15	27	68	15	46	42	14	65	44	12	84	71	8
9	33	15	28	69	15	47	57	14	66	53	12	85	86	8
10	34	15	29	73	15	48	61	14	67	59	12	86	88	7
11	35	15	30	81	15	49	74	14	68	75	12	87	25	6
12	36	15	31	84	15	50	79	14	69	82	12	88	85	6

no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample
13	40	15	32	93	15	51	90	14	70	16	11	89	39	5
14	43	15	33	95	15	52	91	14	71	46	11	90	15	4
15	45	15	34	97	15	53	99	14	72	49	11	91	27	4
16	47	15	35	2	14	54	10	13	73	62	11	92	89	4
17	48	15	36	3	14	55	18	13	74	70	11	93	80	3
18	50	15	37	5	14	56	20	13	75	72	11	94	92	3
19	51	15	38	8	14	57	23	13	76	6	10	95	94	0
SUM of Success samples of all writers														<b>1177</b>

WI accuracy using any single subword out of the 11 subwords can be calculated directly from Table 4.7:

$$100 * (10902/15675) = 69.55 \%$$

The next table shows the results obtained from repeating the main experiment, but taking each of the 15 documents containing all the 11 subwords as one testing trial and matching is declared if the majority of the subwords were matched. Table 4.9 lists the number of successfully match documents for all writers in descending order of success.

**Table 4.9:** Success samples of all writers using 11 subwords (out of 15 in each sample) in high to low order

no	Wr no	succ sample	No	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	No	Wr no	succ sample
1	1	15	20	52	15	39	13	14	58	11	13	77	80	11
2	4	15	21	58	15	40	20	14	59	16	13	78	86	11
3	5	15	22	60	15	41	22	14	60	32	13	79	98	11
4	7	15	23	61	15	42	24	14	61	55	13	80	44	10
5	9	15	24	63	15	43	26	14	62	99	13	81	71	10
6	14	15	25	64	15	44	33	14	63	8	12	82	82	10
7	17	15	26	65	15	45	38	14	64	10	12	83	49	9
8	18	15	27	66	15	46	42	14	65	23	12	84	77	9
9	19	15	28	67	15	47	43	14	66	28	12	85	85	9
10	29	15	29	68	15	48	50	14	67	35	12	86	21	8
11	30	15	30	69	15	49	57	14	68	37	12	87	15	7
12	34	15	31	70	15	50	62	14	69	54	12	88	27	7
13	36	15	32	72	15	51	74	14	70	79	12	89	25	6
14	40	15	33	73	15	52	75	14	71	90	12	90	88	6
15	41	15	34	81	15	53	76	14	72	6	11	91	46	4
16	45	15	35	93	15	54	84	14	73	31	11	92	89	4
17	47	15	36	95	15	55	91	14	74	53	11	93	92	3
18	48	15	37	97	15	56	2	13	75	59	11	94	39	2
19	51	15	38	12	14	57	3	13	76	78	11	95	94	0
SUM of Success samples of all writers using 11 subwords														<b>1198</b>

**In this case,** identification accuracy improved by another modest amount of 2% to:

$$100*(1198/ (95*11)) = 84.0702\%$$



From this table, we see further hints of the presence of the Doddington Zoo phenomena, although the group of writers declared above as unknown (which would have been considered as wolves) has changed slightly with only one writer becoming recognisable as a normal (in our categorisation writer to the system and few more document the others document became identifiable with their writers. Also, there were an increased number of professional writers here increased by 2. The number of writers who have all their documents correctly identified increased by 3 to 37 (i.e., just under 39%).

Although we may be tempted to reduce further the number of tested subwords to get better results, we believe that these results provide sufficient evidences in support of our stated hypothesis that subwords are sufficient for discriminating writers of Arabic text. Instead, we repeated the above two sets of experiments but this time by removing the group of unknown writers. Although one might think that we should accept the failure to recognise few as natural shortcoming reflecting the tough challenge of WI of Arabic handwritten text. However, knowing that handwriting skills are developed through a training process that is influenced by many factors before their writer acquiring a distinct style. But to settle such questions and decide subjectively whether these writers are indeed new to writing in Arabic or the system is unable to detect a distinctive style in their writing, we need to have extra information about such as their age and the length of time they have been writing in Arabic. Unfortunately, no such information is available about the database participants. Hence, I decided to determine the scheme accuracy rate of the scheme by excluding the unknown writers if we to assume the presence of the Doddington Zoo phenomena. Table 4.10 shows a summary of accuracy rates achieved when we include or exclude the unknown writers when we use 22 or 11 subwords. We accept that researchers do not do that in such cases but they use most credible methods such as the score normalisation approaches (Poh, et al., 2010). However, the results of the next refinement of the scheme presented in chapter 5 which will demonstrate that this problem can be remedied.

**Table 4.10:** Performance Summary of our Subwords-based WI schemes

no	Method	Subwords majority success rate %	Samples majority success rate %	Writer majority success rate %
1	<b>All writers + 22 Subwords</b>	<b>63.63</b>	<b>82.60</b>	<b>90.53</b>
2	<b>All writers + 11 Subwords</b>	<b>69.55</b>	<b>84.07</b>	<b>92.63</b>
3	<b>Success writer only + all Subwords</b>	<b>66.89</b>	<b>90.31</b>	<b>100</b>
4	<b>Success writer only + 11 Subwords</b>	<b>72.85</b>	<b>90.70</b>	<b>100</b>

Before we close this section, we shall first determine if the improvement over the pilot could have come about as a result of using the DTW to measure the similarity between the projection features. The next experiment repeats the running of the above code but is done by replacing the DWT code and replaces it with the city block for measuring distances between projections after appending the shorter sequence with sufficient 0's to have the same length of the longest sequence. Ironically, the performance of the latter scheme has dropped dramatically that can only be explained by the fact that the good results achieved in sections 4.3.2 4.3.34.3.3 and 4.3.3 were as a direct consequence of averaging has eliminated the intra-class variation that is present in all biometric schemes. These somewhat disastrous results are shown in Table 4.11 and Table 4.12 below

**Table 4.11:** subwords success hit in order (from high to low) for all testing writers

no	subword	success hit	Wr Ident %	No	subword	success hit	Wr Ident %
1	لكم	862	60.49123	12	قيع	501	35.15789
2	جز	613	43.01754	13	على	483	33.89474
3	حيم	602	42.24561	14	لتو	482	33.82456
4	من	601	42.17544	15	الله	480	33.68421
5	هي	586	41.12281	16	بخير	479	33.61404
6	عن	567	39.78947	17	لر	470	32.98246
7	بسم	554	38.87719	18	بعد	469	32.91228
8	لمحتر	552	38.73684	19	لر	460	32.2807
9	حمن	543	38.10526	20	لكم	438	30.73684
10	في	530	37.19298	21	لسيد	435	30.52632
11	لي	506	35.50877	22	تحبه	341	23.92982
<b>SUM of Success Subwords</b>						<b>11554</b>	

From Table 4.11, we conclude that the WI accuracy of this scheme when using any single subword out of the 22 subwords is:

$$100 * (11554 / 31350) = 36.85486\%$$

This is well below the 63% achieved when we used the DTW as a similarity measure for the two projection features. With such a low performance for the 22 subwords, there was no hope of achieving much better results with 11 subwords. These results also explain the other rather low performance shown in Table 4.12

**Table 4.12:** Success samples of all writers (out of 15 in each sample) in high to low order

no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample
1	58	15	20	81	7	39	16	3	58	40	1	77	30	0
2	64	15	21	90	7	40	78	3	59	59	1	78	31	0
3	51	14	22	99	7	41	84	3	60	60	1	79	32	0
4	29	12	23	4	6	42	2	2	61	62	1	80	36	0
5	65	12	24	41	6	43	21	2	62	69	1	81	46	0
6	75	12	25	17	5	44	23	2	63	76	1	82	49	0
7	9	11	26	19	5	45	43	2	64	79	1	83	53	0
8	52	11	27	33	5	46	47	2	65	82	1	84	54	0
9	42	10	28	63	5	47	55	2	66	91	1	85	61	0
10	73	10	29	72	5	48	66	2	67	98	1	86	70	0
11	34	9	30	8	4	49	77	2	68	3	0	87	71	0
12	48	9	31	14	4	50	7	1	69	6	0	88	74	0
13	97	9	32	20	4	51	11	1	70	10	0	89	80	0
14	57	8	33	45	4	52	12	1	71	13	0	90	85	0
15	93	8	34	50	4	53	22	1	72	15	0	91	86	0
16	24	7	35	68	4	54	26	1	73	18	0	92	88	0
17	38	7	36	95	4	55	35	1	74	25	0	93	89	0
18	44	7	37	1	3	56	37	1	75	27	0	94	92	3
19	67	7	38	5	3	57	39	1	76	28	0	95	94	0
SUM of Success samples of all writers														<b>328</b>

The results in Table 4.12 to the assumption that each of the 15 documents containing all the 22 subwords as one testing trial and matching is declared if the majority of the subwords were matched. We can calculate the accuracy rate for writer identification as follows:

$$100 * \left( \frac{\sum_{wr=1}^n \text{success samples}}{\text{no of writers} * \text{no of testing samples}} \right)$$

$$= 100 * (1177 / (95 * 15)) = \mathbf{23.01754\%}$$

#### 4.4 The Performance of Non-projection Features – Revisited.

In section 4.3.2, the first pilot experiment detected huge differences between the performances of the top 2 projection features ( $f_{14}$ ,  $f_{15}$ ) and the 13 other features. It is not difficult to see that there are some redundancies or dependencies among these 13 features. This may suggest removing some of these features, but this will not improve the performance of the remaining ones, and these features singularly are relevant to writer's style and habits. Instead, it may be possible to retain these features in some way by

replacing the whole list with a smaller number of *meta-features* each being a linear combination of the 13 features with carefully selected weight coefficients. In other words, we shall try to transform linearly the 13-dimensional feature space into a lower dimensional space. This is equivalent removing redundancies in the 13-dimensional feature space by applying dimension reduction transform projections and tests the WI accuracy in the transformed domain.

The process of transforming high dimensional data into a significant representation of reduced dimensionality is known as Dimension Reduction Techniques DRT (van der Maaten, et al., 2009), examples of these techniques are: principal component analysis (PCA), factor analysis (FA), classical multidimensional scaling (MDS), and so on. (van der Maaten, et al., 2007). All these techniques are adaptive, requiring the use of sufficiently large training sets, and the accuracy of such recognition or data analysis tasks are certainly biased by the choice of the training set. Moreover for recognition tasks, these techniques may require retraining when the population increases even by a modest numbers of new participants. An alternative non-adaptive dimension reduction technique is provided by the emerging new paradigm of compressive sensing (ComS). ComS reduces the somewhat severe requirement imposed by the classical Shannon-Nyquist sampling theory, on the number of samples needed to perfectly represent signals, for sparse or approximately sparse signals, i.e. sparse signals and patterns can be restored from what was previously supposed to be incomplete data (Fornasier & Rauhut, 2011). Here we propose using Compressive Sensing (ComS) to reduce the number of features that digitally represent a handwritten subword.

#### **The ComS algorithm:**

The ComS dimension reduction approach is based on linear transforming the 13-dimensional vectors representing the subwords feature vectors that exclude the 2 projection features. ComS-based linear transformations are represented by certain types of rectangular matrices. In our case, these matrices should be of size  $(k \times 13)$  with  $k < 13$  is the number of meta-features obtained. For the ComS approach to compactly represent the given feature vectors, the ComS matrices must satisfy what is known as the **Restricted Isometry Property (RIP)**, which basically means that when applied then the distance between a sparse vector and its image in the transformed lower dimensional space remains sufficiently small. This would allow conducting matching in the transformed space has similar, if not better, performance to matching in the original higher dimensional space. Random Gaussian and Bernoulli matrices are known to be

RIP with high probability, and here we shall test the performance of the ComS algorithm with the two random matrices with  $k=8$ .

For this approach to work, we first need to normalise all subwords 13-dimensional feature vectors using the following equation (4.16), when Norm: Normalisation (*Norm*)

$$Norm_f = \frac{(max_f - f)}{(max_f - min_f)} \text{ where } (f = \text{feature}) \quad (4.16)$$

Where: *Norm*=normalisation, *f*=feature, *max*, *min*=maximum and minimum feature value respectively.

Then the group of 8-dimensional meta-features were generated using Gaussian random 8x13 matrix *Gus* and the Bernoulli random 8x13 matrix

$$Gus = (rand(8,13)) \text{ where } Gus: \text{Gaussian random values} \quad (4.17)$$

$$Ber = (Gus < 0.75) \text{ where } Ber: \text{Bernoulli random values} \quad (4.18)$$

And then the corresponding meta-features were calculated as follows:

$$META_{Gus} = (Gau * Norm_{Features}) \quad (4.19)$$

$$META_{Ber} = (Ber * Norm_{Features}) \quad (4.20)$$

We conducted 2 experiments to test the accuracy of the WI scheme that use the corresponding 8 dimensional Gaussian and Bernoulli meta-features, respectively. We followed the same protocol of pilot experiments of section 4.3.3, i.e. we averaged the meta-features in the 5 training samples and the 15 testing samples. We did not use protocols of section 4.3.3 so that we can compare the performance of using the meta-features with the performance of the 13 single features in section 4.3.2. Moreover, ComS approach is more appropriate for far higher dimensional space than 13-dimensional spaces. This would be more suited for future extended work, in which many more subwords features should be extracted including pressure and wavelet and spectral parameters and using other databases.

Majority rule and nearest neighbours classifier were used for WI tests. In each of the two experiments, we determined the WI accuracy in 4 groups of subwords selected from the 49 subwords. These groups are selected in descending order of discriminating power achieved in the pilot experiment in section 4.3.2:

1. Best performing 7 subwords

عن	بخير	حيم	لر (الرحمن)
لي	هو	من	

2. Best performing 16 subwords:

حمن	لسيد	عن	بخير	حيم	لر (الرحمن)	أ (الله)	لي
في	هي	هو	لشكر	من	أ (المحترم)	أ (السيد)	بعد

3. Best performing 17 subwords (the above list + the subword [ته] ), and

4. All the 49 subwords.

Table 4.13 presents the ComS-based WIR for both the Gaussian and Bernoulli random matrices.

**Table 4.13:** ComS based WI schemes (using Gaussian and Bernoulli methods)

How many subwords used	Gaussian random value %	Bernoulli random value %
7 subwords	57.93	60.14
16 subwords	61.44	59.35
17 subwords with ته	61.65	62.79
All 49 subwords	58.60	58.22

These results obviously outperform the results obtained by any single one of the 13 features and suggest that using ComS like approaches and adopting meta-features that linearly combine the single features has the potential to achieve significant accuracy. As mentioned earlier this provides the incentive to conduct a more extended study in the future to test the validity of this claim.

## 4.5 Summary of the Chapter

In this chapter we investigate, developed and tested subwords based WI for Arabic handwritten text. We have empirically demonstrating the credibility of our hypothesis that such schemes will achieve higher accuracy than those WI schemes that use complete words. The performance of our proposed subwords WI scheme(s) was tested on a publically available text-text-dependent database that is used WI from 27 Handwritten Arabic words.

The developed schemes extracted 15 relevant digital features to represent the discriminating power of subwords. Two schemes were developed through a refinement

process whereby initially the number of features limited to 8 obtained by incremental procedure, and the two schemes are based on first using 22 subwords whereas the second scheme used only 11 schemes. The experimental results confirmed beyond doubts the validity of our hypothesis. Our schemes outperformed existing word-base WI, which only achieves high accuracy at rank 10 nearest neighbours.

Our results exhibited the possibility of the presence of a Doddington Zoo effect. We noted that the performance of both developed schemes (i.e. the 22 subwords scheme and the 11 subwords scheme) had the same pattern in relation to different groups of writers. The groups of writers in the two schemes that have a similar level of accuracy rates were almost identical, and when we removed the lowest group (i.e. wolfs in the Doddington Zoo terminology) the accuracy improved significantly. Accepting that this is not the way that the Doddington Zoo phenomenon is dealt with in the literature, we decided to delay such consideration to a later occasion.

Finally, we conducted another pilot study to investigate the use of compressive sensing (and random projection) approach for dimension reduction to replace the original 13 features with a smaller number of meta-features formed from linear combinations of all the 13 features as an alternative to feature selection. The results are encouraging and provide strong motivation to conduct a more extended study in the future but with a far larger set of single features.

So far the performance of our schemes use subwords as a one unit of text which include a body and one or more diacritics. In the next chapter, we should attempt to decide whether it is essential to include or exclude the diacritics.

## **Chapter 5 : WI based on Subwords without their diacritics**

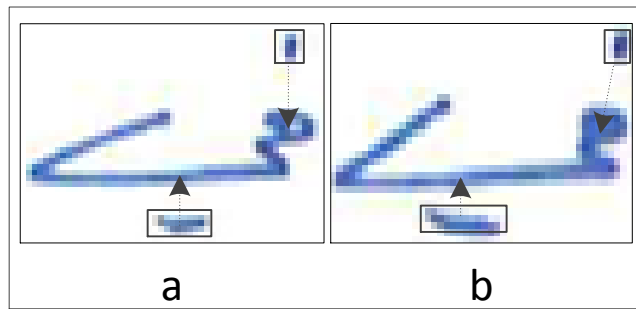
In the last two chapters, we investigated Arabic handwritten text segmentation and the use of subwords (including their own diacritics as a one text unit) for WI from handwritten Arabic text subword. Chapter three dealt with text segmentation to partition the text into its subwords with their diacritics. In chapter four, we tested the performance of various versions of subwords based writer identification schemes differing in the number of subwords used. Although, high accuracy rates were achieved, but few writers were responsible for most errors. By considering instances of such errors, we note that these could be results of the way some writers are not careful with the addition of the diacritics. In this chapter, we test the performance of a new version of our previously developed WI scheme by removing diacritics from subwords. We first illustrate the visual impact of diacritics on the structure of subwords and linkage to personal habits of writing.

### **5.1 Arabic Writer Identification based on Body of Subwords Alone**

Compulsory diacritics can appear as one dot, a pair of dots and sometimes triple dots. Writers sometimes combine these dots when they write them by hand in different ways and styles; this means each writer develops his own habit of writing all kind of strokes and diacritics.

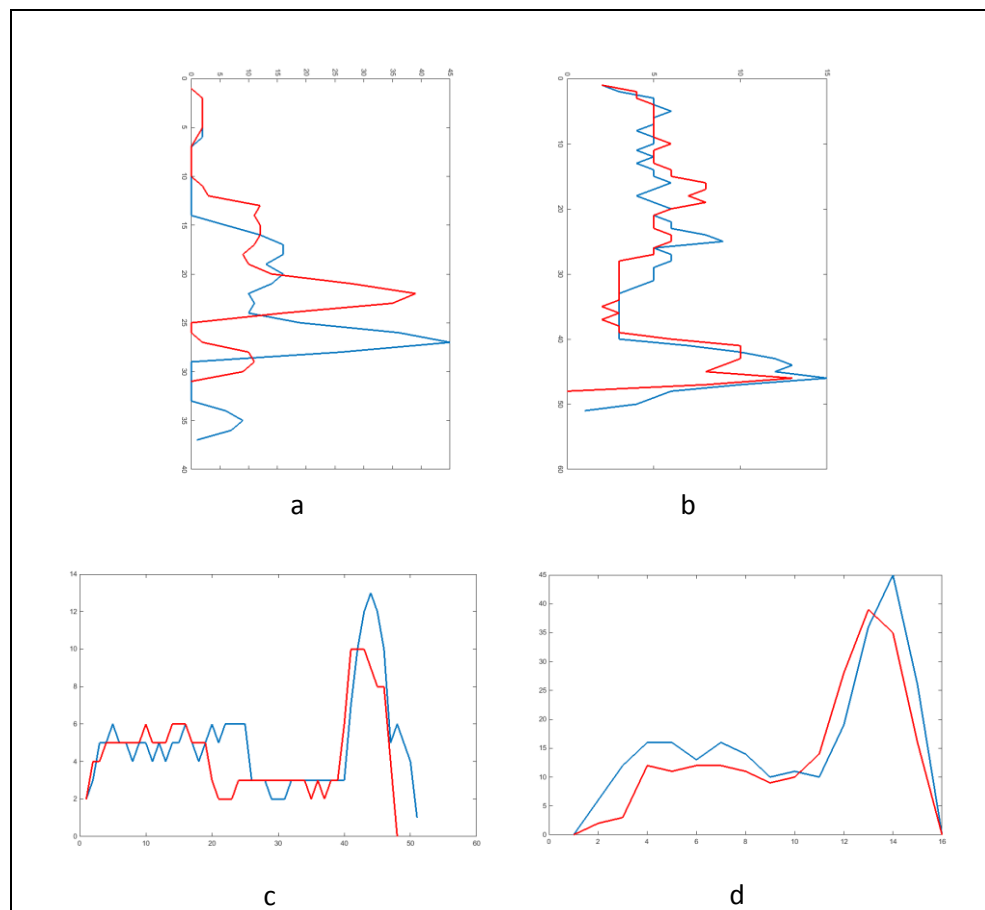
Usually, when a writer writes a word or a subword the first write the body and then adds its diacritics above and/or below the baseline of the subword. However, the positions of the diacritics vary from one writer to another, and even within the different repetition of the text by the same writer. Compared to printed text, the shape of diacritics in handwritten text is subject to slight random variations as well as appearing at variable distances from subword's baseline even for the same writer, which may lead to false identification. Figure 63 illustrates the differences between diacritics in similar subwords written by the same writer.





**Figure 63:** Variation in the position and Shape of diacritics of a subword written by the same writer

Figure 64 illustrate the projections profile for the same subwords written by the same writer (when their diacritics included). Figure 64a and c, are the horizontal and vertical projections prospectively for the subword in Figure 63a. Figure 64b and d are the horizontal and vertical projections prospectively for the subword in Figure 63 b.



**Figure 64:** Projections of subword in of Figure 63, before and after diacritic removing

These variations cause variations between the extracted features from different samples of written subword which in some cases could cause false rejection and /or false acceptance. Consequently, this could be one of the important factors having an adverse impact on the WI accuracy rate.

The above observations are the main incentive to investigate a modified version of the WI schemes developed in the last chapter using the body of the subwords only. The aim of the work done in section 5.3 below is to determine the impact of the addition of diacritics on the accuracy of the subwords based WI. This could also help settle the question about the presence of the Doddington Phenomena in our WI algorithms. Our expectation of getting more informed knowledge about these issues through the removal of diacritics stem from another general observation that we already made about Arabic text.

In chapter three we have noted that omitting diacritics in many words will increase the number of subword bodies to use when identifying a writer, as shown in Table 5.1. Thus, we can get more samples from a smaller text to be able to have a more credible assessment of our WI accuracy rate.

**Table 5.1:** Example of higher number of subwords repeated in dissimilar words when diacritics are omitted

في				
وفي	فيء	رقي	ساقى	في
وافى	قيء	رُقي	راقى	

## 5.2 Diacritics Removal from Subword Boxes

The segmentation process that we developed in chapter three and tested its performance, already separate the subword body from their diacritics. Therefore, we do not need a new segmentation algorithm to accommodate this facility. The process works as follows:

Step 1: Input a subword box

Step 2: Applying LCCA

Step 3: Find the image baseline

Step 4: The label (pattern) that lay on baseline will regards as subword's body.

Step 5: All other labels (patterns) will be discard as they regards as diacritics.

Our WI system, the performance of which based on subword bodies (when their diacritics are excluded) will be tested here using the best 14 WI discriminating subwords that have been used in the chapter 4, as shown in Table 5.2

Table 5.2: The selected 14 subwords without their diacritics

no	Subword including diacritics	Subword without including diacritics	No	Subword including diacritics	Subword without including diacritics
1	لكم	لكم	8	لي	لى
2	هي	هى	9	من	من
3	حليم	حلم	10	عن	عن
4	حمن	حمن	11	في	فى
5	لمحتر	لمحتر	12	على	على
6	بخير	بحر	13	لسيد	لسد
7	بسم	بسم	14	الله	الله

In chapter 3 we conducted a subjective test on the performance of the original subwords segmentation on different databases. However, the work in this chapter is primarily motivated by our interest in deciding if our WI scheme suffers from the Doddington Zoo effect or the shortcomings of the scheme in failing to recognize handwriting of the few “unknown/amateur” participants are due to extreme variation in the way these writers add diacritics to a subword's body. Hence we conducted a subjective testing on the outcome of the above segmentation for a randomly selected sample of handwritten subword text

for 10 writers (5 from the unknown group and 5 from the other groups) in the current database.

For each chosen writer and each of the 14 chosen subwords, we selected randomly 2 samples from the 5 training set and 4 samples from the 15 testing. In total, we examined 840 random samples. Table 5.3 shows the performance of the above segmentation for each of the two groups of writers (unknown and the known). Here the “known” include professional and normal writers, as categorized in chapter 4. It is clear that there is a clear association between failure to recognise the unknown writers and the lower segmentation accuracy achieved for the unknown in comparison to that of the known group. Interestingly, the excellent segmentation accuracy rate obtained for the known group is consistent with the accuracy of segmentation of subwords presented in Chapter 3 for 2 different databases (see Table 3.8)

**Table 5.3:** Performance of Segmentation algorithm

unknown writers		known writers	
no of success	Accuracy rate %	no of success	Accuracy rate %
75	89.28571	84	100
70	83.33333	79	94.04762
72	85.71429	84	100
72	85.71429	84	100
74	88.09524	83	98.80952
363	86.42857	414	98.57143

### 5.3 The Subword Body based WI Scheme

In this section, we present the performance of a refined version of the subword based WI scheme, whereby the diacritics of the subwords are removed. Each subword will be represented by the same 8-dimensional feature vectors that have been re-evaluated after the subwords are stripped of their diacritics. As for the number of subwords to be used for identification, we shall use two sets, the top 14 and the top 11.

We follow the same experimental protocols used in chapter 4, i.e. randomly select 5 samples per writer for training and the other 15 samples for testing, where a sample here consists of a handwritten copy of each of the above 14 subwords. Each subword is represented by the 8 optimal feature vector  $[f_4, f_2, f_3, f_7, f_1, f_6, f_{14}, f_{15}]$  obtain by the incremental procedure of chapter 4, section 4.3.3. Matching is based on the nearest

neighbour using the distance function. As before, we present two different sets of results depending on the basis of writer identification:

**Case 1:** Identification is based on one subword. In this case for each writer success is declared if the majority of the 15 testing samples return the accurate writer.

**Case 2:** Identification is based on a full document of the 14 subwords. In this case for each writer success is declared if the majority of the 15 testing document return the accurate writer.

**In case 1**, the diagram in Figure 61 illustrates the structure of the testing. Moreover, the flowchart given in Figure 62 helps calculate the number of successful test and accuracy for the above experimental cases above.

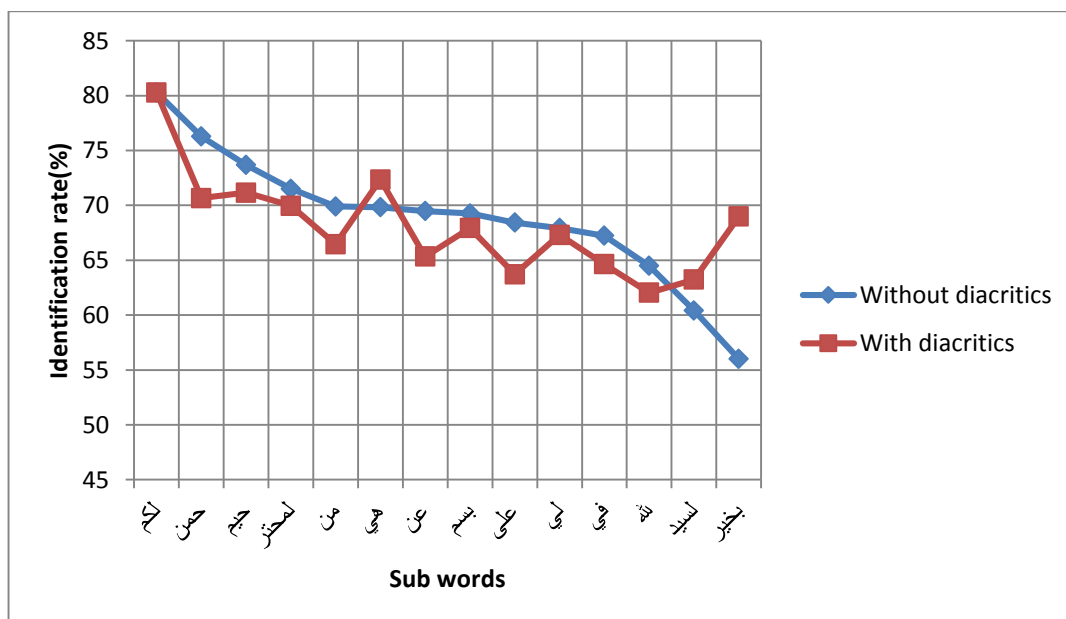
Table 5.4, below, gives the identification accuracy rates based on each single subword (with and without diacritics) of the list of 14 subwords. For each subword, a writer is identified if the majority of his/her 15 written samples of subword have been matched to the nearest neighbour. In this experiment, the total number of single subwords tests is  $19950=14*1425$ , the total number of accurately matched writers for all the 14 subwords is 13747. The accuracy rates for the 14 subwords are listed in the descending order of accuracy of the without diacritics scheme. Note that the order of accuracy of these subwords does not match the order of accuracy of the 14 subwords as listed in Table 5.2, These results further confirm the validity of our hypothesis that subwords in handwritten Arabic text have WI discriminating power. The overall identification accuracy achieved when using all subwords list can be calculated as follows: All try subword list = 100 \*  $(13747 / 19950 )$

$$= \mathbf{68.90727\%}$$

**Table 5.4:** subwords success hit in order (from high to low) using all writers

no	Number of Samples	subword	success match	Without diacritics Accuracy	With diacritics Accuracy
1	1425	لكم	1144	80.2807	80.2807
2	1425	حمص	1087	76.2807	70.66667
3	1425	حلم	1050	73.68421	71.15789
4	1425	لمحصر	1019	71.50877	69.96491
5	1425	من	996	69.89474	66.45614
6	1425	هي	995	69.82456	72.35088
7	1425	عن	990	69.47368	65.33333
8	1425	نسم	987	69.26316	67.92982
9	1425	على	975	68.42105	63.7193
10	1425	لى	968	67.92982	67.29825
11	1425	فى	958	67.22807	64.63158
12	1425	لله	919	64.49123	62.03509
13	1425	لسدد	861	60.42105	63.22807
14	1425	لحصر	798	56	68.98246
<b>Total 14 subword samples</b>	<b>19950</b>	<b>Total Match</b>	<b>13747</b>		
<b>Accuracy rates</b>				<b>68.90727</b>	<b>68.14536</b>
<b>Total 11 subwords samples</b>	<b>15675</b>	<b>Total Match</b>	<b>11169</b>		
<b>Accuracy rates</b>				<b>71.25359</b>	<b>69.55</b>

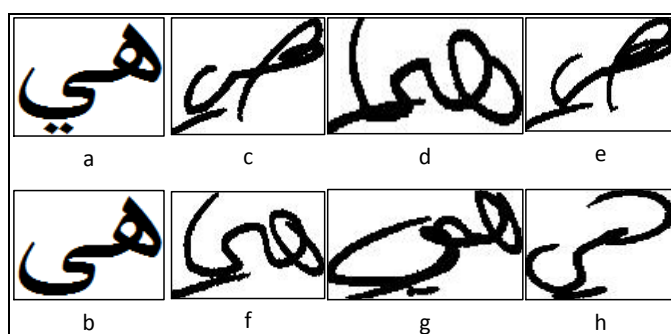
The chart below illustrates the results in the table above and helps in highlighting the change in the pattern of accuracy when diacritics are removing.



**Figure 65:** results that illustrate in table above (Table 5.4)

The results in an above chart (Figure 65) clearly show that for all but 3 subwords removing diacritics from subwords enhances their WI discriminating characteristics. In most cases, the improvement is the significance, e.g. for subword 2 (حمل) accuracy increased by about 5.5%. On the other hand, the discriminating characteristics of subwords هي, لَسِد and حَر deteriorates as a result of removing their diacritics by about 3% for the first two and as much as 13% for subword بخير.

To explain this variation in the effect of diacritics removal on accuracy, we need to have a close look at the failure patterns. For the subword, Figure 66 below illustrates the difficulty some of the possible problems associated with removing the diacritics from the subword هي. The first column shows the printed version of the subword before and after removing diacritics. While the other columns show examples of handwritten versions of the same subword by different people that highlight potential difficulty with segmentation due to the overlap of diacritics with the subword body. The lower table (Table 5.5) displays the same subword written by 3 different writers before and after segmentation. It shows that segmentation destroyed the subword for the first two writers but did not have any effect for the last one.



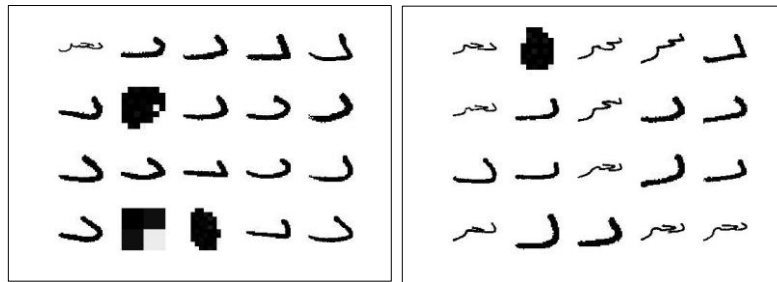
**Figure 66:** some of the possible problems associated with removing the diacritics from the subword هي

**Table 5.5:** subword هي with different writers before and after segmentation as presented in Figure 66

Pre-segmentation	Post segmentation

These examples may provide some explanation as to why the accuracy for the هي subword has dropped by about 3% after removing the diacritics. To explain the huge loss of accuracy for the بخير subword, we examined its segmentation for the 10 randomly selected writers (5 unknown and 5 known). We found that almost all faulty segmentation was coming from the unknown group of the writer. Figure 67 shows the large number of failed segmentation of بخير for two different writers from the unknown group. Recall that each writer has 20 samples. These two examples illustrate that false rejection and false acceptance will be more common when we use this subword, without its diacritics, for WI. Perhaps we can use the success or failure of diacritic segmentation to exclude or include subwords from WI scheme.

Finally, the fact that accuracy improved for all other 11 subwords indicates that the failure of segmentation cannot take the full blame for the reduction in accuracy for these 3 subwords.



**Figure 67:** Faulty segmentation of the subword بخير for 2 different writers

In case 2, this experiment the results are based on taking each of the 15 documents containing all the 14 selected subwords bodies (without their diacritics) as one testing trial and identification is based on the subwords (i.e. more than 8) being matched by the same writer.

Table 5.6 lists the number of successfully matched documents for each of the 95 writers in descending order of success. The results in this table first give the accuracy rate for writer identification is given by:

$$\begin{aligned}
 &= 100 * \left( \frac{\sum_{wr=1}^n \text{success samples}}{\text{no of writers} * \text{no of testing samples}} \right) \\
 &= 100 * (1366 / (95 * 15)) \\
 &= \mathbf{95.86 \%}
 \end{aligned}$$



These results leave no grounds for doubting the validity of our hypothesis that subwords in the handwritten Arabic text have WI discriminating power. Furthermore, the WI discriminating of subwords are mostly improved when their diacritics are removed. The examples of failed diacritics segmentation shown above indicate that identification errors are almost certainly due to the inability of our segmentation scheme to deal with the case where diacritics are connected to their subwords. But this seems to be one the case with some writers has such a habit for some subwords. Perhaps one can design a system that detects such cases and either corrects it or incorporated into the writer profile.

**Table 5.6:** Success samples of all writers (out of 15 in each sample) in high to low order

no	Wr no	succ sample	No	Wr no	succ sample	no	Wr no	succ sample	no	Wr no	succ sample
1	1	15	25	41	15	49	71	15	73	22	14
2	2	15	26	42	15	50	73	15	74	30	14
3	3	15	27	43	15	51	74	15	75	36	14
4	4	15	28	45	15	52	75	15	76	37	14
5	5	15	29	47	15	53	76	15	77	72	14
6	7	15	30	48	15	54	79	15	78	77	14
7	9	15	31	50	15	55	81	15	79	80	14
8	10	15	32	51	15	56	82	15	80	6	13
9	11	15	33	52	15	57	84	15	81	8	13
10	13	15	34	53	15	58	85	15	82	15	13
11	14	15	35	55	15	59	86	15	83	16	13
12	18	15	36	57	15	60	89	15	84	17	13
13	19	15	37	58	15	61	90	15	85	49	13
14	23	15	38	60	15	62	91	15	86	59	13
15	26	15	39	61	15	63	92	15	87	78	13
16	28	15	40	62	15	64	93	15	88	88	13
17	29	15	41	63	15	65	94	15	89	24	12
18	31	15	42	64	15	66	95	15	90	54	12
19	32	15	43	65	15	67	97	15	91	25	11
20	33	15	44	66	15	68	98	15	92	46	11
21	34	15	45	67	15	69	99	15	93	27	10
22	35	15	46	68	15	70	12	14	94	44	10
23	38	15	47	69	15	71	20	14	95	39	8
24	40	15	48	70	15	72	21	14	<b>Sum→</b>		<b>1366</b>

The results in Table 5.6 is also interesting in that it settles the question of the presence of the Doddington Zoo phenomena at least for similar cases where the subwords in the gallery are the same that appear in test samples as in the case of the Al-Madeed et al. text-

dependent database. First of all 100% of the writers were identified by a majority of their handwritten samples. In relation to the writer classification defined in chapter 4, 95.86% of the writers in the database are recognised by our latest WI scheme to be professional writers, and 4.14% are recognised as normal writers, and all writers are known to the scheme.

Finally, we can summarize, the results of this and last chapter in the following table (Table 5.7) to highlight the achievements of our investigation in comparison to existing work in this area. We note that these results significantly outperform the word-based WI scheme of Al-Madeed et al (Al-Ma'adeed, et al., 2008) which has achieved accuracy of 90% but at nearest neighbour rank of top10 while all our results are rank 1 accuracy.

**Table 5.7:** system accuracy in summary

No	With/without diacritics	Method	Subwords majority success rate %	Samples majority success rate %	Writer majority success rate %
1	<b>Subwords WITH diacritics</b>	<b>All writers + all Subwords</b>	<b>63.63</b>	<b>82.60</b>	<b>90.53</b>
2		<b>All writers + 11 Subwords</b>	<b>69.55</b>	<b>84.07</b>	<b>92.63</b>
3		<b>Success writer + all Subwords</b>	<b>66.89</b>	<b>90.31</b>	<b>100</b>
4		<b>Success writer + 11 Subwords</b>	<b>72.85</b>	<b>90.70</b>	<b>100</b>
5	<b>Subwords WITHOUT diacritics</b>	<b>All writers + all Subwords</b>	<b>68.90727</b>	<b>95.86</b>	<b>100</b>
6		<b>All writers + 11 Subwords</b>	<b>71.25359</b>	<b>98.12</b>	<b>100</b>

## 5.4 Summary and Conclusions

Subword this chapter we refined our previous subwords based WI scheme by using subwords bodies without their diacritics. This was based on observations that the inclusion of the diacritics results in significant variation in the way writers place and shape their diacritics around their subwords bodies which in turn cause variation in the extracted feature vectors.

Our diacritics removal segmentation schemes achieved high accuracy with most subwords and most writers in the experimental database that was consistent with what was achieved earlier with two other databases in chapter 3. The performance of segmentation scheme, however, highlighted possible impact on the performance of our WI scheme that could be manifested as increased incidents of false rejections and/or false acceptances, particularly for the group of writers classified in chapter 4 as “unknown”.

The experimental results demonstrated that removing the diacritics has led to significant improvement in the performance of subwords based WI from Arabic handwritten text. More importantly, we have negatively answered the question raised in chapter 4 regarding the Doddington Zoo phenomena. In fact, we have shown, beyond any doubt, that there is no evidence of this phenomena and what we have detected in chapter 4 was due to variation in the way diacritics are added to subwords body. The removal of diacritics provided the ultimate refinement of our subword based WI scheme.

In the next chapter, we should try to investigate the performance of our subword based WI scheme on scenarios when the text samples presented for matching are not available in the gallery. This is a challenging scenario but the most realistic scenario for practical writer identification.

## **Chapter 6 : WI from Text-Independent Handwritten Arabic Texts**

In the last two chapters, we developed a variety of subword WI for handwritten Arabic text schemes. In all these schemes 15-dimensional feature vector representations of subwords (with or without their diacritics) and defined two similarity/distance functions to be used for writer identification. We tested the performance of 3 versions of the developed WI scheme on a database consisting of 20 samples of handwritten texts for 95 writers; each sample is made up of 27 selected words (a total of 49 subwords). We achieved a high accuracy rate for all 3 versions, but the “subwords without diacritics” version was of significantly higher accuracy. However, in reality the more interesting scenarios for testing performance of a WI scheme is when we need to identify the writer of a text when another sample of this text is not stored in the gallery, and we may or may not know if the person is in the database. When an attempt is made to match the input text to any stored text for a writer in the system, we do not expect to see the same subwords, or even more than few common subwords between the two texts. We call this WI from Independent text. This fits more the current needs for security applications. Here we shall investigate the suitability of our subword (with and without the diacritics) based WI scheme. We first test the performance of ours on an in-house in text-dependent database of modest size containing two different handwritten Arabic texts for 50 writers. We shall also simulate text-independent WI testing using Al-Madeed et al. database. We close this chapter by discussing possible complementing the subword based WI scheme by using global and local features extracted from the scanned image of independent text.

### **6.1 WI from Text-Independent Data**

The process of WI which we have previously followed in the text-dependent DB, though useful in identifying best features and best subwords, is different when we attempt to test its performance on in text-dependent DB. Testing subwords based WI schemes on the text-dependent database is made more straightforward by the fact that the testing and training samples contain the same copies of the subwords. In real life applications such as in Forensics, often you attempt to recognize a person from a short text, and you may or may not have the same text handwritten by that person. The problem, in this case that there may only be very few words in common between a fresh text and stored texts.

However, subword based WI in this case should have a much better chance of success than word based WI due to the fact that the probability of finding common subwords between different texts is definitely higher than finding common words. However, unlike the case in the last two chapters, we do not have the luxury of using best WI discriminating from a sufficient number of subwords but should even be prepared to use single character subwords which are more common between different texts. Here we shall slightly adjust our subword base WI scheme to take into account this fact, and to be used to test whether two different texts were written by the same writer.

### **6.1.1 The Adjusted Subword based WI Scheme for Independent Texts**

To accommodate the testing of identical authorship of a new sample text and existing stored different text, we assume that all subwords of the stored texts are segmented, and their 8-dimensional feature vectors are extracted. Therefore, each stored text is represented by a list L indexed by its subwords (with and without their diacritics) and each record consist of the feature vectors of its index subword. To adjust the subword based WI scheme we simply apply the following pre-processing steps:

1. Segment the fresh input text into its subwords, as presented in 3.5.4
2. Identify the list of shared subwords between the fresh list and the list L for the given stored template text.
3. Extract the 8-dimensional feature vectors for each appearance of the shared subwords in texts.

Note that, the feature vectors for a subword with its diacritics is different from when we take subword body without diacritics. Either is used depending whether we are testing the performance of the WI scheme for subwords with diacritics or subwords without. We shall now embark on testing the performance of this adjusted version of our WI scheme

### **6.1.2 The in-House in text-dependent database**

For the purpose of testing the performance of the above-adjusted subword based WI scheme, we used the in-house in text-dependent DB described in detail in Chapter 2. The following is a summary of its characteristics.

1. The DB is made of two different texts (see Figure 12 and Figure 13) written by 50 random writers.

2. Notable differences between each text ranges from the length of the text, the colour of the ink used, the type of paper used on, the time spend between each writing and the subject of the text.
3. The average length of each text is 46 words.
4. The system segmented 110 subwords as an average per text.

Table 6.1 list all the subwords shared between the two texts samples in the database together with the number of time each appears in both texts.

**Table 6.1:** Subwords matching (within text) and between both texts

No	Repeated in Text 1	Repeated in Text 2	Subwords WITH diacritics
1	32	19	أ
2	8	2	ن
3	5	4	و
4	2	2	م
5	1	2	في
6	1	1	لي
7	1	1	لا
8	1	2	من
9	1	2	س
10	1	1	ين
11	1	1	ف

Note that all subwords are of length  $\leq 2$ .

### 6.1.3 WI Based on Subword with and without Their Diacritics

In this section we report the results of 2 similar experiments, to test the performance of the adjusted subword WI scheme, the only difference between them is that in the first we use subwords with their diacritics while the second the diacritics are removed. For

repeated subwords, we shall pool together the results of matching each copy of the input subword against all appearances of the subword in the stored template.

For matching we use the nearest neighbour using the Euclidian distance to all the 8 features including the projection ones where we append the shorter projection sequences by sufficient 0's as necessary to have the same length of the other. We shall first conduct experiments to determine the WI discriminating power of the shared subwords with and without diacritics.

### **Individual subwords writer identification**

In these experiments, for each of the 50 writers and each of 11 subwords, listed in Table 6.1, we input to:

**The Gallery** for each appearance of a subword (s) in text 1, a copy of the 8-dimensional feature vector representing that copy of (s).

Therefore, the gallery of the with diacritics experiment contains

$$(32+8+5+2+7*1) \times 50 = 54 \times 50 = 2700$$

Samples of the 8-dimensional feature.

Similarly, the gallery for the without diacritics experiments contains 2700 samples

**The Test set** for each appearance of a subword s in text 2, a copy of the 8-dimensional feature vector representing that copy of (s).

Therefore, the gallery of the with diacritics experiment contains


$$(19+2+4+2++2 +2+2+4*1) \times 50 = 37 \times 50 = 1850$$

Samples of 8-dimensional feature vectors.

Similarly, the gallery for the without diacritics experiments contains 1850 samples.

Individual subword accuracy

Our first pair of experiments were aimed to test the performance of the two versions of our WI scheme (the with and the without diacritics schemes), whereby identification is based on a single subword. In this case for a subword s, success of identifying the writer is declared for that subword if the majority of the r(s) of its testing samples return the accurate writer. Here r(s) is the number of time s appears in text 2, and a test sample for s returns the writer of the nearest training samples for s.

To illustrate, this criterion, consider the subword  which appears 4 times in text 2. Each writer is accurately identified as long as 3 of its samples in text 2, its nearest neighbour among its 5\*50=250 gallery samples is written by the same writer. While for the subword

أ which appears 19 times in text 2, a writer is accurately identified as long as for 10 or more of the 19 samples, its nearest neighbour among its  $32*50=1800$  gallery samples is written by the same writer.

Table 6.2, below, shows the accuracy rate for each subword for both cases (with and without their diacritics).

**Table 6.2:** Identification rate (%) for individual subword (with and without their diacritics)

No	Subword with diacritics.	Accuracy (%)	Subword without diacritics	Accuracy (%)
1	لي	62.897	لى	66.597
2	من	60.367	مس	64.067
3	في	54.837	فى	60.537
4	ين	52.817	س	59.517
5	لا	61.127	لا	62.827
6	ق	52.017	ق	55.717
7	م	57.494	م	58.166
8	ل	57.227	ل	58.395
9	ن	39.757	ن	43.457
10	و	40.907	و	41.879
11	أ	37.997	ا	39.097

The results in this table confirm that subwords without their diacritics perform better in discriminating writers than when their diacritics are included. As expected the length 2 subwords are more writer discriminating.

### **Whole text writer identification experiments**

In these experiments, the gallery contains 1 text file for each of the 50 writers and the testing set containing 1 text file (different from that in the gallery) for each of the 50



writers. Each text file contains the feature vectors for all the subwords extracted from the handwritten text. In these experiments, we shall only be interested in the feature vectors extracted from the subwords without diacritics. This decision is based on the results of the last section.

Identification of the writer of an input test text file depends on the number of successful subwords for the writer, out of the 11 shared subwords as listed in 6.1. Since there are repeated subwords, then we need to make sure that this will not lead to any bias in our experiments. For example, we have 19 shared appearances of the subword  $\hat{f}$ , whereas the other 10 subwords have only 15 appearances altogether. This means that even if all other subwords have full successful writer identification then for a majority rule we still need at least 3 successful identification of the writer from the  $\hat{f}$ . Note that the  $\hat{f}$  is the least writer discriminating subword. In order to test such a bias does exist, we conducted the first experiment to determine the accuracy of the subwords body based WI scheme on our intext-dependent database whereby the writer of an input test will be determined by the majority voting of all the subwords with their repetitions.

Table 6.3 shows accuracy rates for top k ranks,  $k=1, 2, \dots, 10$ , using the nearest neighbor classifier.

**Table 6.3:** Writer identification based on majority of subwords in text

Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10
61.44	64.71	67.32	74.51	76.47	88.24	88.89	93.33	95.06	98.04

The results here indicate that rank 1 accuracy of identification is lower than accuracy achieved by each of the 3 top writer discriminating subwords and this is a strong indicator that the majority rule when repeated subwords are considered as distinct subwords is biased towards the more frequently repeated subwords.

In order to rebalance this bias, we repeated the experiments but this time we represent each of the 11 subwords will be represented by the average feature vector of all the feature vectors extracted from the its repeated versions. This will be done for both gallery texts and the testing texts. Table 6.4 displays the accuracy rates for top k ranks,  $k=1, 2, \dots, 10$ .using the nearest neighbour classifier. Identification will be based on majority rule, i.e. if 6 or more voted for a single writer.

**Table 6.4:** Writer identification based on (Average of samples algorithm)

Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10
66.67	70.42	76.67	80.83	85.83	88.33	91.25	95.42	96.72	100

The rank 1 accuracy rate, in this case, is only marginally above the accuracy of the top performing subwords. In fact, we may need to include top 3 rank before we notice the contributions of the top 3 writer discriminating subwords. From the previous two chapters, we found that some subwords have more discriminating characteristics than others, and thus it's only sensible to give more weight to high performing subwords than to others. For this, we designed the following weighting map table having experimented with few other maps. The map is shown in Table 6.5, below, was also influenced by our earlier observation that many one character subwords lead to more instances of false rejections and false acceptances.

**Table 6.5:** Subwords weighting map

Subword group	Subword	individual weight
First	من, لي	0.375
Second	ين, لا, في	0.070
Third	ق, ل, م	0.013
fourth	و, ن, أ	0.000

Our last experiment is based on using this weighting map to on top of the feature vectors averaging approach used in the last experiments. Therefore, in this case, for each subword we calculate the weighted Euclidean distances between the input average feature vector and all the corresponding stored average feature vectors for the 50 writers in the database. The identity of the subword writer will be that of the nearest neighbour for rank 1. Finally, the identity text writer will be decided by the strong majority rule which stipulates that the successful writer must be identified by 6 or more of the shared subwords. For rank  $k > 1$ , we assume that the writer is identified within the top  $k$  nearest neighbours for 6 or more of the shared subwords. Table 6.6 shows the accuracy rates achieved in this experiment.

**Table 6.6:** Writer identification based on (Weighted subword algorithm)

Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10
71.24	75.16	80.39	86.27	90.20	94.77	100.00	100.00	100.00	100.00

These results are more encouraging and demonstrate the validity of our subwords hypothesis for the independent text scenario, i.e. the writer discriminating power of subwords in the independent text matching are significantly higher than that of words. Moreover these results convey an important message about the importance of using subwords of length  $>1$  and the importance of using a weighting map that could be learnt from the dependent testing scenario to a multi-subwords fusion schemes.

It is worth noting, that the majority rule applied in this chapter, and the thesis, is somewhat strong version of what is practiced in traditional identity verification systems, whereby the claimed identity of feature vector is verified as true if it is matched to the templates the same person more than to others even if it doesn't get  $> 50\%$  of the votes. Therefore, the errors from our practiced majority rule include the "Don't Know" decisions. We are confident that the performance can be improved if we to relax our majority rule to the traditional approach of verification, but there is no clear indication that such improvement would be significant enough. Instead, we need to try to further investigate the reasons for not achieving higher accuracy for text-independent scenarios. In the next section, we attempt to ascertain if the shortcoming can be attributed to the quality of the database.

## **6.2 Simulating Subset Testing (Text-Independent from a Text-dependent database)**

By comparison to the case of text-dependent database experiments, the accuracy achieved in the last experiment is still well below what is desirable and it may be unrealistic to recommend the use of the last refined subword based WI scheme for independent text applications that require high level of accuracy without either complementing it with other schemes or refining it further. In order to consider any further refinement, we need to determine the influencing factors that led to this undesirable result. First of all we need to see how much these results were influenced by the choice of the rather difficult in text-dependent database which only contained 2 text samples per writer and these samples only share a few short subwords. It is certainly unrealistic and time-consuming to construct a new larger database, for testing purposes, with several text samples that share several common subwords. However, we can simulate the effect of constructing such an

in text-dependent database using any text-dependent database. This is due to the fact, that for our tests we only need the shared list of subwords, and their extracted feature vectors, that would have been segmented from the text documents and nothing else from these documents.

The Al-Madeed et al. database provides an excellent choice for use in our simulation which is aimed to answer the above question. It consists of 20 sample handwritten documents for 95 persons and, each consisting of 49 subwords. Therefore, we can simulate a variety of text-independent test simply by randomly selecting a list of subwords from the 49 subwords and treating the 20 samples per writer as if they were 20 different documents that only shared the given list of subwords. In this way, we avoid the costly effort of building a large in text-dependent database without affecting the results of the experiments. Some people may naively argue that in this way the experiments we conducted in Chapters 4 & 5, when we only used 11 and 22 subwords out of all 49, meet the requirements of text-independent experiments but these were not selected randomly and it is not realistic to expect the presence of such a large list of high writer discriminating power shared among 20 documents.

We have conducted two sets of simulated experiments, using subwords without diacritic randomly selected from the Al-Madeed et al. list of subwords. Each set of experiments consisting the running of the algorithm 5 times. The input to each running in the first set of experiments consisted of 5 randomly selected subwords from the best 11 discriminating subwords displayed in Table 4.6. As before 5 of the 20 sample documents were used as templates in the gallery, and the other 15 were used for testing. Table 6.7, below, show the accuracy achieved by all the 95 writers when each document was recorded for the writer when 3 or more subwords identified him/her.

**Table 6.7:** Simulated WI using 5 random subwords

group	subwords group	succ hits	succ sample %
1	في   عن   بسم   حيم   لكم	1147	80.49
2	في   من   بخير   الي   لمحتر	1088	76.35
3	عن   من   بخير   لكم   لمحتر	1157	81.19
4	بخير   حيم   حمن   لكم   هي	1188	83.37
5	عن   لكم   حيم   بسم   لمحتر	1186	83.23
<b>Average</b>		<b>1153.2</b>	<b>80.93</b>

The second set of experiments is similar to the first but this time every run start by inputting 9 randomly selected subwords from the best 22 discriminating subwords displayed in Table 4.6. Again 5 of the 20 sample documents were used as templates in the gallery, and the other 15 were used for testing. Table 6.8, below, show the accuracy achieved by all the 95 writers when each document was recorded for the writer when 5 or more subwords identified him/her.

**Table 6.8:** Simulated WI using 9 random subwords

group	subwords group	succ hits	succ sample %
1	في   قيع   بسم   لكم   السيد   من   بخير   احمن   هي	1166	81.82
2	على   لي   بسم   لكم   السيد   المحتر   بخير   احيم   بعد	1132	79.44
3	قيع   على   لر   السيد   المحتر   بخير   لكم   في   الله	1098	77.05
4	جزا   عن   بسم   السيد   المحتر   بخير   لكم   الله   هي	1161	81.47
5	عن   المحتر   بخير   من   هي   في   الي   احمن   احيم	1162	81.54
<b>Average</b>		<b>1143.8</b>	<b>80.27</b>

The results from those two sets of experiments demonstrate a significantly improved performance of the WI scheme over the accuracy that we got from the real in text-dependent database. It is natural to attribute this success to the fact that in those two cases the selected subwords only included those of length 2 or 3. We also observe that increasing the number of shared subwords does not necessarily increase the performance of the scheme. This observation needs to be validated with more repetition of the simulated experiments with different sets of subwords. However, we should advise operators of such system to qualify the output identification decisions when testing the authorship of multiple different texts by considering the number and the mix of length among the shared subwords. In fact, the simulation approach seems to provide an efficient and reliable alternative to distinguishing text-dependent scenarios from text-independent ones. Perhaps more efforts need to collect more handwritten subwords for writers than building huge databases by collecting different texts from large populations.

Taking into account the effect of the strict majority rule and the fact that in many cases we are only using a relatively small portion of the input texts, one might be satisfied with accuracy rates of around 80% for text-independent writer identification. However, more investigations would be needed using additional information to optimize WI from text-independent texts. In the next section, we shall discuss few possible areas for future investigations in this respect.

### **6.3 How to Improve Writer Identification for text-Independent Scenario**

The results of the two experiments for WI from the text-independent text, though excellent, raises a number of possible avenues to complement the subword based schemes. The first question one might ask what any modifications one could make on the subword scheme? In this respect, one might consider extracting more features from subwords, or even combining and expanding features from subwords that form words in the text. For example the vertical/horizontal gaps between subwords belonging to the same words, projections in directions other than horizontal or vertical, and changes in the slope of their baselines.

The last set of suggestions are reasonable grounds to consider the use features from other text components such as line, paragraphs and pages when we deal with text-independent writer identification. In fact, such investigations could benefit from digital image features used general pattern recognition schemes and camera based biometric verifications. Note that automatic WI in the text-independent scenarios process and analyse scanned images of paper written texts. In the rest of this section we shall describe a number of such image-based features that we consider relevant to future investigations, which can be extracted from the sub-images of various text components including subwords, words, lines and paragraphs.

#### **Using ZigZag feature for a subword**

The zigzag scanning pattern for run-length coding of the quantized DCT coefficients was established in the original MPEG standard. We borrow this concept to encode a subword in such a way that can encapsulate the pattern created by the writing of the subword text in the box. If the box of a scanned subword image, considered as a binary matrix, then the zigzag representation of the subword can be extracted as a binary string of a fixed length binary string. Hamming distance of binary strings can be used to measure the similarity between two input subwords. However, one needs to overcome the variation in size of subword's image boxes. One possible solution is to normalize the box size, or better divide the box image into a fixed number of cells and each cell can be represented by 1 if the number of printed pixels is above a certain threshold. The Zigzag feature can be used to represents the entire scanned text page or a rectangular image block. Future

investigation should include determining the distribution of hamming distances and selecting an appropriate classifier.

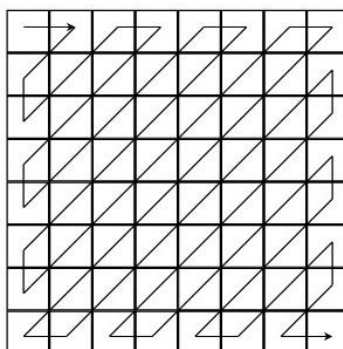


Figure 68: image zigzag (Pantech, 2014)

### The LBP feature of text block/subword

The LBP is a widely used method to represent texture in an image or an image block. It has many different versions. The most common simple version replaces each image pixel value by an 8-bit byte formed by comparing the pixel values in its 3x3 neighbourhood in a clockwise manner starting from the top left corner. The procedure assigns a 0 to each pixel on the boundary of the 3x3 window if the pixel value is less than the central pixel, 1 otherwise. The LBP image encapsulates the texture in the original image that is commonly used in pattern recognition, a 256-bins (or better well-defined 59-bins) histogram of the LBP image has been used as a feature vector for face recognition. The 59-bins LBP histogram includes 58 uniform patterns (binary patterns containing at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is traversed circularly) and 1 non-uniform for all other patterns. The LBP concept is illustrated in the figure below. For more information (Ojala, et al., 2002).

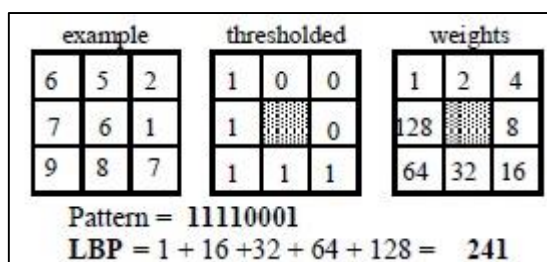
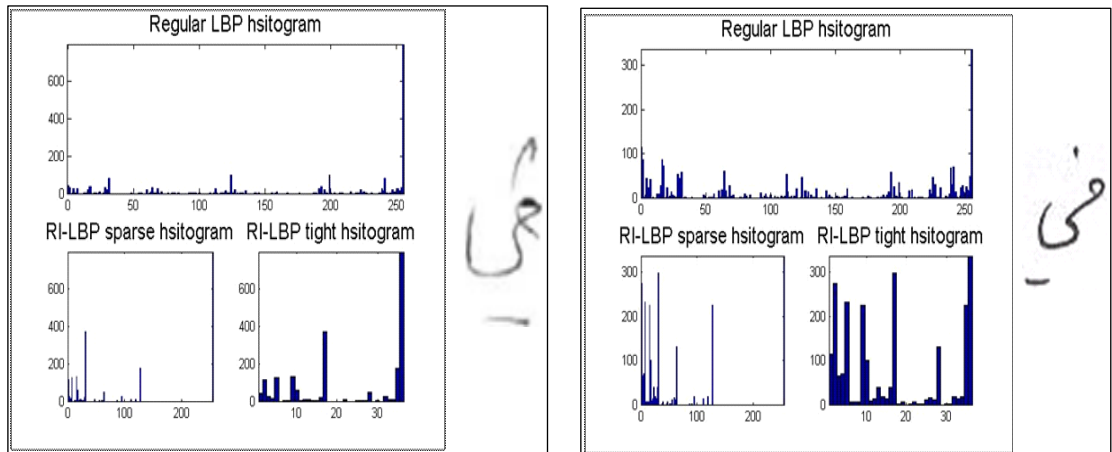
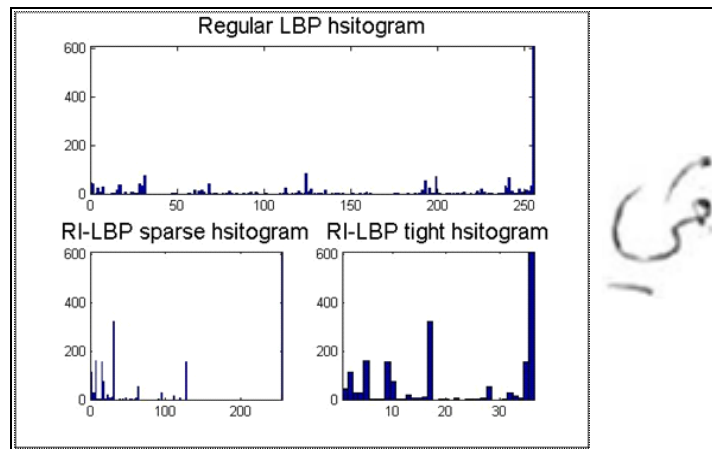


Figure 69: example LBP algorithm (Ojala, et al., 2002)

Below are 3 examples of LBP of the subword (في) written by 2 different writers. From these examples, one can see that histogram intersection can provide an appropriate distance function necessary for writer identification.



**Figure 70:** LBP results of Left: writer 1 subword 1 and Right writer 2 subword 1.



**Figure 71:** LBP results of writer 1: subword 2

### **Gabor filter feature for text line or text document**

Gabor wavelet transforms is another widely used tool for image texture analysis in different directions. It is capable of multi-resolutions and multi-orientation analysis of images and it is suitable for pattern recognition due to its distortion tolerance characteristics (Gdyczynski, et al., 2014). Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. A set of Gabor



filters with different frequencies and orientations are usually for extracting useful features from an image. Gabor filters have been widely used in pattern analysis applications. (Kohonen, 1982). The scanned images of text written by different writers seem to have settled differences in local texture orientation. Indeed Gabor filters have been used analysis of text line or/and text document, (see: Marti et al. (Marti, et al., 2001), Hertel and Bunke. (Hertel & Bunke, 2003), Rafiee et al (Rafiee & Motavalli, 2007)). Below are examples of applying Gabor filter to 2 handwritten Arabic text paragraphs. In these examples, the lower left side pattern are the actual Gabor wavelet atom filters while the right side boxes represent the responses obtained from applying these filters to the corresponding scanned text images

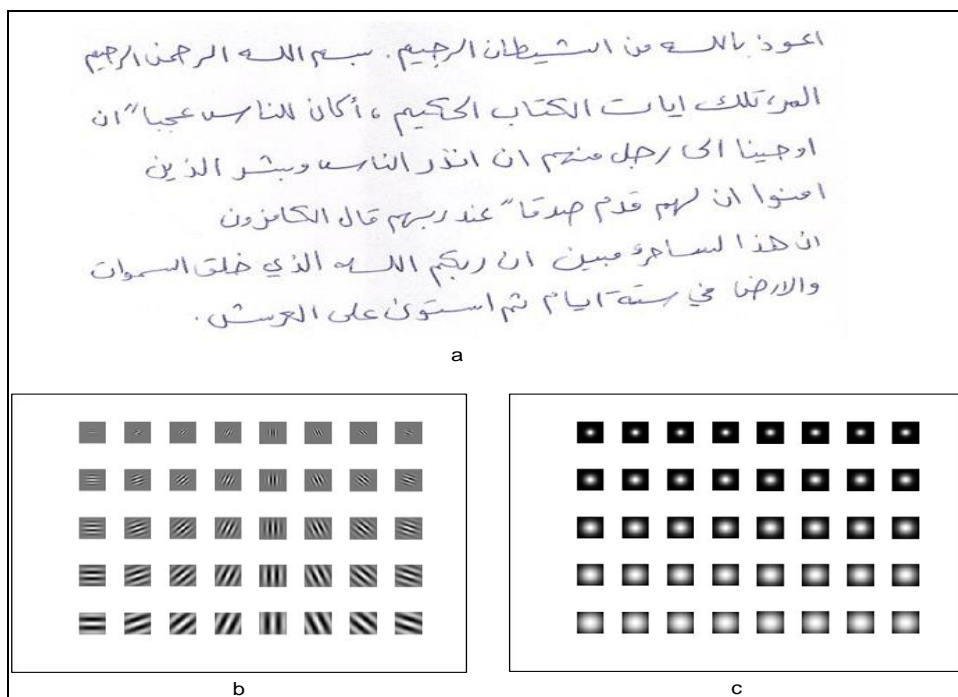


Figure 72: Apply Gabor filter for text1 writer1

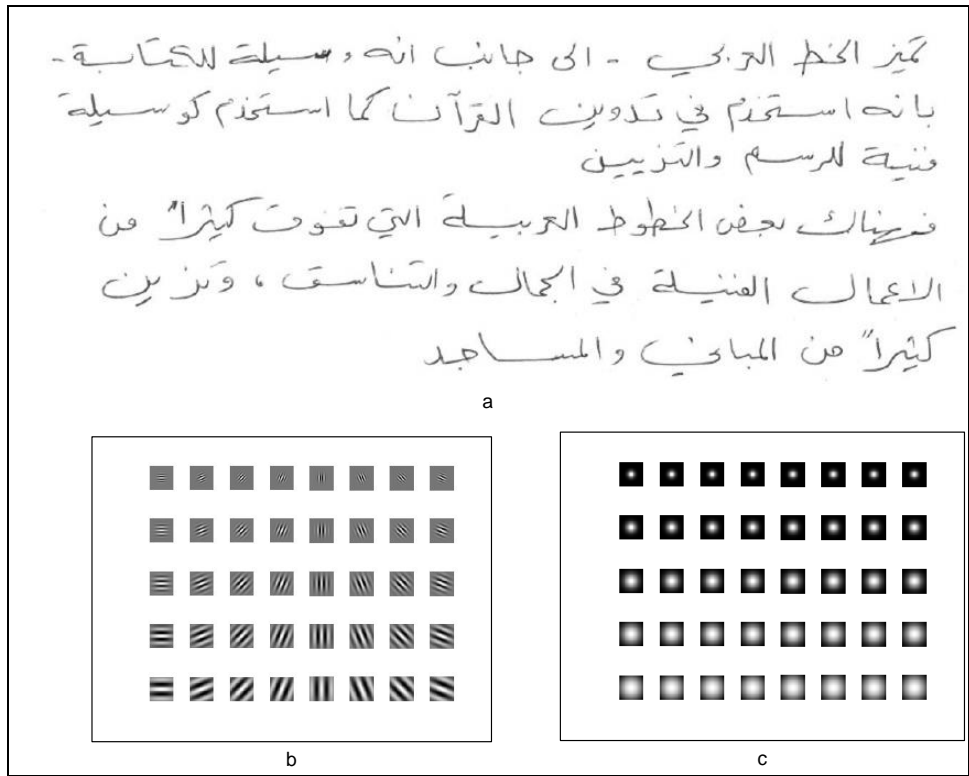


Figure 73: Apply Gabor filter for text2 writer1

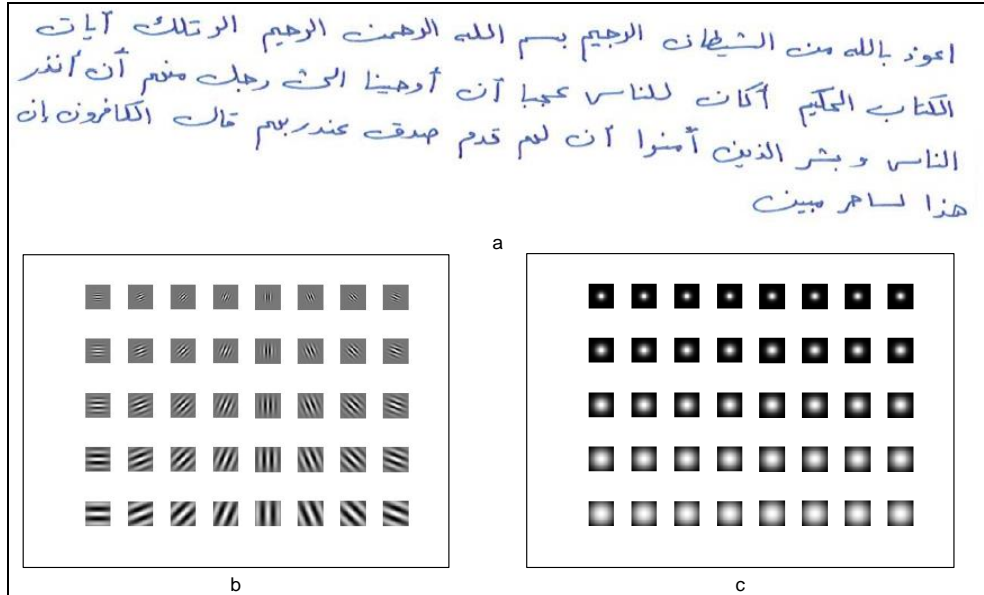


Figure 74: Apply Gabor filter for text1 writer2

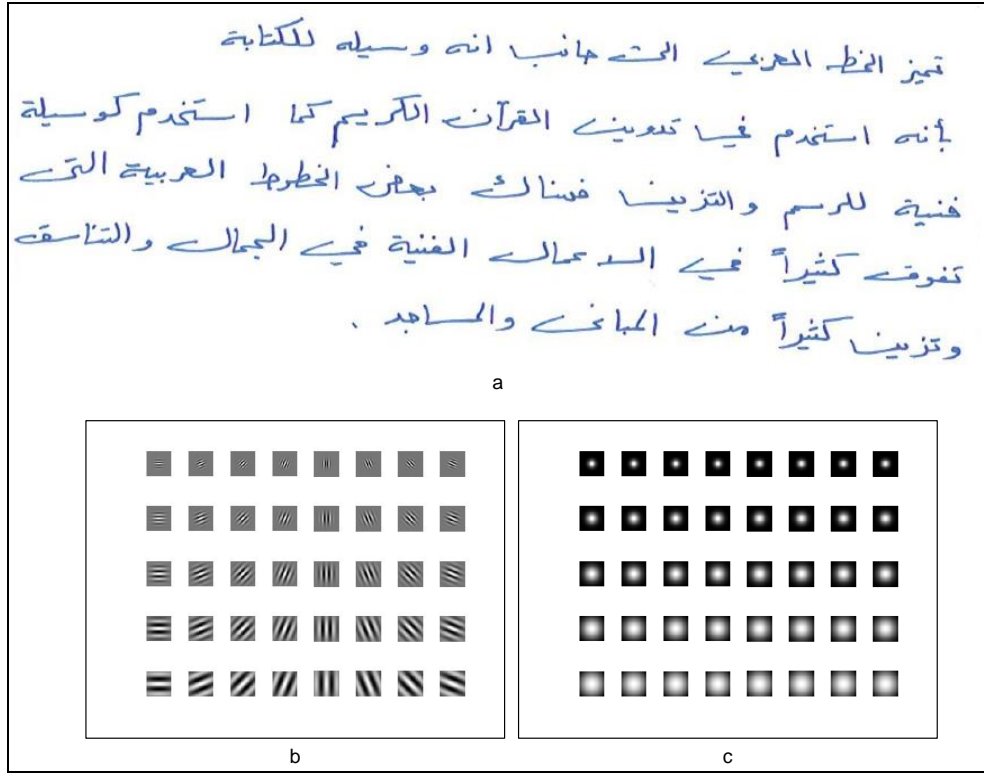


Figure 75: Apply Gabor filter for text2 writer2

It is worth noting that, that Gabor features can also be useful for text recognition. This provides a rich source of texture features but investigating their use for writer identification would certainly form a major task for the future not only for Arabic language but for many different ones.

### Connected strokes feature

Further investigation may take place in the future to do many processes on subword strokes (tail) like segmenting the last part of these subwords. The extra feature will extract to represent the tail (i.e. subword connected stroke). Table 6.9 shows different Arabic connected strokes which are written by different writers. These examples in this table below show the unique habit/style of each writer which is essential in identifying a particular writer.

**Table 6.9:** Examples of Arabic handwriting strokes for different writers.

English CS →								
Arabic CS →								
Writer1								
Writer2								
Writer3								

## 6.4 Summary of the Chapter

This chapter was designed to investigate the suitability of subwords based WI of Arabic text in the text-independent scenario whereby one attempts to identify the writer of a given text document/paragraph which is different from stored template text files. We adjusted our schemes developed in the previous chapters by first identifying the shared subwords in the different samples. This is another advantage of the subword base scheme in the sense that there are more chances to have several common subwords even if no words are shared. We tested the performance of the adjusted scheme on a modest size in-house built database and demonstrated that the thesis main hypothesis is still valid for this scenario albeit the reasonably good accuracy rate could not reach the optimal accuracy obtained when testing subwords without diacritics schemes on a relatively large text-dependent database

To determine whether this dip in performance can be attributed to the shortcoming of the tested database size and/or limited number of shared subwords, we faced the tough challenge of building a large population-representative database with a variety of samples. We solved this problem, what we claim to be a useful and adequate solution, by using the existing text-dependent database to simulate the creation of an imagined large database with large variation in the recorded text. The outcome of limited experiments on the simulated intext-dependent database has led to significant improvements in the performance of our final version of the subword WI scheme.

The achieved performance is suitable for many applications, but in some cases where high confidence in the automatic decision is a must we need to complement this scheme with other procedures. In the last section, we discussed a variety of future directions in research to complement our schemes. These included some suggestions on finding other features that may involve the way subwords are combined in words or lines. However, we discussed the use of scanned images of handwritten text and applying few well-known image analysis techniques that are normally used for image texture analysis and image-based pattern recognition research. We identified the Zigzag pattern coding widely used in image compression, the LBP maps and histogram analysis from pattern recognition, and the most sophisticated Gabor wavelet transforms that are widely used in multi-resolution and multi-orientation texture analysis and recognition. These are to be the subject of future work to deal with independent text writer identification and or handwritten text recognition.

## Chapter 7 : Conclusion

Writer identification (WI) from handwritten text in any language is a behavioral biometrics that is learnt, sharpened and refined from the early age when a person attends lessons and is influenced by many factors including their teachers, peers and family. The style of writing settles after a period of training and doesn't change significantly afterward. In general most people and particular teachers develop a reasonable skill in recognizing their close friends and family members from their handwriting. However, forensic experts specializing in writer identification go through extensive and complex training programs to develop their skills before their advice being accepted in a court of law with a high degree of confidence. Good knowledge of the language used is essential even when recognizing the text is required. In recent years, the rise of international terrorism interest in the identification of handwritten Arabic text due to the current wave of terrorist attacks originating from the Middle East. Automatic identification of a person from handwritten Arabic text samples was the main focus of our investigations in this thesis. Such investigations attract much interest well beyond fighting terrorism and forensics to include other personal interest in digitizing religious and historical archives in my homeland, Iraq, and the wider Middle East.

Reviewing the literature, presented in chapter 2, I found that in all languages handwriting habits and styles are embedded in certain parts/components of written texts, although the organisation of these components within paragraphs encapsulate important clues about the writer identity. In fact, word based WI techniques for an Arabic writer is one of the most common approaches in the literature. We also came to notice that WI was historically based on analysing paragraph(s), line(s), word(s), character(s), and/or a part of a character. Arabic is a cursive language when handwritten, and unlike many other languages, each word consists of one or more subwords and at the end of each subword there is a connected stroke. Many experts consider subwords stroke as a remarkable feature, unique only to a specific writer and recommend to take into account when identifying writers of Arabic text. An important distinct feature of Arabic writing is the variety of diacritics that have to be written within the words, and subwords, without which their meaning and pronunciation become difficult in most cases. Those who are fluent in Arabic language, let alone those whose mother tongue is Arabic, known for fact that subwords are more frequent in any written Arabic text and many subwords appear as part

of several different words or on their own as single words. This knowledge motivated the development of our hypothesis that subwords based WI would yield significant improvement in accuracy than existing approaches. The rest of the thesis was devoted to examine the hypothesis validity and refine the developed scheme for optimal performance.

The first challenging steps in automating WI for Arabic include (1) segmenting the written into its words, (2) extracting relevant measurements to form digital feature vector representation of the different words, and (3) determining the pattern recognition approach to be adopted. Once such an automatic WI scheme is developed, we need to provide empirical evidence to establish the validity the stated hypothesis. This requires the implementation of traditional biometric experimental protocols, and evaluation measures, using appropriate databases. However, we identified two seemingly different application scenarios: (1) the dependent scenario where we assume that sample of texts stored as templates in the gallery would coincide with fresh samples presented for identification, and (2) the text-independent scenario where the fresh text is different from those in the gallery.

In chapter3, we investigated different approaches to automatically segmenting Arabic handwritten texts into subwords. The scanned text images were pre-processed by a combining global thresholding followed by a morphological cleaning resulting a visibly clear binary image from which noise and other scanner-caused artefacts were removed. The problem of text orientation skew that has an adverse effect on segmentation of lines was dealt by introducing a Horizontal Projection Peak scheme to be used for correcting text orientation and successfully segment the text lines. The vertical version of this projection helped to segment the text lines into its words, subwords and diacritics component. Though this was very successful, we detected cases of severe text overlapping. This was dealt with by introducing an active shape LCC scheme to segment all the required text components. The lines and subwords LCC based segmentation scheme was further refined. The performance was tested on large text samples from two databases (the IFN/ENIT DB, and an in-house DB) and we demonstrated its superior performance of more than 98%, and most importantly the LCC algorithm maintains the features that are most important to our investigation like pattern slope.

Having succeeded in developing automatic subwords segmentation, we were now to progress to the next step of designing the subwords based WI schemes, i.e. define a set of measurements that can be obtained from each subword and from a writer

discriminating feature vector representation of a subword. We defined 15-dimensional attributes vectors, 13 of which were single value measurements that the last two were sequences of integers representing the horizontal and vertical projections. This has resulted in a dilemma as to how to define a similarity/distance function on subword feature vector. This would be needed for writer verification for two copies of a subword. These 15 features were described and illustrated in Chapter 4 where we also designed the first version of subwords WI scheme. We also conducted a pilot experiment to test performance of scheme(s) on a publically available text-text-dependent database of a large number of text samples consisting of 27 Handwritten Arabic words who copied these words 20 times. That DB was recorded by Al-Madeed et al., and in total we extracted 49 subwords. The pilot study, helped to verify the viability (not the validity) of our WI scheme and through an incremental procedure we reduced the dimensionality of the subwords feature vectors to 8 and ranked the 49 subwords. These results confirmed what might otherwise be expected, namely that subwords of length 1 do not have strong writer discriminating property.

In the rest of chapter 4, two schemes were developed through a refinement process whereby initially the number of features limited to 8 obtained by incremental procedure, and the two schemes are based on first using 22 subwords whereas the second scheme used only 11 subwords. The performance test experimental protocols were based on 5 text samples per writer being used as a gallery and the other 15 samples were used for testing. The database consisted of 20 text samples from 95 writers. In these experiments we used two different similarity functions, one where we considered the Euclidian distance function while the other one combines the Dynamic Time Warping (DTW) for the 2 projection attributes with Euclidian for the other 6 features. The experimental results confirmed beyond any doubt the validity of our hypothesis and outperformed existing word-base WI, which only achieves high accuracy at rank 10 nearest neighbours. The 11 subwords based schemes with the DTW related similarity function achieved the higher accuracy. These results, however, exhibited the possibility of the presence of a Doddington Zoo effect. We noted that the performance of both developed schemes (i.e. the 22 subwords scheme and the 11 subwords scheme) had the same pattern in relation to different groups of writers. The groups of writers in the two schemes that have a similar level of accuracy rates were almost identical, and when we removed the lowest group (i.e. wolfs in the Doddington Zoo terminology) the accuracy improved significantly. Accepting that this is not the way that the Doddington Zoo phenomena is dealt with in the literature, we decided not to follow this line.



Another pilot study was conducted to investigate the use of compressive sensing (and random projection) approach for dimension reduction to replace the 13 original single-valued features with a smaller number of meta-features formed from linear combinations of all the 13 features as an alternative to feature selection. The results are encouraging and provide strong motivation to conduct a more extended study in the future but with a far larger set of single features.

In chapter 5, we refined our previous subwords based WI scheme by using subwords bodies without their diacritics. This was based on observations that the inclusion of the diacritics results in significant variation in the way writers place and shape their diacritics around their subwords bodies which in turn cause variation in the extracted feature vectors. We first developed a diacritics removal segmentation scheme that achieved high accuracy with most subwords and most writers. The performance of segmentation scheme, however, highlighted possible impact on the performance of our WI scheme that could be manifested as increased incidents of false rejections and/or false acceptances, particularly for the group of writers classified based on subword including their diacritics(in chapter 4) as “unknown”.

The experiments demonstrated that removing the diacritics has led to significant improvement in the performance of subwords based WI scheme. Interestingly, the results have negatively answered the question regarding the Doddington Zoo phenomena and demonstrated beyond any doubt; that there is no evidence of this phenomenon and what we have been detected earlier was due to variation in the way diacritics are added to subwords body. The removal of diacritics provided the ultimate refinement of our subword based WI scheme.

Having succeeded in achieving optimal accuracy on dependent WI scenario, chapter 6 focused on investigation of the suitability of subwords based WI of Arabic text in the text-independent scenario. The previously developed scheme was adjusted first by determining the shared subwords in the different samples. This is another advantage of the subword base scheme in the sense that there are more chances to have several common subwords even if no words are shared. We tested the performance of the adjusted scheme on a modest size in-house built database and demonstrated that the thesis main hypothesis is still valid for this scenario albeit the reasonably good accuracy rate could not reach the optimal accuracy obtained when testing subwords without diacritics schemes on a relatively large text-dependent database

To determine if this dip in performance can be attributed to the shortcoming of the tested database size and/or limited number of shared subwords, we faced the tough challenge of building a large population-representative database with a variety of samples. We solved this problem, what we claim to be a useful and adequate solution, by using the existing text-dependent database to simulate the creation of an imagined large database with large variation in the recorded text. The outcome of limited experiments on the simulated in text-dependent database has led to significant improvements in the performance of our final version of the subword WI scheme.

The achieved performance is suitable for many applications, but in some cases where high confidence in the automatic decision is a must we need to complement this scheme with other procedures. In the last section, we discussed a variety of future directions in research to complement our schemes. These included some suggestions on finding other features that may involve the way subwords are combined in words or lines. However, we discussed the use of scanned images of handwritten text and applying few well-known image analysis techniques that are normally used for image texture analysis and image-based pattern recognition research. We identified the Zigzag pattern coding, the LBP maps and histogram analysis scheme of pattern recognition, and the most sophisticated Gabor wavelet transforms that are widely used in multi-resolution and multi-orientation texture analysis and recognition. These are to be the subject of future work to deal with independent text writer identification and or handwritten text recognition. Therefore, we consider that subwords are rich with vital information that can be used in the process of Arabic handwriting WI especially when diacritics are not included.

Finally, we note that Kurdish, Persian, Urdu, and other languages similar to Arabic have common features with Arabic writing structure, and all these languages are read from right to left. One of the main features of these languages is that the word might consist of many subwords as in Arabic and diacritics are also used. We shall attempt to extend, in the future, our work to some of these languages.

## References

- Abboud, A. J. & Jassim, S. A., 2012. *Incremental fusion of partial biometric information*. s.l., s.n., pp. 84060K--84060K.
- Abdi, M. N. & Khemakhem, M., 2010. Off-Line Text-Independent Arabic Writer Identification using Contour-Based Features. *International Journal of Signal and Image Processing*, Volume 1-2010/Iss.1, pp. 4-11.
- Aboul-Ela, S., Ghanem, N. & Ismail, M., 2015. *Comparative study on language independent forensic writer identification*. s.l., s.n.
- AL-ASSAM, H., 2013. *Entropy Evaluation and Security Measures for Reliable Single/Multi-Factor Biometric Authentication and Biometric Keys*. Buckingham, United Kingdom: Applied Computing Department, The University of Buckingham.
- Al-Dmour, A. & Zitar, R., 2007. Arabic writer identification based on hybrid spectral--statistical measures. *Journal of Experimental \& Theoretical Artificial Intelligence*, 19(4), pp. 307-332.
- AlKhateeb, J. H., Jiang, J., Ren, J. & Ipson, S., 2009. Component-based segmentation of words from handwritten Arabic text. *International Journal of Computer Systems Science and Engineering*, 5(1).
- Al-Ma'adeed, S., Mohammed, E. & Al Kassis, D., 2008. "Writer identification using edge-based directional probability distribution features for arabic words". pp. 582-590.
- AL-Shatnawi, A. M., AL-Salaimeh, S., AL-Zawaideh, F. & Omar, K., 2011. Offline Arabic Text Recognition--An Overview. *World of Computer Science and Information Technology Journal (WCSIT)*, 1(5), pp. 184-192.
- Amin, A., 1998. Off-line Arabic character recognition: the state of the art. *Pattern recognition*, 31(5), pp. 517-530.
- Arivazhagan, M., Srinivasan, H. & Srihari, S. N., 2007. A statistical approach to handwritten line segmentation. *Document Recognition and Retrieval XIV, Proceedings of SPIE, San Jose, CA, February* , Volume 6500T, pp. 1-11.

- Awaida, S. & Mahmoud, S., 2012. State of the art in off-line writer identification of handwritten text and survey of writer identification of Arabic text. *Educational Research and Reviews*, 7(20), pp. 445-463.
- Baghshah, M., Shouraki, S. & Kasaei, S., 2006. *A Novel Fuzzy Classifier using Fuzzy LVQ to Recognize Online Persian Handwriting*. s.l., s.n., pp. 1878-1883.
- Bar-Yosef, I., Hagbi, N., Kedem, K. & Dinstein, I., 2009. "Line Segmentation for Degraded Handwritten Historical Documents". *IEEE, Document Analysis and Recognition, 10th International Conference*, July, pp. 1161-1165.
- Bensefia, A., Nosary, A., Paquet, T. & Heutte, L., 2002. *Writer identification by writer's invariants*. s.l., s.n., pp. 274-279.
- Bensefia, A., Paquet, T. & Heutte, L., 2003. *Information retrieval based writer identification*. s.l., s.n., p. 946.
- Bensefia, A., Paquet, T. & Heutte, L., 2005a. A writer identification and verification system. *Pattern Recogn. Lett.*, Volume 26, pp. 2080-2092.
- Bensefia, A., Paquet, T. & Heutte, L., 2005b. Handwritten document analysis for automatic writer recognition. *Electronic letters on computer vision and image analysis*, 5(2), pp. 72-86.
- Berkani, D. & Hammami, L., 2002. "Recognition system for printed multi-font and multi-size Arabic characters". *The Arabian Journal for Science and Engineering*, April, 27(1B), pp. 57-72.
- Brik, Y., Chibani, Y., Zemouri, E.-T. & Sehad, A., 2013. *Ridgelet-DTW-based word spotting for Arabic historical document*. s.l., s.n., pp. 194-199.
- Bruzzone, E. & Coffetti, M. C., 1999. An algorithm for extracting cursive text lines. *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, p. 749.
- Bulacu, M., 2007. *Statistical pattern recognition for automatic writer identification and verification*. s.l.:Citeseer.
- Bulacu, M. & Schomaker, L., 2004a. *Analysis of texture and connected-component contours for the automatic identification of writers*. s.l., s.n., pp. 371-372.
- Bulacu, M. & Schomaker, L., 2006. *Combining multiple features for text-independent writer identification and verification*. s.l., s.n., pp. 281-286.

- Bulacu, M., Schomaker, L. & Brink, A., 2007. Text-Independent Writer Identification and Verification on Offline Arabic Handwriting. *Document Analysis and Recognition, International Conference on*, Volume 2, pp. 769-773.
- Chanda, S., Franke, K., Pal, U. & Wakabayashi, T., 2010. *Text independent writer identification for bengali script*. s.l., s.n., pp. 2005-2008.
- Favata, J. & Srikantan, G., 1996. A multiple feature/resolution approach to handprinted digit and character recognition. *International journal of imaging systems and technology*, 7(4), pp. 304-311.
- Feldbach, M. & Tönnies, K. D., 2001. "Line Detection and Segmentation in Historical Church Registers". *Sixth International Conference on Document Analysis and Recognition*, September. pp. 743-747.
- Fletcher, L. A. & Kasturi, R., 1988. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6), p. 910–918.
- Flusser, J., 2000. On the independence of rotation moment invariants. *Pattern Recognition*, 33(9), pp. 1405-1410.
- Fornasier, M. & Rauhut, H., 2011. Compressive sensing. In: *Handbook of Mathematical Methods in Imaging*. s.l.:Springer, pp. 187-228.
- Gazzah, S. & Amara, N., 2008. Neural Networks and Support Vector Machines Classifiers for Writer Identification Using Arabic Script. *The International Arab Journal of Information Technology*, 5(1), pp. 93-102.
- Gazzah, S. & Amara, N. B., 2007. *Arabic Handwriting Texture Analysis for Writer Identification Using the DWT-Lifting Scheme*. s.l., s.n., pp. 1133-1137.
- Gazzah, S. & Ben, N. E., 2006. Writer identification using modular MLP classifier and genetic algorithm for optimal features selection. *Advances in Neural Networks- ISSN 2006*, pp. 271-276.
- Gdyczynski, C. et al., 2014. On estimating the directionality distribution in pedicle trabecular bone from micro-CT images. *Physiological Measurement*, 35(12), p. 2415.
- Gonzalez, R. C., Woods, R. E. & Eddins, S. L., 2009. *Digital Image Processing Using MATLAB*. s.l.:Gatesmark Publishing, Knoxville, TN..

- Güler, İ. & Meghdadi, M., 2008. A different approach to off-line handwritten signature verification using the optimal dynamic time warping algorithm. *Digital Signal Processing*, 18(6), pp. 940-950.
- Helli, B. & Moghadam, M. E., 2008b. *Persian Writer Identification Using Extended Gabor Filter*. s.l., Springer-Verlag, pp. 579-586.
- Helli, B. & Moghaddam, M., 2009. A writer identification method based on XGabor and LCS. *IEICE Electronics Express*, 6(10), pp. 623-629.
- Helli, B. & Moghaddam, M. E., 2008a. *A Text-Independent Persian Writer Identification System Using LCS Based Classifier*. s.l., s.n., pp. 203-206.
- Helli, B. & Moghaddam, M. E., 2010. A text-independent Persian writer identification based on feature relation graph (FRG). June, 43(6), pp. 2199-2209.
- Hertel, C. & Bunke, H., 2003. *A set of novel features for writer identification*. s.l., s.n., pp. 1058-1058.
- Huber, R. & Headrick, A., 1999. *Handwriting identification: facts and fundamentals*. s.l.:CRC.
- HU, M., 1962. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8(2), pp. 179-187.
- Kohn, M. S., Sassi, P. & Thies, j., 2011. <http://www.handwritinginsights.com/lesson.html>. [Online].
- Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1), pp. 59-69.
- Kumar, J., Abd-Almageed, W., Kang, L. & Doermann, D., 2010. *Handwritten arabic text line segmentation using affinity propagation*. s.l., s.n., pp. 135-142.
- Kumar, K. S., Namboodiri, A. M. & Jawahar, C. V., 2006. "Learning Segmentation of Documents with Complex Scripts". *Fifth Indian Conference on Computer Vision, Graphics and Image Processing*, Volume LNCS 4338, pp. 749-760.
- Leedham, G. & Chachra, S., 2003. Writer identification using innovative binarised features of handwritten numerals. *Document Analysis and Recognition*, Volume 1, p. 413.
- Lewis, M. P., 2009. *Ethnologue: Languages of the World*. s.l.:Sixteenth edition..

- Likforman-Sulem, L. & Faure, C., 1994. "Extracting text lines in handwritten documents by perceptual grouping". *Advance in handwriting and drawing : a multidisciplinary approach* C. Faure, P. Keuss, G. Lorette and A. Winter Eds, pp. 117-135..
- Likforman-Sulem, L., Hanimyan, A. & Faure, C., 1995. A Hough based algorithm for extracting text lines in handwritten documents. *Proceedings of the Third International Conference on Document Analysis and Recognition*, p. 774–777.
- Likforman-Sulem, L., Zahour, A. & Taconet, B., 2007. ' Text Line Segmentation of Historical Documents: a Survey '. *International Journal on Document Analysis and Recognition*, Springer-Verlag, April .Volume 9 (2).
- Li, Y., Zheng, Y., Doermann, D. & Jaeger, S., 2006. "A new algorithm for detecting text line in handwritten documents". in *International Workshop on Frontiers in Handwriting Recognition*, p. 35–40.
- Lorigo, L. & Govindaraju, V., 2006. Offline Arabic handwriting recognition: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5), pp. 712-724.
- Lorigo, L. M. & Govindaraju, V., 2006. Offline Arabic handwriting recognition: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, May , Volume vol.28, no.5, pp. 712-724.
- Lutf, M., You, X. & Li, H., 2010. *Offline Arabic Handwriting Identification Using Language Diacritics*. s.l., s.n., pp. 1912-1915.
- Maaten, L. v.-d. & Postma, E., 2005. *Improving automatic writer identification*. s.l., s.n., pp. 260-266.
- Maliki, M., Al-Jawad, N. & Jassim, S., 2013. *Sub-word based Arabic handwriting analysis for writer identification*. s.l., s.n., pp. 87550M--87550M.
- Maliki, M., Jassim, S., Al-Jawad, N. & Sellahewa, H., (2012). "Arabic Handwritten: Pre-Processing and segmentation,". Baltimore Maryland USA, SPIE Defence Security and Sensing (To be published) in 23-27 April.
- Maliki, M., Jassim, S., Al-Jawad, N. & Sellahewa, H., 2012. *Arabic handwritten: pre-processing and segmentation*. s.l., s.n., p. 10.

- Manmatha, R. & Rothfeder, J. L., 2005. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Trans. Pattern Anal. Mach. Intell*, Volume 27 (8), p. 1212–1225.
- Marti, U. & Bunke, H., 2002. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1), pp. 39-46.
- Marti, U., Messerli, R. & Bunke, H., 2001. *Writer identification using text line based features*. s.l., s.n., pp. 101-105.
- Mathieu, B., 2009. *Mathieu's log (Machine Learning, Data Mining, Natural Language Processing)*. [Online] Available at: <http://www.mblondel.org/journal/2009/08/31/dynamic-time-warping-theory/> [Accessed 31 August 2015].
- Morse, B. S., 2000. Lecture 4: Thresholding. *Brigham Young University, Utah*.
- Niels, R., Vuurpijl, L. & others, 2005. *Using Dynamic Time Warping for intuitive handwriting recognition*. s.l., s.n., pp. 217-221.
- Ojala, T. & Pietikäinen, M., 1999. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3), pp. 477-486.
- Ojala, T., Pietikäinen, M. & Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1), pp. 51-59.
- Ojala, T., Pietikainen, M. & Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7), pp. 971-987.
- Otsu, N., 1979. A Threshold Selection Method from Gray-Level Histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1), pp. 62-66.
- Pantech, 2014. *Pantech Solutions Technology Beyond The Dreams*. [Online] Available at: <https://www.pantechsolutions.net/blog/matlab-code-to-read-matrix-in-a-zigzag-scanning/> [Accessed 29 August 2015].



- Pechwitz, M., Maddouri, S. S., Maergner, V. & Ellou, N., 2002. IFN/ENIT - database of handwritten Arabic words. *Proc. of CIFED*, p. 129–136.
- Poh, N., Bengio, S. & Ross, A., 2006. *Revisiting Doddington's Zoo: A Systematic Method to Assess User-dependent Variabilities*. s.l., s.n.
- Poh, N., Kittler, J. & Boutilier, T., 2010. Quality-Based Score Normalization With Device Qualitative Information for Multimodal Biometric Fusion. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, May, 40(3), pp. 539-554.
- Rafiee, A. & Motavalli, H., 2007. *Off-Line Writer Recognition for Farsi Text*. s.l., s.n., pp. 193-197.
- Ram, S. & Moghaddam, M., 2009. *A Persian Writer Identification Method Based on Gradient Features and Neural Networks*. s.l., s.n., pp. 1-4.
- Ram, S. S. & Moghaddam, M., 2009. *Text-independent Persian Writer Identification Using Fuzzy Clustering Approach*. s.l., s.n., pp. 728-731.
- Safabakhsh, R. & Adibi, P., 2005. Nastaaligh handwritten word recognition using a continuous-density variable-duration HMM. *Arabian Journal for Science and Engineering*, 30(1), pp. 95-120.
- Said, H., Baker, K. & Tan, T., 1998. *Personal identification based on handwriting*. s.l., s.n., pp. 1761 -1764 vol.2.
- Said, H., Peake, G., Tan, T. & Baker, K., 1998. *Writer identification from non-uniformly skewed handwriting images*. s.l., s.n., pp. 478-487.
- Schlapbach, A., Kilchherr, V. & Bunke, H., 2005. *Improving writer identification by means of feature selection and extraction*. s.l., s.n., pp. 131-135.
- Schlapbach, A., Liwicki, M. & Bunke, H., 2008. A writer identification system for on-line whiteboard data. *Pattern recognition*, 41(7), pp. 2381-2397.
- Schomaker, L., 2007. *Advances in Writer Identification and Verification*. s.l., s.n., pp. 1268-1273.
- Schomaker, L. & Bulacu, M., 2004. Automatic writer identification using connected-component contours and edge-based features of uppercase Western script. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6), pp. 787-798.

- Schomaker, L., Bulacu, M. & Franke, K., 2004b. Automatic writer identification using fragmented connected-component contours.
- Schomaker, L., Franke, K. & Bulacu, M., 2007. Using codebooks of fragmented connected-component contours in forensic and historic writer identification. *Pattern recognition letters*, 28(6), pp. 719-727.
- Shahabi, F. & Rahmati, M., 2006. *Comparison of Gabor-based features for writer identification of Farsi/Arabic handwriting*. s.l., s.n.
- Shahabi, F. & Rahmati, M., 2007. *A New Method for Writer Identification and Verification Based on Farsi/Arabic Handwritten Texts*. s.l., s.n., pp. 829-833.
- Sreeraj.M & Idicula, S. M., 2011. A Survey on Writer Identification Schemes. *International Journal of Computer Applications*, 26(2), pp. 23-33.
- Srihari, S., Cha, S.-H. & Lee, S., 2001. *Establishing handwriting individuality using pattern recognition techniques*. s.l., s.n., pp. 1195-1204.
- Tan, T., 1998. Rotation invariant texture features and their use in automatic script identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(7), pp. 751-756.
- Tomai, C., Zhang, B. & Srihari, S., 2004. Discriminatory power of handwritten words for writer recognition. *Pattern Recognition*, Volume 2, pp. 638-641.
- van der Maaten, L. J., Postma, E. O. & van den Herik, H. J., 2009. Dimensionality reduction: A comparative review.
- van der Maaten, L., Postma, E. & van den Herik, H., 2007. *Matlab toolbox for dimensionality reduction*. s.l., s.n., pp. 439-440.
- Wu, X., Tang, Y. & Bu, W., 2014. Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform. *Information Forensics and Security, IEEE Transactions on*, 9(3), pp. 526-536.
- Zhang, B. & Srihari, S., 2003. Analysis of Handwriting Individuality Using Word Features. *Document Analysis and Recognition*, Volume 2, p. 1142.
- Zhu, Y., Tan, T. & Wang, Y., 2001. Font recognition based on global texture analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10), pp. 1192-1200.

- Zoghbi, P., 2007, Accessed 13 April 2013. *History of Arabic Type Evolution from the 1930's till present, 29letters*. s.l.:s.n.
- Zois, E. & Anastassopoulos, V., 2000. Morphological waveform coding for writer identification. *Pattern Recognition*, 33(3), pp. 385-398.
- Zuo, L., Wang, Y. & Tan, T., 2002. *Personal handwriting identification based on pca*. s.l., s.n., pp. 766-771.

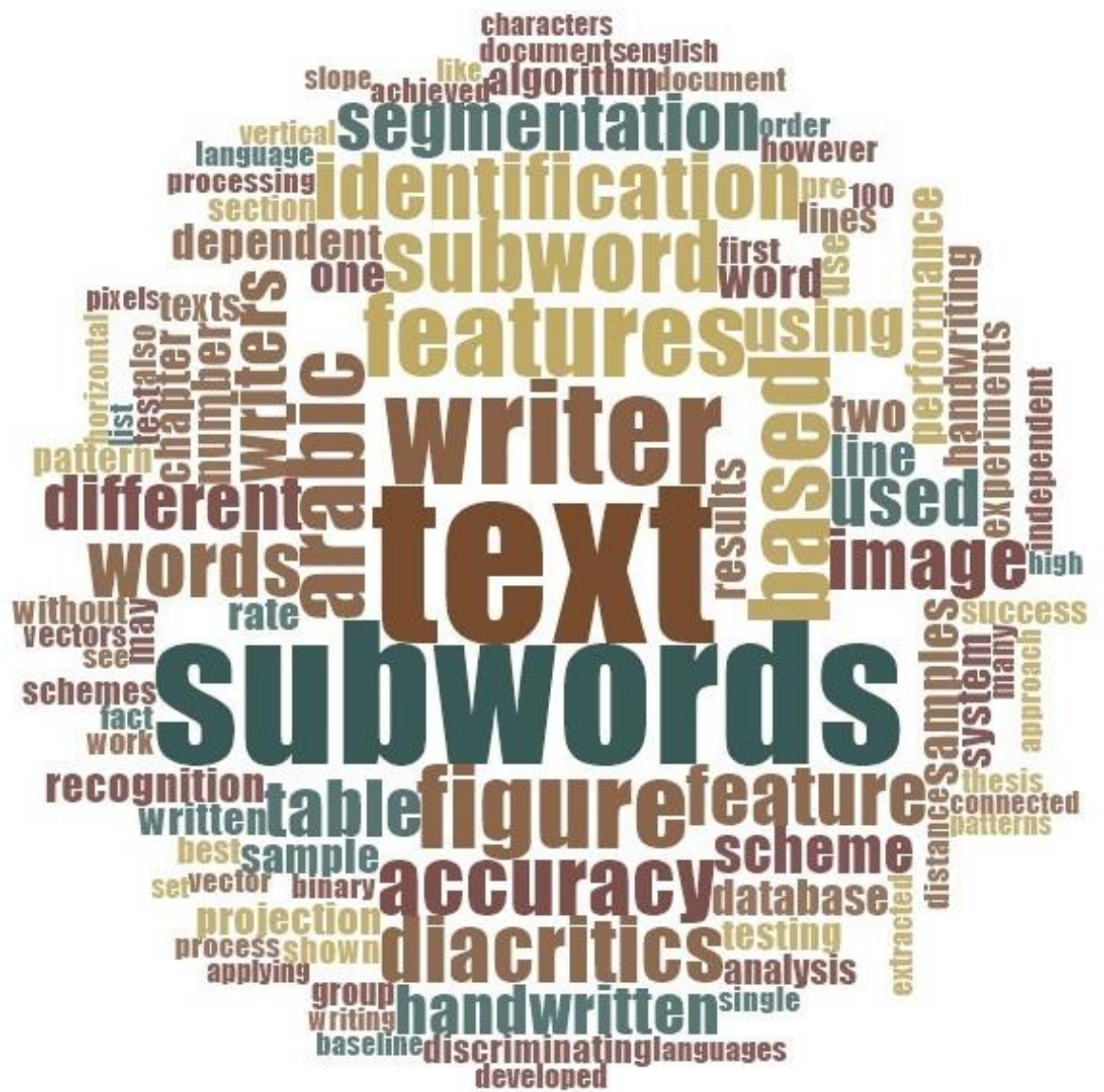
## Appendix

**Table 1:** The Arabic alphabet

No	Name of Letter in Arabic	Sound	Example in English	Isolated	Initial	Medial	Final
1.	Alif	Ā	'a' as in 'father'	ا	ا	ا	ا، ا
2.	Baa	B	'b' as in 'bed'	ب	ب	ب	ب، ب
3.	Taa	T	't' as in 'tent'	ت	ت	ت	ت، ت
4.	Thaa	Th	'th' as in 'think'	ث	ث	ث	ث، ث
5.	Jiim	J [d̤ʒ] , [ʒ] , [g] ]	Jam	ج	ج	ج	ج، ج
6.	Haa	H	(Deep H)	ح	ح	ح	ح، ح
7.	Khaa	ḥ (kh)	'ch' as in German 'Bach'	خ	خ	خ	خ، خ
8.	Daal	D	Deer	د	د	د	د، د
9.	Thaal	ḏ (dh, ð)	There	ذ	ذ	ذ	ذ، ذ
10.	Raa	R	Run	ر	ر	ر	ر، ر
11.	Zay	Z	Zoo	ز	ز	ز	ز، ز
12.	Siin	S	Sit	س	س	س	س، س
13.	Shiin	š (sh)	Shut	ش	ش	ش	ش، ش
14.	Saad	ṣ	(deep S) sold	ص	ص	ص	ص، ص

No	Name of Letter in Arabic	Sound	Example in English	Isolated	Initial	Medial	Final	
15.	Dhaad	ḍ	(Deep D) bulldozer	ض	ضـ	ضـ	ض، ضـ	
16.	Taa	ṭ	(Deep T) Tasmania	ط	طـ	طـ	ط، طـ	
17.	Dhaa	ẓ	(Deep Z) those	ظ	ظـ	ظـ	ظ، ظـ	
18.	Ayn	ʿ	(Deep and thirsty 'a') Aww	ع	عـ	عـ	ع، عـ	
19.	Ghayn	ġ (gh)	Paris (French R)	غ	غـ	غـ	غ، غـ	
20.	Faa	F	Free	ف	فـ	فـ	ف، فـ	
21.	Qaaf	Q	Qum	ق	قـ	قـ	ق، قـ	
22.	Kaaf	K	King	ك	كـ	كـ	ك، كـ	
23.	Laam	L	Lift	ل	لـ	لـ	ل، لـ	
24.	Miim	M	Moon	م	مـ	مـ	م، مـ	
25.	Nuon	N	Net	ن	نـ	نـ	ن، نـ	
26.	Haa	H	House	ه	هـ	هـ	ه، هـ	
27.	Waaw	W	wonder	و	وـ	وـ	و، وـ	
28.	Yaa	Y	yellow	ي	يـ	يـ	ي، يـ	
Modified Characters	29.	Alif with above Hamza	<i>a, 'u</i>	/a/, /u/ with a sudden stop	أ	أ	أ	أ
	30.	Alif with below Hamza	<i>'i</i>	/i/ with a sudden stop	إ	إ	إ	إ
	31.	Alif with above Madda	<i>'ā</i>	/a:/ with a sudden stop	آ	آ	-	-
	32.	Hamza	U	Ugh	ء	-	-	ء

No	Name of Letter in Arabic	Sound	Example in English	Isolated	Initial	Medial	Final
33.	Taa (tied)	<i>h</i> or <i>t / h / t̄</i>	/a/, /at/	ة	-	-	ة،ة
34.	Waaw with above Hamza	au : /ɔu/	Show, boat	ؤ	-	ؤ	ؤ،ؤ
35.	Alif Maqsura	<i>ā / ȳ:</i> /a:/	Car	ى	-	-	ى،ى
36.	Alif Maqsora with above Hamza	<i>a, 'u</i>	bus	ئ	ئ	ئ	ئ،ئ



Word Frequency Clued of the Thesis

