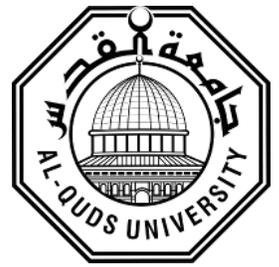


**Deanship of Graduation Studies**

**Al-Quds University**



**Auditing Electronic Files of Qur'an Using Optical Character  
Recognition**

**Alaá Fathala Abdlazez Hamda**

**M.Sc. Thesis**

**Jerusalem - Palestine**

**1438 / 2017**

# **Auditing Electronic Files of Qur'an Using Optical Character Recognition**

**Prepared By:**

**Alaá Fathala Abdlazez Hamda**

B.Sc. : Computer System Engineering, Palestine Technical University- PTUK / Palestine

Supervisor: Dr. Labib Arafa

Athesis Submitted in Partial Fulfillment of the Requirements for  
the Degree of Master of Electronic and Computer Engineering,  
at Al-Quds University

**1438/2017**

**Al-Quds University**

**Deanship of Graduation Studies**

**Electronic and Computer Engineering/Faculty of Engineering**

**Thesis Approval**

Auditing Electronic Files of Qur'an Using Optical Character Recognition

Prepared By: Ala'a Fathala Abdlazez Hamda

Registration No.: 201411737

Supervisor: Dr. Labib Arafa

Master thesis submitted and accepted, Date: 9/1/2017

**The names and signatures of the examining committee members are as follows:**

1- Head of Committee: Dr. Labib Arafeh

Signature .....

2 -Internal Examiner : Dr. Ahmad Qutob

Signature.....

3 -External Examiner : Radwan Tahboub

Signature.....

**Jerusalem – Palestine**

**1438/2017**

**Declaration:**

I Certify that this thesis submitted for the Degree of Master is the result of my own research, except where otherwise acknowledged, and that this thesis (or any part of the same) has not been submitted for a higher degree to any other university or institution.

**Signed: .....**

**Ala'a Fathala Abdlazez Hamda**

**Date: 9, January, 2017**

## **Dedication**

I dedicate this work to my parent who always supported me.

**Ala'a Fathala Abdlazez Hamda**

## **Acknowledgement**

As I am writing the last words of this thesis, I greatly appreciate the thesis supervisors Dr. Labib Arafah and Dr. Radwan Tahboub for their support and their time that they spent with me in order to make this thesis a successful one.

I would like to thank the thesis examiner Dr. Ahmad Qutob for their valuable suggestions and corrections on this work, which greatly helped me to improve it in various aspects.

Special Thank for, my friend Rash Saffarini and Israa Abusada for the support they gave to accomplish this achievement.

Last but not least, most profound gratitude and respect to my family, especially my beloved parents Fathala Hamda and Basema Hamda, and my family who are the ultimate source of my motivation to work hard and they are also the inspiration of my life. Therefore, I proudly dedicate this work to all of them. And may Allah SWT bless all of them.

## **Abstract**

Optical character recognition (OCR) systems improve human- machine interaction. They are widely used in many areas such as editing and storing previously printed or handwritten documents. Much of research has been done regarding the identification of printed font and handwritten script and achieves acceptable recognition. In this thesis we have used optical character recognition for special Arabic fonts in Auditing electronic files of the Holy Quran.

The ultimate goal of this research is building a software tool to perform auditing Holy Quran pages and compare with reference, before converting the page to the printing stage and publish the soft copy in order to reduce the time and manual auditing efforts during the audit of electronic files in the prepress stage. Reduce errors ratio that may occur during print phase and finally detect errors that may appear and documenting their location on each page.

The main contributions in this thesis, in the segmentation stage it used Up-Contour Extraction to spilt sub-word into character and to avoid overlapping problem which appears in OCR systems ,and using matrix summation to recognition character and the other main contribution in auto correct system which corrects the detected error depending on the reference copy of the Quran.

In this Thesis, 24 “Suras” from Holy Quran tested in many cases, and created a data base for this “Suras”. Furthermore, because the system is dealing with the Quran that must achieve 100% success rate. For this, in the future work the system will be expanded to include all the Holy Quran and create database greater than used now.

تدقيق القرآن باستخدام التعرف الضوئي على الحروف

إعداد : آلاء فتح الله عبد العزيز حمدة

إشراف : د. لبيب عرفة

### ملخص :

يحسن نظام التعرف الضوئي على الحروف التفاعل بين الإنسان والحاسوب. ويستخدم على نطاق واسع في العديد من المجالات مثل تحرير و أرشفة الوثائق المطبوعة او المكتوبة بخط اليد . و تم إنجاز العديد من الابحاث في مجال خط الطباعة والكتابة بخط اليد وحققت نتائج جيدة في التعرف على الحروف. في هذه الأطروحة تم استخدام التعرف الضوئي على الحروف في تدقيق صفحات القرآن الكريم المرسومة بالخط العثماني والتأكد من خلوها من أي أخطاء.

والهدف النهائي من هذا البحث هو بناء أداة لتدقيق الملفات التي تحتوي على صفحات القرآن ومقارنتها مع مرجع تم اعتماده من قبل مطبعة الملك فهد قبل إرسال هذه الملفات إلى مرحلة الطباعة والنشر من أجل تقليل الوقت والجهد المبذول في التدقيق اليدوي و مراجعة الملفات والكشف عن الأخطاء التي قد تظهر وتوثيق موقعها في كل صفحة.

المساهمات الرئيسية في هذه الأطروحة، في مرحلة تقسيم الحروف تم استخدام إطار الكلمة لتقسيم الكلمة الى حروف وتجنب مشكلة التداخل التي تظهر في نظام التعرف الضوئي على الحروف .

في مرحلة التعرف على الأحرف تم بناء خوارزمية تعتمد على ارتفاع وعرض الحرف ومجموع النقاط في الصفوف والأعمدة التي يتكون منها الحرف والمساهمة الرئيسية الأخرى في النظام هو التصحيح التلقائي للأخطاء اعتماداً على نسخة مرجعية من القرآن الكريم.

في هذه الأطروحة تم اختبار 24 سورة من القرآن الكريم وبناء قاعدة بيانات خاصة بهذه السور. وفحص قدرة النظام على اكتشاف الخطأ المختلفة التي تظهر في هذه السور سواء في الحروف أو الحركات أو علامات الضبط. ولكن لأن النظام يتعامل مع القرآن يجب تحقيق نسبة نجاح 100% حتى يكون نظام فعال. لهذا في العمل المستقبلي سيتم توسيع النظام ليشمل كل سور القرآن الكريم وإنشاء قاعدة بيانات أكبر من المستخدمة حالياً تشمل جميع السور.

## Table of content

<b>1 Introduction and Theoretical Background</b>	<b>2</b>
1.1 Background .....	3
1.2 Motivation .....	4
1.3 Research Objectives .....	5
1.4 Scope of this Work .....	5
1.5 Contribution .....	6
1.6 Theoretical Background .....	6
1.6.1 Computer vision .....	7
1.6.2 Character Recognition .....	8
1.6.3 Different Families of Character Recognition.....	9
1.7 Optical Character Recognition Process .....	12
1.7.1 The Imaging Stage .....	12
1.7.2 The OCR Process Stage .....	12
1.7.3 Preprocessing Phase.....	14
1.7.4 Segmentation .....	17
1.7.5 Recognition Phase .....	20

1.8 World Languages and Scripts :	21
1.8.1 Non-Cursive Script	22
1.8.2 Cursive Script	23
1.8.3 Arabic Script	23
1. 9 Thesis Roadmap	26
1.10 Summary :	27
<b>2 Literature Survey</b>	<b>28</b>
2.1 Introduction	28
2.2 English language OCR System	29
2.3 Approaches for printed Arabic script OCR	30
2.3.1 Ligature-based Approach :	32
2.3.2 Segmentation-based Approach :	32
2.4 Previous Work in Ligature-Based Arabic OCR	32
2.5 Previous Work in Segmentation-Based Arabic OCR	38
2.6 OCR in Holy-Quran	44
2.7 Summary	45
<b>3 Auditing Electronic Files Of Quran</b>	<b>48</b>
3.1 Introduction	48
3.2 proposed system	49
3.2.1 Electronic Auditing Tool of the Holy Quran Stage	49

3.3 Image Acquisition.....	50
3.4 Pre-processing.....	51
3.4.1 Grayscale: .....	52
3.4.2 Image Filtering:.....	54
3.4.3 Sharpening .....	55
3.4.4 Image Rotation.....	55
3.4.5 Text Skeletonization .....	56
3.5 Segmentation .....	58
3.5.1 Text to line Segmentation: .....	59
3.5.1.1 Problem in segmentation of lines: .....	59
3.5.2 Line to words or sub-words: .....	62
3.5.3 Segment each word or sub-word into characters: .....	65
3.5.3.1 Up-Contour Extraction: .....	68
3.5.3.2 Splitting Area extraction: .....	69
3.5.3.3 The cases that the splitting areas should be ignored:.....	70
3.5.4 Extract diacritics from characters: .....	72
3.6 Recognition.....	73
3.6.1 X-Y projection: .....	73
3.6.2 Freeman chain code: .....	74

3.6.3 AddCharacterstoDatabase:.....	75
3.6.4 Prove"MatrixSummation"ascharacterfeature: .....	76
3.6.5 RecognitionAlgorithm: .....	78
3.7 Auditfiled .....	79
3.8 Auto correction .....	79
3.9 Summary .....	82
<b>4 Evaluation and result</b>	<b>83</b>
4.1 Introduction.....	83
4.2 SystemEvaluation .....	83
4.3 ProveOCRSystemDatabase .....	84
4.4 AuditingResult .....	85
4.5 Database.....	94
4.6 Auto correction .....	96
4.7 ResultEvaluation.....	97
4.8 Symmary .....	98
<b>5 Conclusion</b>	<b>99</b>
5.1 Introduction.....	99
5.2 Conclusion .....	99
5.3 Contribution .....	101

5.4 Obstacles.....	101
5.5FutureWork.....	102
<b>6 References</b>	<b>103</b>
<b>7 Appendix A</b>	<b>110</b>

## List of Figure

Figure 1.1: The basic processes of an OCR	7
Figure 1.2: Sub-fields of Artificial Intelligence	8
Figure 1.3: The different families of character recognition.	10
Figure 1.4: Offline text containing just special information (left), online text containing temporal sequence of points traced out by the pen (right).	12
Figure 1.5: Offline Optical Character Recognition Classic process flow	14
Figure 1.6: A skewed text line.	17
Figure 1.7: Hierarchy of Offline segmentation step of OCR	18
Figure 1.8: Cursive and non-cursive scripts.	22
Figure 1.9: The base line	24
Figure 1.10: Ligatures	25
Figure 1.11: Segmenting ligature as one character.	25
Figure 2.1 Ligature-based approaches	31
Figure 3.1 OCR System Stage.	48
Figure 3.2 Remove border	49
Figure 3.3 Output Image after Auditing Holy-Quran	50
Figure 3.4 Image Acquisition	51
Figure 3.5 Pre-processing stage	52
Figure 3.6 Gray scale operations	53
Figure 3.7 Filtered image	54
Figure 3.8 Image before and after Sharpening	55

Figure 3.9 Image before and after Rotation	56
Figure 3.10 Image before and after Skeletonization	57
Figure 3.11 overlaps in Arabic word.	58
Figure 3.12 Segmentation step .	59
Figure 3.13 horizontal projection Line1.	60
Figure 3.14 horizontal projection Line2.	60
Figure 3.15 Vertical segmentation Sub-Word Result	65
Figure 3.16 Junction lines Example.	65
Figure 3.17 boundary example	67
Figure 3.18 Up contour Example.	68
Figure 3.19 Up-Contour extractions	69
Figure 3.20 suggest split area example	69
Figure 3.21 Splitting area ignore in case of SAD	70
Figure 3.22 Splitting area ignore in case of SEEN	71
Figure 3.23 Segment each word or sub-word into characters	72
Figure 3.24 Extract diacritics	72
Figure 3.25 Chain codes based freeman chain	74
Figure 3.26 kaf letter ' ك ' vector	74
Figure 3.27 Dal ' د ' in Binary	75
Figure 3.28 V & Horiz vectors sample	76
Figure 3.29 Sample System Result	80
Figure 3.30 Sura from Reference	80

Figure 3.31 Compare Input With Reference	81
Figure 4.1 Pre- process results in Al-Ekhlras "سورة الإخلاص" part 1	85
Figure 4.2 Pre- process results in Al-Ekhlras "سورة الإخلاص" part 2	86
Figure 4.3 Segmentation results on Al-Ekhlras "سورة الإخلاص"	87
Figure 4.4 result of Audit stage	88
Figure 4.5 number of errors before updating Database	89
Figure 4.6 Modify database	89
Figure 4.7 number of errors after updating Database	90
Figure 4.8 sura in two page part1	91
Figure 4.9 sura in two page part2	92
Figure 4.10 small-sized Quran part1	93
Figure 4.11 small-sized Quran part 2	94
Figure 4.12 database part 1 connected Character	94
Figure 4.13 database part 2 spilt character	95
Figure 4.14 database part 3 adjustment sign	95
Figure 4.15Auto correction	96

## List of Algorithm

Algorithm 1 show Pre-processing steps	57
Algorithm 2 Text to line segmentation	61
Algorithm 3 Words Segmentation	63
Algorithm 4 Sub Words Segmentation	64
Algorithm 5 Find base line	67
Algorithm 6 Matrex summation	78

## List of Abbreviations

<b>ANN</b>	Artificial Neural Network
<b>BPM</b>	Bitmap
<b>DIP</b>	Digital Image processing.
<b>DPI</b>	Digit Per Inch.
<b>H</b>	Number of holes.
<b>HMM</b>	Hidden Markov Models
<b>LTR</b>	from left to right
<b>MATLAB</b>	Matrix Laboratory
<b>NN</b>	Neural Network
<b>OCR</b>	Optical Character Recognition
<b>PR</b>	Pattern Recognition.
<b>RTF</b>	Rich Text Format
<b>TIFF</b>	Tagged-Image File Format

## Chapter 1

---

### Introduction and Theoretical Background

Artificial Intelligence is a broad field of computer science. Elaine Rich and Kevin Knight as gave one of the most popular definitions to Artificial Intelligence (AI), “Artificial Intelligence (AI) is the study of how to make computers do things which, at the moment, people do better”. One important branch of Artificial intelligence is Computer vision, which aims to imitate human vision and forms the basis of all image acquisition, its processing, document understanding and recognition. [1]

Computer vision relies on a solid understanding of the physical process of image formation to obtain simple inference from individual pixel values like shape of the object and to recognize objects using geometric information or probabilistic techniques. Character recognition is a sub-field of pattern recognition in which images of characters from a text image are recognized and as a result of recognition respective character codes are returned, these when rendered give the text in the image. [2]

This thesis concentrates on auditing electronic files of the Quran using optical character recognition for special Arabic font Uthmani (الرسم العثماني). Electronic files contain pages of Quran, compared electronically with their assets, before converting the files to the printing stage or publish the soft copy. In order to reduce the time and manual effort during the audit of electronic files in the prepress stage, and reduce errors ratio that may occur during the electronic processing phase and. finally detect errors that may appear, and documenting their location on each page.

## 1.1 Background

Optical Character Recognition is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded/computer-readable text. It is widely used as a form of data entry from some sort of original paper data source, whether passport documents, invoices, bank statement, receipts, business card, mail, or any number of printed records. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, key data extraction and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision. [3]

Early OCR versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs. Some systems are capable of producing formatted output that closely approximates the original page including images, columns, and other non-textual components. [3]

OCR consists of many Types that include [4]:

- 1- Optical Character Recognition (OCR) – targets typewritten text, one character at a time.
- 2- Optical Word Recognition – targets typewritten text, one word at a time (for languages that use a space as a word divider).
- 3- Intelligent Character Recognition (ICR) – Also, targets handwritten print script or

cursive text one glyph or character at a time, usually involving machine learning.

- 4- Intelligent Word Recognition (IWR) – also targets handwritten print script or cursive text, one word at a time. This is especially useful for languages where glyphs not separated in cursive script.

OCR is generally an "offline" process, which analyses a static document. Handwriting movement analysis used as input to handwriting recognition. Instead of merely using the shapes of glyphs and words, this technique is able to capture motions, such as the order in which segments are drawn, the direction, and the pattern of putting the pen down and lifting it. This additional information can make the end-to-end process more accurate. This technology also known as "online character recognition", "dynamic character recognition", "real-time character recognition", and "intelligent character recognition".[4]

## **1.2 Motivation**

The Quran is the holy book for Muslims, and must preserved it from alterations and changes, for that we use modern technology to serve Quran.

In this proposal, we use optical character recognition for special Arabic font in auditing the Quran.

The goal of this research building computer software tool auditing electronic files that contain pages of Quran and compare it with their assets, before converting the files to the printing stage or publish the soft copy in order to reduce the time and manual effort during the audit of electronic files in the prepress stage. Detect errors that may appear, and documenting their location on each page electronically.

### 1.3 Research Objectives

The main objective of implementing an automatic Quran auditing is to Auditing electronic files of the Quran using optical character recognition for special Arabic font.

The various benefit of this tool includes:

1. Reduce the time and manual effort during the audit of electronic files in the print stage.
2. Detect errors that may appear, and documenting their location on each page.

### 1.4 Scope of this Work

The research is designed to classify and recognize a scanned image containing pages from Quran while dealing with the specialties of the Arabic language in Holy-Quran, some difficulties arises due to the differences between Quran written and the standard Arabic. In this thesis, the Holy Quran (برواية حفص عن عاصم) as diacritic standard for Arabic language text Adopted. We use Holy-Quran printed in king Fahd. Write in Uthmaani Font (الرسم العثماني).

With the recent increased computing power of modern computers, research in Auditing electronic files of the Holy-Quran focus mostly on finding a number of errors in each page and places of existence. Therefore, this research will not deal with time-response where it be proposed as a further work but will deal with number of errors in each page.

This project is software based system and not restricted by any hardware

implementation. Thus, only MATLAB coding is used .

## 1.5 Contribution

The following points summarize the main contributions in this thesis:

- Using optical character recognition to audit the electronic files of the Quran in special font "Uthmaani " .
- In pre-process stage there is a skeleton for the image, due to the absence of a base line in Uthmaani Font ( الرسم العثماني ), special algorithm built to find the base line in this thesis .
- In the segmentation stage, due to the method of drawing characters in Uthmaani Font, which is different from other Arabic fonts, special algorithm, built for segment words and Sub-words into character.
- Also in the segmentation stage, the one character segmented into three levels, the character, diacritics, and signs of adjustment.
- In auditing stage, compare the output of OCR system with Printed of king Fahd Glorious Quran Printing Complex Saudi Arabia, Madinah as a reference.

## 1.6 Theoretical Background

Optical character recognition (OCR) is an important research area in the field of pattern recognition. The objective of an OCR system is to recognize alphabetic letters, numbers, or other characters, which are in the form of digital images, without any human intervention [5].

This accomplished by searching a match between the features extracted from the

given character's image and the library of image models. Ideally, we would like the features to be distinct for different character images so that the computer can extract the correct model from the library without any confusion. At the same time, we also want the features to be robust enough so they not affected by viewing transformations, noises, resolution variations and other factors. Figure 1.1 illustrates the basic processes of an OCR system [5].

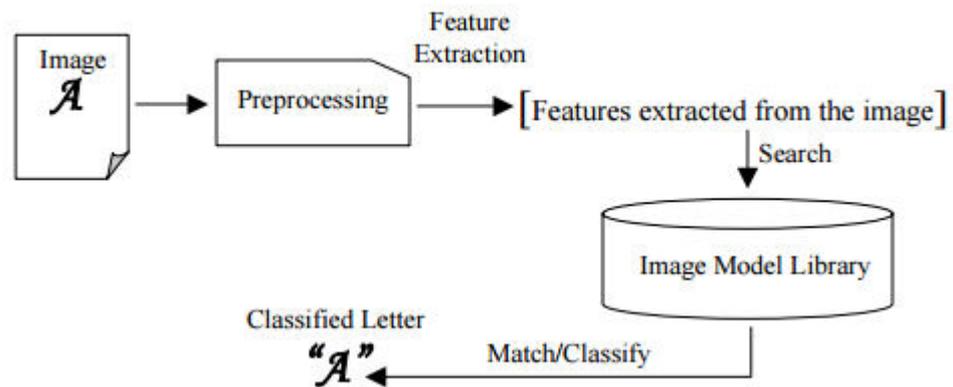


Figure 1.1: The basic processes of an OCR [6]

### 1.6.1 Computer Vision

One important branch of Artificial intelligence is computer vision, as mentioned in Figure 1.2 that shows few sub-branches of computer vision, the area of research which aims to imitate human vision and forms the basis of all image acquisition, its processing, document understanding and recognition [1].

Computer vision relies on a solid understanding of the physical process of image formation to obtain simple inference from individual pixel values like shape of the object and to recognize objects using geometric information or probabilistic techniques [2].

In its own turn, document understanding is a vast and difficult area for the focus of research today lies in being able to make content based searches which hope to allow machines to look beyond the keywords, headings or merely topics to find a piece of information. A far more streamlined field of document recognition and understanding is Optical Character Recognition which attempts to identify a single character from an optically read text image as a part of a word that can be then used to process further information on. The area gains rising significance as more and more information each day needs to be stored processed and retrieved rather than being keyed in from an already present printed or handwritten source [2].

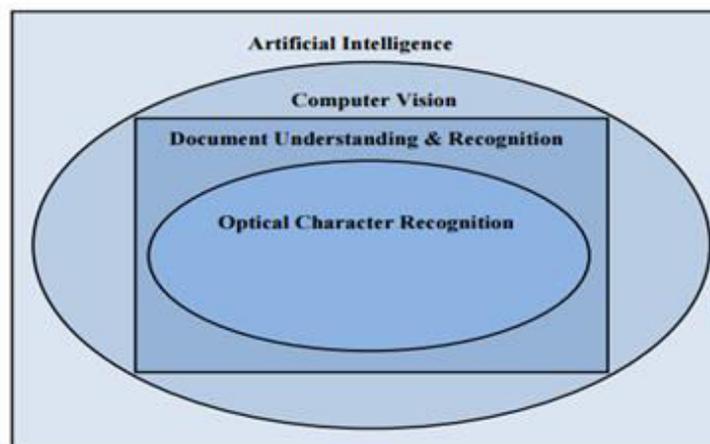


Figure 1.2: Subfields of Artificial Intelligence [13]

### 1.6.2 Character Recognition

Character recognition is a sub-field of pattern recognition in which images of characters are recognized and as a result of recognition respective character codes are returned, these when rendered give the text in the image.

The problem of character recognition is the problem of automatic recognition of raster images as being letters, digits or some other symbol and it is like any other

problem in computer vision [7].

Character recognition further classified into two types according to the manner in which input is provided to the recognition engine. Considering Figure 1.3 that shows the classification hierarchy of character recognition, the two types of character recognition are:

- a. On-line character recognition
- b. Off-line character recognition

### **1.6.3 Different Families of Character Recognition**

Figure 1.3 shows the different families of character recognition. Two different families are included in the general term of character recognition [8]:

- On-line character recognition
- Off-line character recognition

On-line character recognition deals with a data stream which comes from a transducer while the user is writing. The typical hardware to collect data is a digitizing tablet which is electromagnetic or pressure sensitive. When the user writes on the tablet, the successive movements of the pen are transformed to a series of electronic signal which is memorized and analyzed by the computer [9].

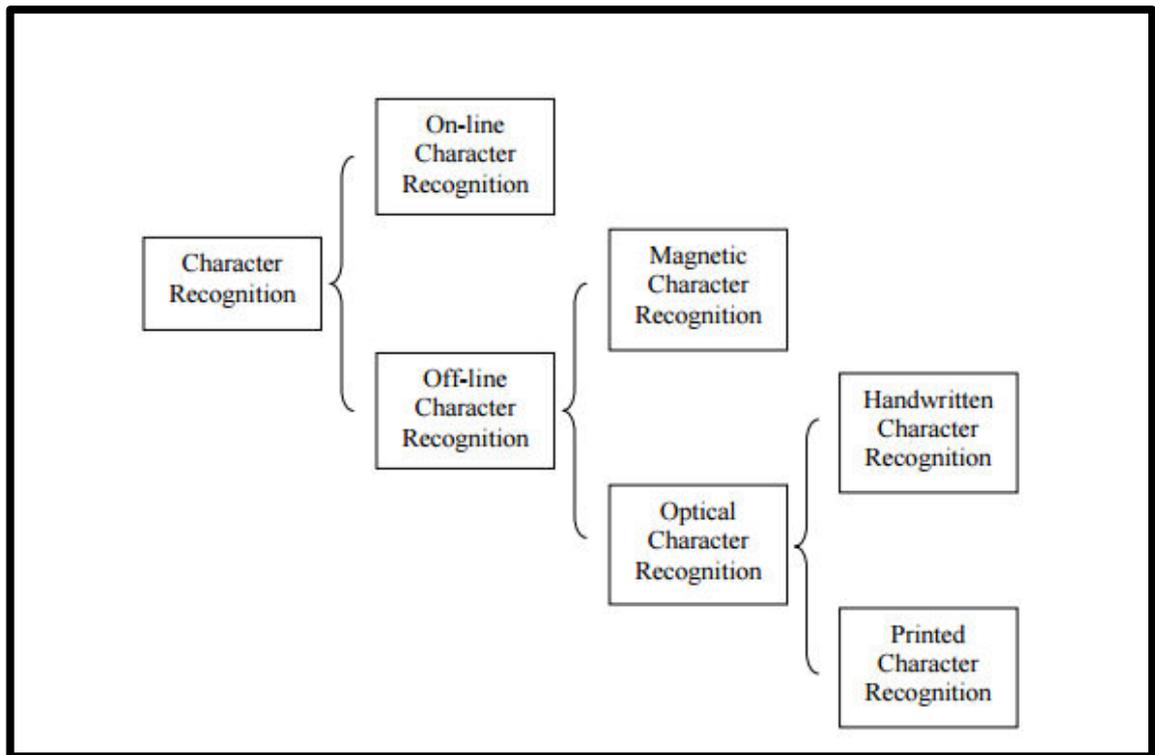


Figure 1.3: The different families of character recognition. [6]

Online systems obtain the position of the pen as a function of time directly from the interface. This is usually done through pen-based interfaces where the writer writes with a special pen on an electronic tablet [9].

Dynamic information, which are usually available for online text recognitions, are number of strokes, order of strokes, direction for each stroke, and speed of writing within each stroke [6].

This valuable information assists in recognition of documents and frequently leads to better performing systems compared to offline recognition. Some of applications of online optical recognition are in PDAs, smart phones, and Tablet computers. The advantage of online recognition system over offline systems is interactivity, adaptation of writer to digitizer (or vice versa), less prone noise, and available temporal information. The disadvantage is that the whole document is not available for

processing; and therefore, information needs to be processed dynamically [9].

Off-line character recognition is performed after the writing is finished. The major difference between online and offline character recognition is that on-line character recognition has time-sequence contextual information but off-line data does not. This difference generates a significant divergence in processing architectures and methods.

The off-line character recognition can be further grouped into [10]:

- Magnetic character recognition (MCR)
- Optical character recognition (OCR)

In MCR, the characters are printed with magnetic ink. The reading device can recognize the characters according to the unique magnetic field of each character. MCR is mostly used in banks for check authentication [10].

OCR deals with the recognition of characters acquired by optical means, typically a scanner or a camera. The characters are in the form of pixelized images, and can be either printed or handwritten, of any size, shape, or orientation [11].

The OCR can be subdivided into handwritten character recognition and printed character recognition. Handwritten character recognition is more difficult to implement than printed character due to the diversified human handwriting styles and customs. In printed character recognition, the images to be processed are in the forms of standard fonts like Times New Roman, Arial, Courier, etc [12].

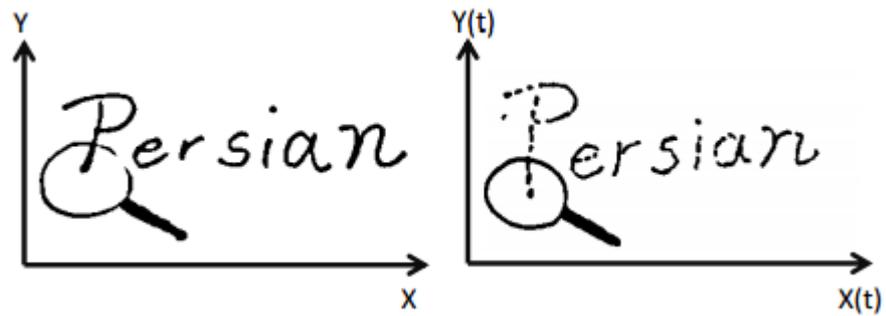


Figure 1.4: Offline text containing just special information (left), online text containing temporal sequence of points traced out by the pen (right) [14].

## 1.7 Optical Character Recognition Process

The process of converting documents into electronic forms, which usually referred to as digitization undertaken in different steps. The process of scanning a document and representing the scanned image for further processing called the pre-processing or imaging phase. The process of manipulating the scanned image of a document to produce a searchable text called the OCR processing stage [14].

### 1.7.1 The Imaging Stage

The imaging process involves scanning the document and storing it as an image. The most popular image format used for this purpose is called Tagged-Image File Format (TIFF) [14].

### 1.7.2 The OCR Process Stage

The major steps of the OCR processing stage are shown below.

### **Distinguishing between text and images – Segmentation**

In this step, the process of identifying the text and image blocks of the scanned image is undertaken. The boundaries of each image analyzed in order to recognize the text.

### **Character recognition – feature extraction**

This step involves recognizing a character using a method known as feature extraction. OCR tools store rules about the characters of a given script using a method known as the learning process. A character then identified by analyzing its shape and comparing its features against a set of rules stored on the OCR engine that distinguishes each character.

### **Recognition of words**

Following the character recognition process, word identification process is performed by comparing the string of characters against an existing dictionary of words. Additional processes such as spell-checking are performed under this step.

### **Correction of unrecognized characters – error correction**

In this step, the user allowed to provide corrections to unrecognized characters.

### **Output formatting**

The final step involves storing the output in one of the industry standard formats such as RTF, PDF, WORD and plain UNICODE text.

The input of an OCR system is an image. The image can be from a scanner, a camera, or simply a print-screen of a page. The image may contain not just the text but pictures, equations, tables, etc. Therefore, the first step of any OCR system is pre-

processing.

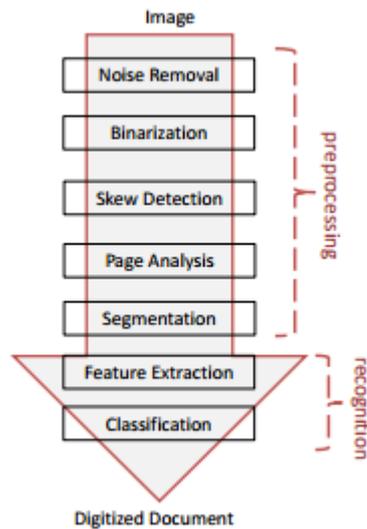


Figure 1.5: Offline Optical Character Recognition Classic process flow [14]

The goal of pre-processing is to extract the text and convert the raw image of the text to segmented components. Then, in the recognition phase, the features of these segmented components are extracted and fed into a classification module. Finally, the outcome of the algorithm is a machine editable text (Figure 1.5).

### 1.7.3 Preprocessing Phase

Typical pre-processing includes the following steps, not necessarily in this order:

- **Scanning**

A flatbed scanner usually used at 300dpi, which converts the printed material on the page scanned into a bitmap image.

- **Document image analysis**

The bitmap image of the text is analyzed for the presence of skew or slant and

consequently these are removed. Quite a lot of printed literature has combinations of text and tables, graphs and other forms of illustrations. It is therefore important that the text area is identified separately from the other images and could be localized and extracted [14].

- **Noise removal**

Documents may contain noise for many reasons. One type of noise is the marginal noise which appears as a large dark region around the document image. Another type of noise is background noise which appears from uneven contrast, background spots, printer and scanner malfunctions. Finally, page rule lines are another source of noise interfering with text objects. There are several methods in dealing with each noise type [15].

For page rule lines, mathematical morphology based methods trace line like structures as candidate for rule lines. Hough transform [16] can also be used to find imperfect instances of objects with certain class of shapes using a voting procedure. Finally, projection profile based methods work by creating a horizontal histogram in which the hills of the histogram are the center locations of the horizontal rule lines [17].

For marginal noise, there are two groups of methods. One group aims at identifying noise components. In this method, the noise patterns in an image are searched by extracting noise features [18]. For example, Peerawit method [19] uses Sobel edge detection and identifies noises to be removed by comparing the edge density of marginal noise and text. Another group for marginal noise removal is by Identifying Text Components [20].

For background noise, many techniques have been introduced. One common

method is to use a low-pass filter to remove as much of the noise as possible while retaining the entire foreground pixels. More advanced methods are Binarization and Thresholding Based Methods [21], Fuzzy Logic Based Methods [22], Histogram Based Methods [23], and morphology based methods [24].

- **Binarization**

Binarization is the process of converting gray scale images to binary images. A large number of methods have been proposed for binarization. These methods generally fall into two main approaches. One approach is based on global thresholding and the other is based on local thresholding [25].

In global thresholding, based on statistical attributions of a document, a single value is used as threshold between background and foreground pixels. Otsu method [25] is one of the most commonly used global binarization techniques. The main drawback of this method is that it cannot adapt well to noise and illuminations. A recent work by Lazzara [26] focuses on Sauvola binarization method. This method performs relatively well on classical documents, however, three main defects remain: the window parameter of Sauvola formula does not fit automatically to the contents, it is not robust to low contrasts, is not invariant with respect to contrast inversion.

In local thresholding, the threshold value is varied based on the local content of the image. Commonly used Niblack binarization method [27] is based on local thresholding.

- **Skew detection and correction**

A text document consists of several text lines. To estimate the skew angle of a text line,

a straight line can be drawn through its characters. The angle of this straight line with the horizontal edges of the page is the skew angle of the text line (Figure 5) [14].



Figure 1.6: A skewed text line [14]

The dominant skew angle of the text lines in a page determines the skew angle of that page. A document originated electronically with a text editor has skew angle of zero. However, when a document is printed, photocopied or scanned, a non-zero skew angle will be introduced [14].

Since document analysis, algorithms such as text recognition or page layout analyzers usually assume a zero skewed page, skew detection and correction is considered a required preprocessing step. Moreover, to improve the quality of scanned documents, many scanners perform document skew correction immediately after a scan and before a document image is displayed on a computer monitor [14].

#### **1.7.4 Segmentation**

Segmentation performance of an OCR system directly affects the recognition performance, as the output of the segmentation step is directly fed into the recognition engine. Figure 6 depicts hierarchy of offline segmentation step

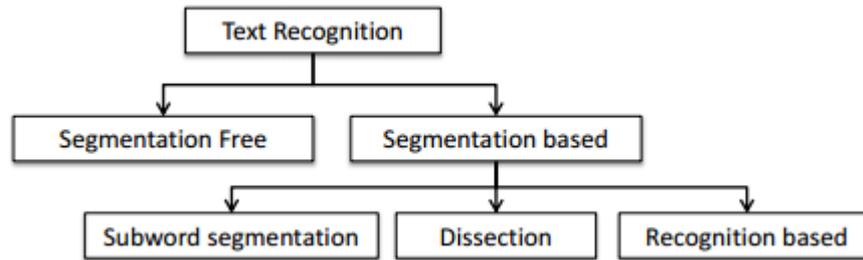


Figure 1.7: Hierarchy of offline segmentation step of OCR [14].

Text segmentation of Persian text is not a trivial step as characters could overlap, slant, and have different styles and fonts. Some researches skip the segmentation step entirely and instead take a holistic approach. The idea is to recognize whole words against a dictionary. The obvious problem of such an approach is the number of classes present in the recognition phase. In Persian there are 114 contextual forms of its 32 alphabets. Characters in the Persian language can have different shapes depending on its position within a word [14].

A character that is used in the beginning of a word will have a different appearance than one that is in the middle or end of a word and these might be different in appearance than stand-alone characters. A technique that segments a Persian word into characters and then uses a classifier trained on all 114 shapes can potentially recognize any text [14].

On the other hand, a holistic classifier needs to learn as many classes as the number of dictionary words, names of individuals and countries. Therefore, in holistic approach, training of a classifier for many classes is one of the major issues. Based on this limitation, segmentation based approach is more practical than the holistic approach for real world problems. Many techniques have been developed for holistic approaches [14].

Benouaret [14] used holistic method for Arabic word recognition. To build a feature vector sequence, two segmentation schemes are incorporated to divide a word into frames. The first one is uniform segmentation, which vertically divides a word into equal sized frames. The second one is non-uniform segmentation, which has variable frame size. After segmenting word into frames, statistical and structural features are extracted by capturing ascenders, descenders, concavity, dots, and stroke direction. Vinciarelli [28] used similar technique in which prior to the information retrieval, the individual words on the handwritten document need to be recognized correctly. Using a fixed size sliding window, density feature is extracted for HMM to perform recognition by calculating the likelihood of a word against dictionary. To reduce the computational cost of holistic approach, Mozaffari [29] proposed a lexicon reduction scheme for offline Farsi handwriting recognition by analyzing dots within characters.

Segmentation based approaches, can be performed by either the dissection or recognition based technique. Dissection is decomposition of the image into a sequence of sub-images using general features. Projection analysis, connected component processing, and whitespace are some of the common dissection techniques used by OCR systems [30]. These techniques are suitable for scripts which have spaces between characters.

The basic principle of recognition-based character segmentation is to use a mobile window of variable width to provide the tentative segmentations. Characters are byproducts of the character recognition for systems using such a principle to perform character separation. The main advantage of this technique is that it bypasses serious character separation problems [30].

In some language scripts, like Persian, segmenting words to characters is a very

difficult task as characters overlap. For this reason, in chapter 4 in segmenting Persian texts, we neither segment words to their characters nor skip the segmentation step entirely. Instead, we try to get the best of each method by segmenting text to its sub-words, a much easier process with a much better success rate. On the other hand, rather than dealing with a huge class size, we deal with a more manageable database of sub-words. The largest Persian sub-words dictionary reported in literature contains 7317 sub-words, which is just a small fraction of more than a million words in Persian language [31].

### **1.7.5 Recognition Phase**

In this phase, segmented text is fed into a feature extraction algorithm and finally to classification module.

- **Feature extraction**

Feature extraction is to find a set of features that define the shape of the underlying character as precisely and uniquely as possible. Selection of feature extraction method is probably the most important factor in achieving high performance in recognition. Feature extraction methods are very much application specific and there is no universally accepted set of feature vectors in document image recognition. Some of the available methods in feature extraction include image invariant, projection histograms, zoning, and n-tuples. Image invariant features are popular choice in many OCR systems. Image invariant methods can be categorized to boundary-based and region-based methods [32].

A classical boundary-based method is Discrete Fourier Transform. In this method, Fourier transform is used to analyze a closed planar curve. Several variations of Fourier Transform for feature extraction have been introduced. For examples, Zahn applies the

Fourier transform to the sequence of angular differences between line segments in the curve [32].

In region-based methods, moments of different points present in a character are used as a feature. Hu [33] introduced the concept of classical moment invariant in 1962. Hu's other moments are statistical measure of the pixel distribution about the center of gravity of the character. In 1982, Teh [34] defined a set of moments called Zernike based on theory of orthogonal polynomials.

- **Classification**

In classification stage, based on the features extracted and relationships among the features, an OCR process assigns labels to character images. This step is the final stage of the OCR system in which characters or sub-words are recognized and are output to machine editable form. Classification methods can be divided into two categories: learning-based and non-parametric classifiers. The learning-based classifiers require an intensive learning phase of the classifier parameters. Neural network, Support Vector Machine (SVM), Boosting, and decision tree are examples of commonly used classifiers in this category [34].

## 1.8 World Languages and Scripts

Communication around the world takes place in more than several hundred languages today. There is a great variety of ways in which these languages are written down but it has been found out that more than 90 languages use the Latin script to scribe their words, English being one of them. There are several other scripts that serve as means to write down a combination of languages. The Arabic script stands second to

Latin as it has been adopted by more than 25 different languages to form their alphabet [35].

According to the way the script is written down and the patterns it follows helps us divide them in two separate categories as Figure 1.8 shows.

- Non-cursive scripts.
- Cursive scripts.

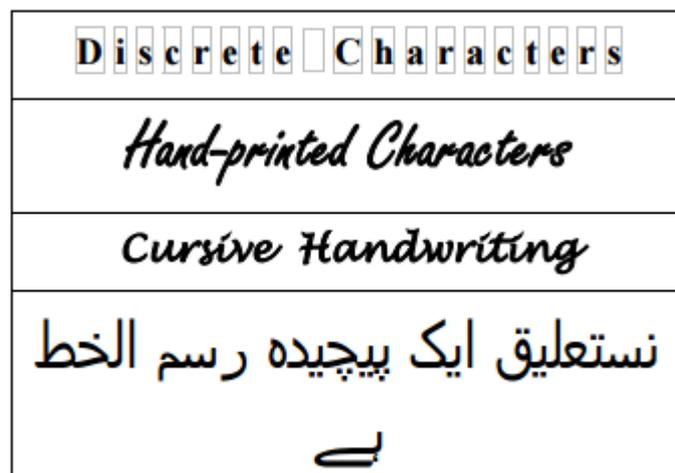


Figure 1.8: Cursive and non-cursive scripts. [13]

### 1.8.1 Non-Cursive Script

The non cursive scripts which are more common are inherently discrete as far as the printed text is concerned. This means that each character has a separate and a definite shape that combines with the next one by being placed side by side with it and never overlapping or shadowing the preceding or succeeding letters. However, when these scripts are handwritten the scribe's hand can make the letters decorative, cursive and more flowing in form and shape. The Latin script is an example of such a script where

handwritten text can be as decorative and cursive as the writer's choice while printed text is easily recognizable [35].

### **1.8.2 Cursive Script**

On the other hand the naturally cursive scripts e.g. Arabic have a unique feature for the formation of words. The characters in these scripts are not discrete but are joined to each other to form ligatures and then words. These free flowing character forms create words by overlapping each other sometimes even stacking on each other vertically [35].

The non-discrete nature of these scripts makes them ever more difficult to be developed as font types for printing as well as pose challenges for character recognition. Creating discrete characters for text processing for the cursive scripts and placing them side by side to form words like discrete scripts for convenience in character recognition [35].

### **1.8.3 Arabic Script**

Arabic is a popular script. It estimated that there are more than one billion Arabic script users in the world. However, there is no OCR for Arabic, at least commercial OCR, so, If OCR systems are available for Arabic characters, they will have a great commercial value. The leakage of Arabic OCR is due to the complexity of Arabic character and text nature , some of these difficulties are listed below [36].

Challenging points in Arabic scripts: There are various challenges to deal with Arabic Script that include:

- 1- Arabic is written from right to left.

2- It has 28 characters. The shape of the character varies according to its position in the word. Each character has either two or four different forms. Obviously, this will increase the number of classes to be recognized from 28 to 112, however, there are 6 characters can be connected only from the right, these are: ( ا, د, ذ, ر, , ) which will reduce the number of classes to 100. In addition to the 28 characters there are some special characters such as 'ى'.

3- Arabic always written cursively.

4- Words separated by spaces. Clearly, the above six characters, if appeared in a word, will cause the word to be divided into blocks of connected components called sub-words. Thus, a word can have one or more sub-words. Sub-words are also separated by spaces, but usually shorter than the ones between words. So, this issue needs to be considered to avoid segmenting a word into two [36].

Examples of words in which all characters are connected: ( كُتِبَ , مُحَمَّدٌ , خَشَبٌ )

Examples of words consist of sub words: ( سَأَلٌ , بَاعٌ , وَرُودٌ )

5- Arabic characters 'normally' connected on an imaginary line called baseline, as shown in Figure 1.9. This line is as thick as a pen point and much less than the width of the beginning character [36].

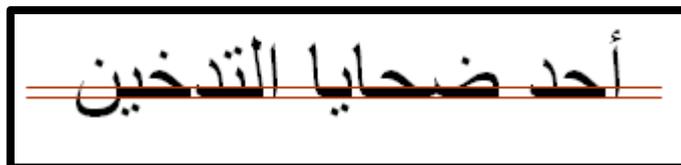


Figure 1.9: The base line [13]

- 6- The length of an Arabic character is variable; see for example (ك ) and (ل ) . They differ also in height, for instance: (ا ) and (ل ). Furthermore, the width and height vary across the different shapes of the same character in different positions in the word. It is maximal either when the character is situated at the end of the subpart or when it is an isolated character. For example: (ب ) and (ـب ). Another example is the difference Between (ح ) and (ـح ) in terms of height. Hence, segmentation based on a fixed size width (sometimes called pitch segmentation) is not applicable to Arabic.
- 7- Arabic writing uses many fonts and writing styles. Some characters, may overlap with their neighboring characters forming what is called "ligature" in which the second character may starts before the end of the first one or even before the beginning of it, see Figure 1.10 Most of the researchers considered the
- 8- Ligature as one character, although that will increase the number of classes, Figure 1.10 illustrate this concept.

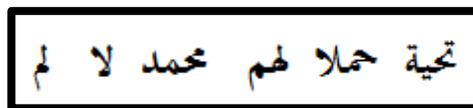


Figure 1.10: Ligatures

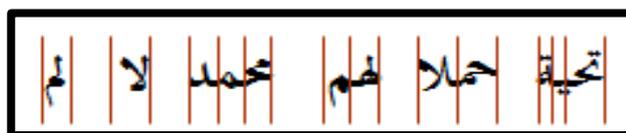


Figure 1.11: Segmenting ligature as one character.

## 1.9 Thesis Roadmap

This thesis contains five chapters including the introduction chapter. Each chapter is subject to certain scopes, which formulates the thesis contents. Below are the chapter numbers, titles and summaries of each documented one in this thesis.

### Chapter 1: Introduction and Theoretical Background

Chapter 1 presents the definition and brief background of the project. It also includes the problem statement, research objectives, scope of research and thesis roadmap of the thesis.

In part two of this chapter, highlights the theoretical concepts, systems, tools, and techniques that are related to this research. Its start with computer vision, and then it moves to the optical character recognition OCR. A quick description about Arabic Language is mention. This chapter also include the type of OCR system. Finally, it represents the quality evaluation concept.

### Chapter 2: Literature Review

Chapter 2 shows some related researches, algorithms and techniques that are related and relevant to this research.

### Chapter 3: Holy-Quran auditing system

Chapter 3 introduces Holy Quran auditing system. Through this chapter OCR stage are detailed Pre-processing, Segmentation, Recognition, and comparison with reference. Each part discussed with algorithms and charts in details in this chapter.

## Chapter 4: Evaluation and Results

Chapter 4 introduces the evaluation methods that applied in thesis. Evaluation and testing producers executed on sample to determine the performance of the overall auditing system. Finally, the obtained result analyzed and discussed.

## Chapter 5: Conclusion and future works

Chapter 5 summarizes the work accomplished and discusses obstacle. The recommendation for the future research highlighted.

### **1.10 Summary**

In general, a cursive word recognized through a hierarchical analysis; a word decomposed into letters, letters into strokes that presents the segmentation part. Before that pre-processing stages are assumed done, these stages may include format conversion, grayscale conversion, filtering, Smoothing and Sharpening, change dpi, rotate and skew and skeletonization. Finally, recognition part starts to identify each segmented character.

Arabic Script is a cursive-type language, which written from right to left, and so recognition should occur this way. There are 28 characters in the Arabic alphabet. Each character has two to four different forms, which depends on its position in the word or sub words. As a result, there are 100 classes to be recognized.

## CHAPTER 2

---

### Literature Survey

#### 2.1 Introduction

The overwhelming volume of paper-based data in corporations and offices challenges their ability to manage documents and records. Computers, working faster and more efficiently than human operators, can be used to perform many of the tasks required for efficient document and content management. Computers understand alphanumeric characters as ASCII code typed on a keyboard where each character or letter represents a recognizable code. However, computers cannot distinguish characters and words from scanned images of paper documents. Therefore, where alphanumeric information must be retrieved from scanned images such as commercial or government documents, tax returns, passport applications and credit card applications, characters must first be converted to their ASCII equivalents before they can be recognized as readable text. OCR allows us to convert a document into electronic text, which we can edit and search etc [37].

Optical character recognition for English has become one of the most successful applications of technology in pattern recognition and artificial intelligence. OCR is the machine replication of human reading and has been the subject of intensive research for 12 more than five decades [37].

While a phenomenal amount of research in IT has made Roman script languages extremely adaptable in all areas of computing – insubstantial work for Arabic in this area accounts for very little computerization in this script.[13]

The Arabic language is considered to be a difficult one with a much richer alphabet than the Latin, the form of the letter is the function of its position in the word: isolated, initial, medial or final, it changes its shape depending upon its position, and each shape has multiple instances, words are written from left to right (LTR) [38].

Chapter 2 shows some related researches, algorithms and techniques that are related and relevant to this research. Focus on Arabic OCR.

## 2.2 English language OCR system

Character recognition originated as early as 1870 when Carey invented the retina scanner, which is an image transmission system using photocells. It used as an aid to the visually handicapped by the Russian scientist Tyurin in 1900. However, the first generation machines appeared in the beginning of the 1960s with the development of the digital computers. It is the first time OCR realized as a data processing application to the business world. The first generation machines characterized by the “constrained” letter shapes, which the OCRs can read. These symbols specially designed for machine reading, and they did not even look natural. The first commercialized OCR of this generation was IBM 1418, which designed to read a special IBM font, 407. The recognition method was template matching, which compares the character image with a library of prototype images for each character of each font.[69]

Nowadays, there is much motivation to provide computerized document analysis systems. OCR contributes to this progress by providing techniques to convert large volumes of data automatically. A large number of papers and patents advertise recognition rates as high as 99.99%; this gives the impression that automation problems seem to have been solved. Although OCR is widely used presently, its accuracy today is still far from that of a seven-year old child, let alone a moderately skilled typist. Failure of some real applications show that performance problems still exist on composite and degraded documents (i.e., noisy characters, tilt, mixing of fonts, etc.) and that there is still room for progress [69].

Various methods have been proposed to increase the accuracy of optical character recognizers. In fact, at various research laboratories, the challenge is to develop robust methods that remove as much as possible the typographical and noise restrictions while maintaining rates similar to those provided by limited-font commercial machines [69] .

## 2.3 Approaches for printed Arabic script OCR

Arabic language characters have features that make direct application of algorithms for character classification in other languages difficult to achieve as the structure of Arabic is very different [39].

Extensive literature survey on Arabic script OCR showed that the researchers in this area followed mainly two different approaches for the implementation of an OCR system for printed Arabic script text, namely

- Segmentation-based techniques [39].

- Ligature-based approaches [39].

Ligatures are very common. In font terms, a ligature is a single glyph that represents more than one underlying character.

The example shown in Figure 2.2 is of a mandatory ligature in Arabic. A “LAM” (ﻻ) character followed by an "ALEPH" (ﺍ) character must always be displayed as a single lam-alef shape. Note carefully, however, that you should continue to use two characters in memory to represent this sequence: a LAM and an ALEPH.

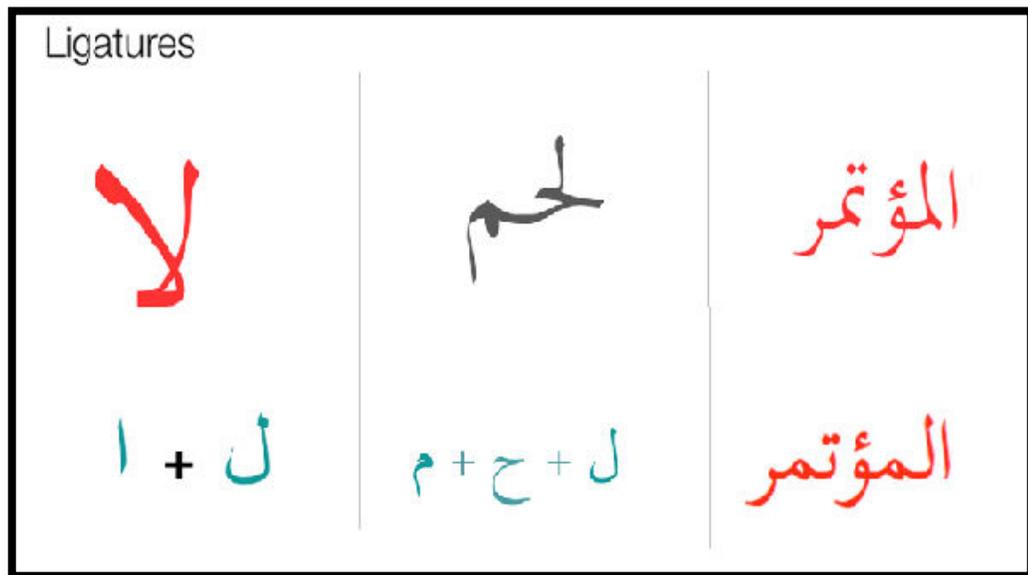


Figure 2.1 Ligature-based approaches [39]

In fact, while some fonts represent lam-alef using a single ligature glyph, others combine smaller partial glyphs to achieve the same effect. We will use the term 'ligature' here in a very loose sense to mean a combination of characters that are displayed as what, to the reader, looks like a single shape.

### **2.3.1 Ligature-based approach**

In the ligature-based approach for the implementation of Arabic script OCR there are only two levels of segmentation at the end of which the system gets isolated character shapes or the ligatures segmented into lines of text [39].

### **2.3.2 Segmentation-based approach**

If the segmentation-based approach is followed for the implementation of Arabic script OCR then at the recognition phase all the text images segmented till the character level will have to be needed, similar to a Latin script OCR, that is all the ligatures in the words are segmented into their constituent character shapes [39].

## **2.4 Previous work in ligature-based Arabic OCR**

Here we present a brief overview of the previous work that has been done on ligature based Arabic OCR.

Al-Badr and R. Haralick [40] highlight some of the hindrances in the development of an Arabic OCR and the reasons for inadequate research in this area. It attributes the difficulties in Arabic character recognition to the more complex features of the Arabic script e.g. cursiveness, vertical stacking of characters to form many of the ligatures, context-sensitivity, vertical overlapping of shapes etc. The paper discusses the design and implementation of an Arabic Word Recognition system which works on the principles of

symbol recognition without initially segmenting words into characters, claiming that most recognition errors occur at the crucial stage of segmentation because of the typical shape combinations of characters in the Arabic script.

The system first recognizes the input word by detecting a set of ‘shape primitives’ on the word. It then matches the regions of the word with a set of symbol models. The recognized word is thus presented in the form of a spatial arrangement of symbol models matching the region of the word. Since the possible combinations of symbol models are potentially large, the system imposes constraints in terms of word structure and spatial consistency. The accuracy of the system is shown to be 94.1% for isolated, scanned symbols and 73% for scanned words [40].

Pechwitz and Margner [41] present an off-line recognition system for Arabic hand written text. The work highlights the efficiency of cursive text recognition methods based on Hidden Markov Models (HMMs). The study was conducted on a semi-continuous, one dimensional HMM and describes in detail the modification and adaptation of the preprocessing and feature extraction processes for recognition of Arabic writing. The experiments were based on first estimating the normalization parameters of each binary word image and following it by normalization of height, length and baseline skew. The features are then collected using a sliding window technique.

Alma’adeed et. al. [42] present a complete scheme for character recognition of totally unconstrained Arabic text based on a Model Discriminant HMM. The system proposes feature extraction following the removal of variations in the word images which do not

affect the identity of the written word. The system then encodes the skeleton and edges of the word and a classification process based on the HMM is used. The result is a word matching one in a dictionary. The study gives indication of successful results of a detailed experiment.

Alma'adeed et. al. [43] present a scheme to recognize hand written Arabic text. The overall engine is modeled on the basis of multiple HMMs and a global feature extraction scheme. The system initially removes variations in the word images and then encodes the skeleton and edge of the word for feature extraction. A rule based classification is then used as a global recognition engine. Finally for each group, the HMM approach is used for trial classification. The given output is a word that matches one present in a lexicon. Once the model has been established the Viterbi algorithm is used to recognize the segments of letters composing a word. The study gives details about the segmentation step as well as off-line recognition operations. The study emphasizes the development of two substantially different recognition engines because there are multiple ways in which one Arabic word can be written down. The first engine is a global feature scheme using some ascender and descender features and making use of a rule-based classification engine. The second scheme is based on a set of features using a HMM classifier.

Farah [44] presents his work that implies the construction of a recognition system around a modular architecture of feature extraction and word classification units. This was done in the attempt to solve the problem of recognizing handwritten Arabic bank checks. The research stresses the efficiency of a multi classifying system with three parallel classifiers working on the same set of structural features. The classification stage results are

first normalized and after using contextual information present in the syntactical module the final decision on the candidate words is made.

El-Hajj [45] gives the description of a one dimensional HMM off-line handwriting recognition system using an analytical approach. Specific models are used for each character and word models are built by concatenating the appropriate character models. The system is supported by a set of robust language independent features extracted on binary images.

The study lays focus on baselines as an important feature in character recognition. The baseline dependant features are then added to the original set of features. Feature vectors are extracted using the sliding window technique.

Alaa Hamid and Ramzi [46] present a technique to segment hand written Arabic text through a neural network. These include three initial steps to achieve the end result before the Artificial Neural Network (ANN) verification: scanning, binarization and finally feature extraction. A recursive, conventional algorithm is used to segment text into connected blocks of characters and generate pre-segmentation points for these blocks. This heuristic algorithm is responsible for generating the topographic features from the text and calculating presegmentation points. An Artificial Neural Network then verifies the accuracy of these segmentation points. The results have shown an accuracy range between 52.11% and 69.72% considering various features. The inaccuracies have been attributed to complexities in shapes of characters or due to dislocated external objects.

Khorsheed [47] proposes a segmentation free approach to recognize Arabic text using the HMM toolkit, a portable toolkit for building and manipulating Hidden Markov Models.

It decomposes the document image into text line images and extracts a set of simple statistical features using the sliding window technique. The Hidden Markov Model Toolkit is then used to develop a sequence of training images and finally for recognition of characters. This implies that the feature vector, extracted from the text, is computed as a function of an independent variable. The experiments were initially conducted on a corpus of data collected in the Arabic font 'Thuluth' and later with others. Tahoma and Andulas scored highest recognition rates while Naskh and Thuluth performed lowest. The system was however capable of recognizing complex ligatures and overlaps and showed an overall improvement with the use of the tri-model scheme. Suggested improvements for future developments are expansion of the data corpus and utilization of HTK capabilities.

Erlandson et al [48] implemented a word-level Arabic text recognition system that did not require character segmentation. They characterized the shape of Arabic words by unique feature vectors. These feature vectors were then matched against a database of feature vectors derived from a dictionary of known words. The database stored multiple feature vectors for each word in a dictionary of 48,200 words. The word whose feature vectors strongly matched was returned as the hypothesis.

Al-Badr and Haralick [49] designed and implemented an Arabic word recognition system that recognized an input word by detecting a set of shape primitives on the word. The regions of words represented by these shape primitives were then matched with a set of symbol models. The description of the recognized word was obtained from a spatial arrangement of symbol models that were matched to regions of the word.

Bazzi et al [50] implemented a segmentation free OCR system that was based on Hidden Markov Model (HMM). They chose a text line as a major unit for training and recognition. A page of printed text was decomposed into a set of horizontal lines using horizontal position along each line as an independent variable. Hence they scanned a text line from right to left and at each horizontal position a feature vector was computed from a narrow vertical strip of input. The system was based on 14-state HMMs of each character. The output of the system comprises sequences of characters that had the maximum likelihood.

Obaid and Dobrowiecki [51] proposed a segmentation free method called N-markers for recognition of printed Arabic text. Their method was a mixture of global and structural approaches. They collected the informative points lying in the center of the characters. These points were the basis of the coordinate system for the configuration of sensors designed to identify the necessary strokes and were called N-markers. By distributing enough markers over the character, a letter or a group of letters in a text line was detected.

Reza Safabakhsh et al [52] proposed a system for Farsi Nastalique handwritten words recognition using a continuous-density variable-duration hidden Markov model CDVDHMM. In this system after the pre-processing stage the ascenders, descenders, dots and other secondary strokes are eliminated from the original image. Segmentation is done by analyzing upper contour thus avoiding the under-segmentation problem. Variable-duration states in the 48 system cover the over-segmentation problem. Features are extracted which are invariant to size and shift. At the recognition stage a modified version of Viterbi algorithm is used.

Mohammad S Khorsheed et al [53] introduced a holistic approach for Arabic word recognition which uses a normalization process to compensate for dilation and translation. The process adapted transforms the image of an Arabic word from Cartesian coordinates to polar coordinates similar to log polar transformation. Rotation is also converted into translation by this transformation.

## 2.5 Previous work in segmentation-based Arabic OCR

Ligature-based OCR has its limitation and cannot be expected to identify all possible ligatures because it needs a prohibitively large database to store all possible combinations of characters that can form long ligatures. Since every ligature is composed of characters, the segmentation of ligatures could generate individual characters that can be recognized [39].

Machine generated (printed) Arabic (Naskh style) text usually follows a horizontal base-line where most characters, irrespective of their shape, have a horizontal segment of constant width. If we can separate these horizontal constant-width segments from a ligature, the remaining components of the ligature could be recognized. However, this is not the case with Nastalique which has multiple base-lines, horizontal as well as sloping, making Nastalique a complex style for optical recognition [39].

This section presents a brief overview of the previous work that has been done on segmentation-based Arabic OCR.

Bousslama and Kishibe [54] proposed a method that combined the structural and statistical approach for feature extraction and a classification technique based on fuzzy logic. They segmented characters into a main and complement characters. The main segment was then centred and projected horizontally and vertically. The features of classification were then extracted from the number of complement characters and from the horizontal and vertical projection profiles of the main character. A set of fuzzy rules was used for classification. The recognition algorithm was tested on three different fonts and high recognition rates were achieved.

Zidouri et al [55] proposed a sub-word segmentation technique that was independent of font type and font size. After applying pre-processing techniques, they employed horizontal and vertical segmentation to segment a page into separate lines and lines into sub-words respectively. To divide sub-words into characters, they first skeletonized the image of subwords without dots. Then for all the rows, they scanned the image row-wise from right to left, to find a band of horizontal pixels of length greater than or equal to the width of the smallest character. The vertical projection of this scanned band was then taken and if no pixel was found, a vertical guide band was drawn in an empty image. Thus several guide bands were drawn for all the rows. A special mark for the guide bands for each row was used below the location of the baseline. In order to select the correct guide band, several features were extracted and tested for several predefined rules. If it satisfied rules then it was selected, otherwise rejected.

Motawa et al [56] proposed an algorithm for automatic segmentation of Arabic words using mathematical morphology. They first digitized the image using 300 dpi scanner and then detected and corrected any slanted strokes. They applied erosion operation on the

image and computed the average slope of all the strokes. Every pixel was then transformed to a new location according to a formula to correct the slant. After slant correction, connected components were constructed which formed the skeleton for all future analysis of the image. Morphological operations, opening and closing, were applied to word image to allocate singularities and regularities. Singularities represent the start, the end or the transition to another character. Regularities contain the information required for connecting a character to the next character. So the regularities were the candidates for segmentation.

Tolba and Shaddad [57] proposed a segmentation algorithm for the separation of Arabic characters. In their algorithm, they slid a window over a word horizontally from right to left and at each instant they calculated a segmentation parameter which was then matched with predefined set of threshold values. If the segmentation parameter was less than the threshold value, the region was marked as a silence region. Detecting the silence region after the beginning of the letter identified the end of a letter. When the value of the segmentation parameter increased, the beginning of the next letter started.

Al-Yousefi and Udpa [58] introduced a statistical approach for Arabic character recognition. They used a two-level segmentation scheme. They first segmented the words into character. Then a lower level segmentation was applied to segment the characters into primary and secondary parts (dots and zigzags). Then they computed the moments of horizontal and vertical projections of the primary parts and normalized them to zero order moment. The features were extracted from the normalized moments of the vertical and horizontal projections and the classification of the primary characters was done using quadratic Bayesian classifier. The secondary parts were isolated and identified separately. Their recognition was done during pre-processing and segmentation stage.

Amin and Mari [59] proposed a structural probabilistic technique for automatic recognition of multi-font printed Arabic text that was based on character recognition and word recognition. They first transformed the image of text into separate lines of text by taking a horizontal projection and then segmented the text lines into words and sub-words by taking vertical projection. To segment words into characters, they took vertical projection of the word and the least sum of the average value over all columns showed the connectivity point. Thus each part of the word having a value less than the average sum was segmented into a different character. This resulted in the number of segments that were then connected together in the recognition phase to form the basic shape of the character and the segments that were not connected to any other segment were considered to be complementary characters. They used Freeman codes of the characters and consulted character recognition dictionary to recognize characters. The word recognition part of the technique used the tree representation lexicon and the Viterbi algorithm.

Al-Emami and Usher [60] presented an on-line system to recognize handwritten Arabic words. Words are segmented into primitives that are usually smaller than characters. The system is taught by being fed the specifications of the primitives of each character. In the recognition process, parameters of each primitive are found and special rules are applied to select the combination of primitives that best matches the features of learned characters. The method requires manual adjustment of some parameters. The system was tested against only 170 words, written by 11 different subjects for 540 characters.

Zahour et al [61] presented a method for automatic recognition of off-line Arabic cursive handwritten words based on a syntactic description of words. The features of a word are extracted and ordered to form a tree description of the script with two primitive classes:

branches and loops. In this description, the loops are characterized by their classes and the branches by their marked curvature, their relationship, and whether they are in clockwise or counterclockwise direction. Some geometrical attributes are applied to the primitives that are combined to form larger basic forms. A character is then described by a sequence of the basic forms. The reported recognition rate of the system is 86%.

Abuhaiba [62] presented a text recognition system, capable of recognizing off-line handwritten Arabic cursive text. A straight-line approximation of an off-line stroke is converted to a one-dimensional representation. Tokens are extracted from this onedimensional representation. The tokens of a stroke are re-combined to meaningful strings of tokens. Algorithms to recognize and learn token strings were presented. The process of extracting the best set of basic shapes that represent the best set of token strings that constitute an unknown stroke was described. A method was developed to extract lines from pages of handwritten text, arrange main strokes of extracted lines in the same order as they were written, and present secondary strokes to main strokes. Presented secondary strokes are combined with basic shapes to obtain the final characters by formulating and solving assignment problems for this purpose. The system was tested against the handwriting of 20 subjects, yielding overall sub-word and character recognition rates of 55.4% and 51.1%, respectively.

Haraty and Ghaddar [63] (2003) propose the use of two neural networks to classify previously segmented characters. Their method uses a skeleton representation and structural and quantitative features such as the number and density of black pixels and the numbers of endpoints, loops, corner points, and branch points. On 2,132 characters, the recognition rate is over 73%

Amin (2003) [64] presents an automatic technique to learn rules for isolated characters. Structural features including open curves in several directions are detected from the Freeman code representation of the skeleton of each character and the relationships are determined with Inductive Logic Programming (ILP). Test data consist of 40 samples of 120 different characters by different writers with 30 character samples used for training and 10 for testing for most experiments. A character recognition rate of 86.65% is obtained.

Alaei et al. (2010)[65] proposed a two-stage approach for isolated handwritten Persian character recognition. They extracted features based on modified chain code directional frequencies and employed an SVM for classification. They obtained 98.1% and 96.6% recognition accuracy with 8-class and 32-class problems, respectively.

Desai (2010)[66] presented a technique for Gujarati handwritten numeral recognition. the author used features abstracted from four different profiles of digits with a multilayered feed forward neural network, and achieved an approximate 82% recognition accuracy for Gujarati handwritten digit identification.

Sharma and Jhajj (2010) [67] extracted zoning features for handwritten Gurmukhi character recognition. They employed two classifiers, namely k-NN and SVM. They achieved a maximum recognition accuracy of about 72.5% and 72.0% with k-NN and SVM, respectively .

Kumar et al. (2011) [68] extracted intersection and open end points features for offline handwritten Gurmukhi character recognition. They used SVM for classification, taking

90% of the dataset as a training set and 10% of the dataset as a testing set. They achieved a maximum recognition accuracy of about 94.3%.

Khader Mohammada, Muna Ayyeshb, Aziz Qaroushc, Iyad Tumar [71] a novel segmentation approach for machine Arabic printed text with diacritics is proposed. The proposed method reduces computation, errors, gives a clear description for the sub-word and has advantages over using the skeleton approach in which the data and information of the character can be lost. Both of initial evaluation and testing of the proposed method have been developed using MATLAB and shows 98.7% promising results.

## 2.6 OCR in Holy-Quran

There are no published studies and research where OCR is used to read the texts of the Quran which written in Arabic and "Uthmaani" font. But according to one of the reports issued by the King Fahd Complex for the Printing of the Holy Quran A report entitled "The efforts of King Fahd Complex for the Printing of the Holy Quran use the modern technologies to serve the Holy Quran."

The report touched on a number of works, programs and technical tasks concerned with the compound developed and harnessed to serve the Holy Quran and its sciences such as check electronic files tool during preparations the search mentioned that the tool had prepared and programmed by the unit of research and development Computer in the complex, which is currently in use at the department concerned with the two sizes of the Quran Format, planning is under way to prepare additional copies for the

tool that mentioned; to serve the rest of the other sizes. There is no details on how to develop it and algorithms which used in the construction of the tool.

## 2.7 Summary

Table 2.1 shows all studies on Arabic Optical Character Recognition

#	Authors	Type	Feature Extraction and Classification Tech	Used Datasets	Reported Achievements
1	M. Sarfraz et al.[72] 2003	Printed	Moment Invariants for Feature Extraction and Neural Network,RBF Network for classification	Tested with many different sizes of Naskh font ,image is a gray scale of size 640x480. The input image is composed nearly of 200 characters.	73%
2	Haraty and Ghaddar [73] (2003)	Printed	two neural networks to classify previously segmented characters. Their method uses a skeleton representation and structural and quantitative features such as the number and density of black pixels and the numbers of endpoints, loops, corner points, and branch points. On 2,132 characters	On 2,132 characters, the recognition rate is over 80%.	80%

3	Amin[74] (2003)	Printed	<p>presents an automatic technique to learn rules for isolated characters. Structural features including open curves in several directions are detected from the Freeman code representation of the skeleton of each character and the relationships are determined with Inductive Logic Programming (ILP).</p>	<p>Test data consist of 40 samples of 120 different characters by different writers with 30 character samples used for training and 10 for testing for most experiments.</p>	86.65%
4	Ben Amor et al. [75] 2006	Printed	<p>Wavelets transformation &amp; Hough Transforms for feature extraction HMM Model for classification</p>	<p>Due to the absence in Arabic OCR of a data base, created our own corpus which is formed by 85000 samples in five different fonts among the most commonly used in Arabic writing which are: Arabic transparent, Badr, Alhada, Diwani, Koufi.</p>	98.97%
5	Alaei et al. [67] (2010)	handwritten	<p>proposed a two-stage approach for isolated handwritten Persian character recognition. They extracted features based on modified chain code directional frequencies and employed an SVM for classification.</p>	<p>They obtained 98.1% and 96.6% recognition accuracy with 8-class and 32-class problems, respectively</p>	98.1% and 96.6%

6	Desai [77] (2010).	handwritten	used features abstracted from four different profiles of digits with a multilayered feed forward neural network	and achieved an approximate 82% recognition accuracy for Gujarati handwritten digit identification[5]	82%
7	Nehad H A Hammad[78] 2015	handwritten	divide and conquer using character's curve tracing and number of dots information. A robust connected component labeling algorithm were used to divide the characters into four groups. To overcome the problem of size and scaling the tangent direction is used as feature (Divide-and-conquer). Four neural networks (NN) were used to classify each groups.	The off-line Arabic isolated character Dataset is Sudan University For Science and Technology Arabic Recognition Group (SUST ARG) used in proposed methods.	88.11%
8	Ozturk[79] 2009	28 different machine-printed	Artificial Neural Network	28 different machine-printed	%95 classification accuracy for the characters in these fonts

## CHAPTER 3

---

### Auditing Holy-Quran

#### 3.1 Introduction

Proposed system can be broken down into the following stages: image acquisition, pre-processing, segmentation, feature extraction, classification and post-processing, compare result of OCR with soft reference of Quran. Figure 3.1 illustrates these stages. The implementation of segmentation stage depends on the system scale, isolated characters or cursive words.

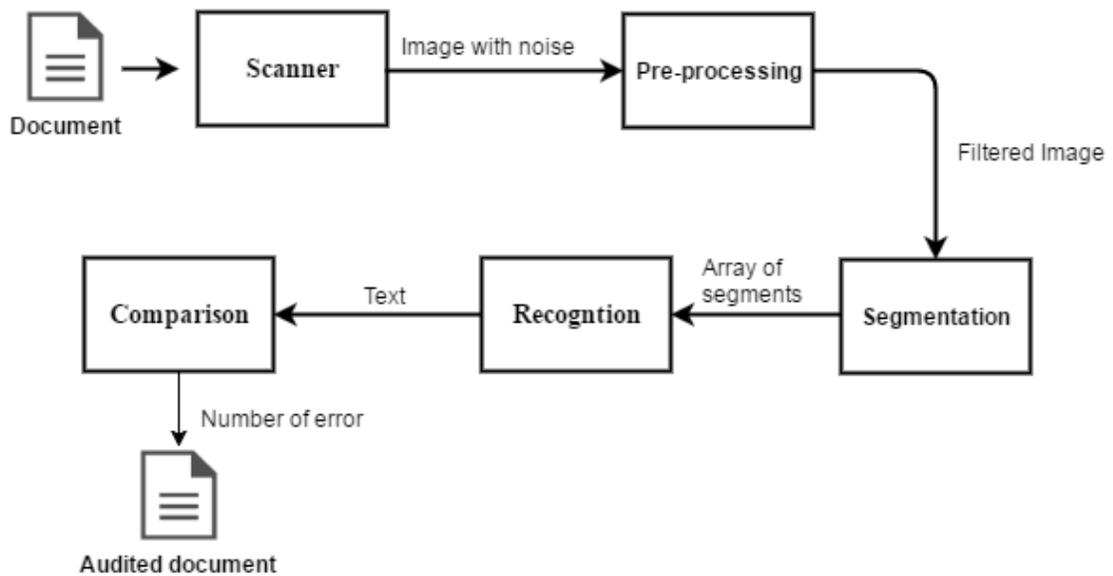


Figure 3.1 OCR System Stage.

## 3.2 Proposed system

In this thesis, we use the Optical Character Recognition approach for a special Arabic font to audit electronic files of the Quran.

Before sending pages of Quran to print must make sure there are no mistakes in whether character, adjust signs, or diacritical marks. In this thesis, we proposed a mechanism to check the pages of the Qur'an and determine if it contain mistakes and places of existence and automatic correction of these mistakes by comparing with certified reference, which it is an electronic copy of the Quran, audited by King Fahd Complex.

### 3.2.1 An Electronic Auditing Tool of the Holy Quran (EATHQ) Stage:

Scanning and analysis of the components of the image, removing the border frame from the scanned page so that we are left with the inner frames, which contain the names of the "Sura". Figure 3.2 show remove border.

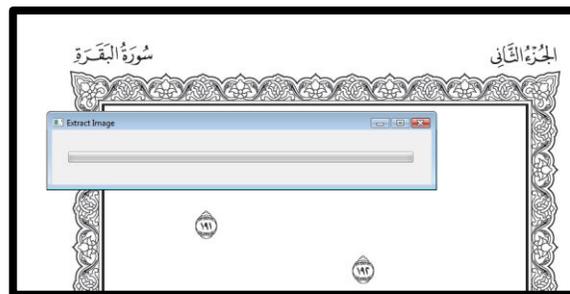


Figure 3.2 Remove border

2. Compare and note the differences between the file that sent for printing or publication and the original version, which does not contain any errors and certified.

In case of differences, the areas of these differences are marked with red squares as shown in Figure 3.3.

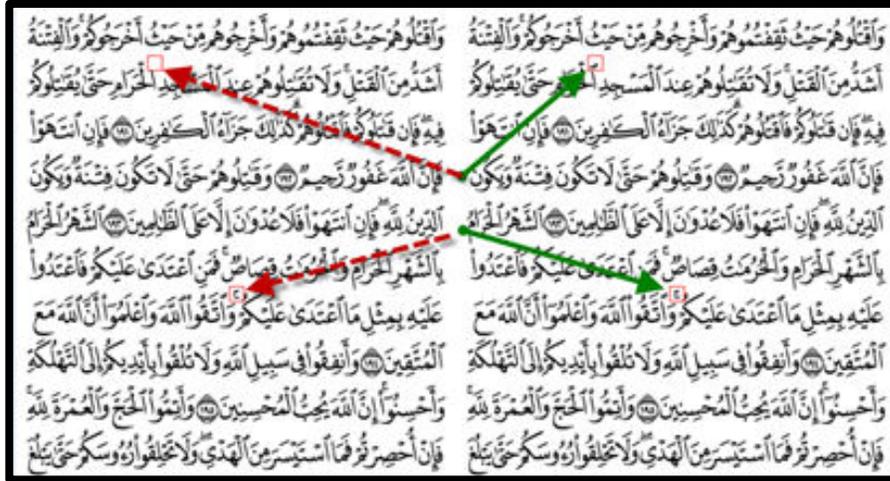


Figure 3.3 Output Image after Auditing Holy-Quran

In preprocessing step, we analyze the components of the image, stripping page of the foreign frames, the inner frames wall that contains the names, if any. Then split the image to line and identify termination of each ‘‘Aya’’ After that split to word and Identify character, Adjust signs, Diacritical marks. Then classification this character and compare word to reference if have error compare character to reference and determines where the error.

### 3.3 Image Acquisition

The first step in any OCR system is to capture text data and transform it into a digital form. The recognition systems differ in how they acquire their inputs. There are two different systems: on–line and off–line.

The Auditing Electronic Files of Qur'an is Off-line systems, where we used holy-Quran pages written in Arabic language and "Uthmaani" font as shown in figure 3.4.

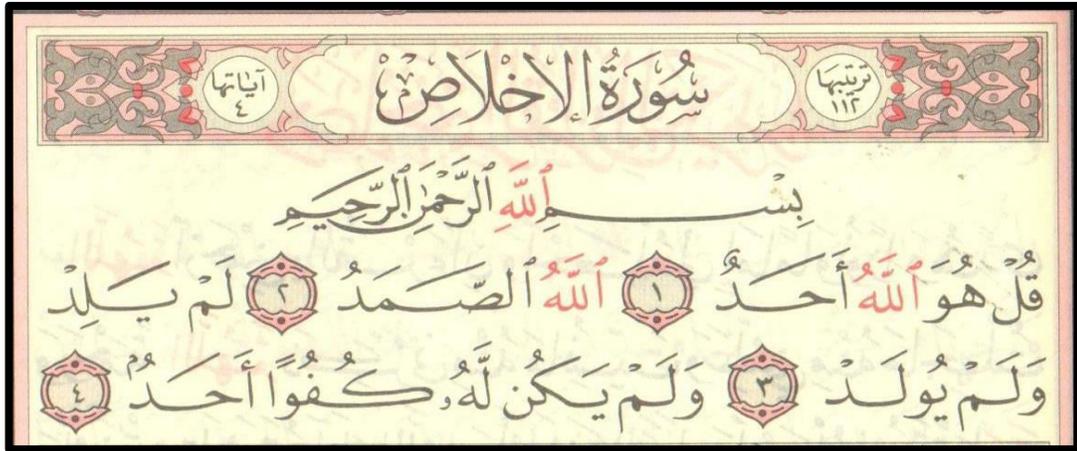


Figure 3.4 Image Acquisition

### 3.4 Pre-processing

A digital image represented as a two-dimensional matrix in the computer system. It is a function of 2 variables,  $f(x, y)$  where  $x$  and  $y$  indicate the row and the column in which the pixel lies, respectively. The value of the function  $f(x, y)$  represents the gray level associated with the pixel  $(x, y)$  and it varies from 0 (black) to 255 (white) and the midway is a mixture of white and black [80].

Thresholding is the mapping of pixels' values of a two-dimensional image into one of 2 values (usually 0 or 1) using a specified threshold level. Various methods used to calculate the threshold level. Thresholding can be global or local. Global thresholding calculates the threshold level from the whole image [80].

However, global thresholding methods can fail when the background illumination is un-even. A common practice is to pre-process the image to compensate for the illumination problems and then apply a global threshold to the pre-processed image, thresholding is an essential step before proceeding to the segmentation stage [81].

Pre-processing is an important stage in which many operations applied on the image in order to decrease the noise as much as possible and increase the accuracy and clarity of it. So we can obtain more accurate result from the segmentation.

Pre-processing consists of a set of operation as shown in the figure 3.5.

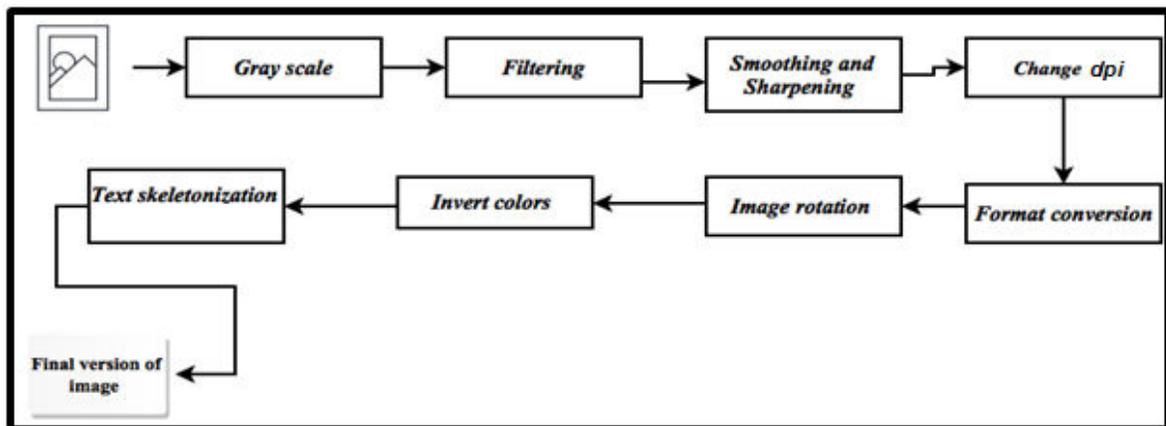


Figure 3.5 Pre-processing stage

### 3.4.1 Gray scale:

A gray scale operation is applied on the image, by this operation the colored image is converted to gray scale. In photography and computing, a grayscale or grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information.

`I = rgb2gray(RGB)`; This function converts the true color image RGB to the gray scale intensity image I. `rgb2gray` converts RGB images to gray scale by eliminating the hue and saturation information while retaining the luminance.

`M=im2bw (I)` this function converts the grey scale image I to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). Specify level in the range (0, 1). This range is relative to the signal levels possible for the image's class. Therefore, a level value of 0.5 is midway between black and white, regardless of class [82]. Gray scale operation is very important, because in the next parts of the project depends on the operation of binarization, which is done by converting the black pixel to 1 and the white pixel to 0, so this series of zeros and ones is used to either segment or recognize the character. Figure 3.6 show gray scale operations

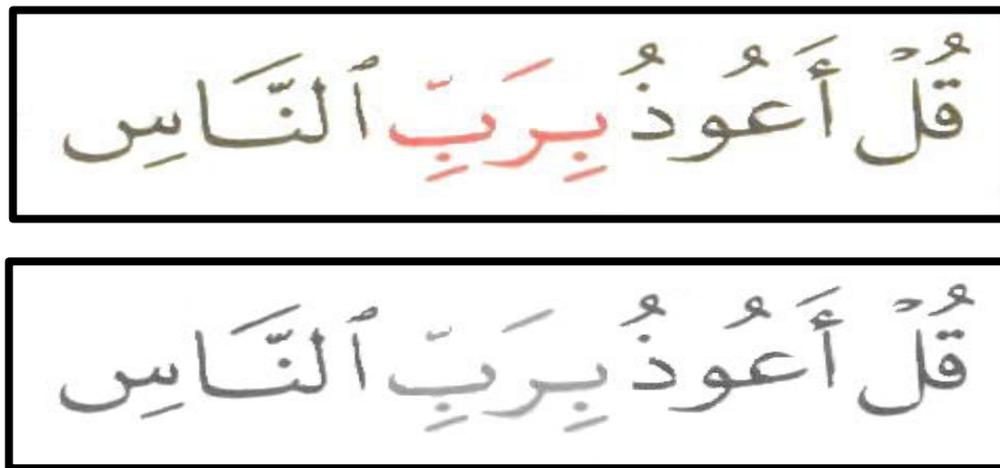


Figure 3.6 Gray scale operations

### 3.4.2 Image Filtering:

Filtering is an operation in signal processing which can be accomplished by a hardware device or a software component that removes from a signal some unwanted features (e.g. black dots from scanner) with the aim of enhancing the quality of the signal and prepare it for efficient use.

To achieve that, we used Matlab components, Matlab has very powerful and efficient components for Image Processing applications.

The filtering operation is made by `B =imfilter(A,h)` function, The `imfilter ()` filters the multidimensional array A with the multidimensional filter h. The result B has the same size and class as A. "imfilter" computes each element of the output B, using double-precision floating point. If A is an integer or logical array, "imfilter" truncates output elements that exceed the range of the given type, and rounds fractional values [83].

The result we get from this operation reduce noise, but not as we want or expected, because the output image from this operation is so blurred and the pixels isn't clear enough, so we have to use the next operation for better result . As shown in figure 3.7

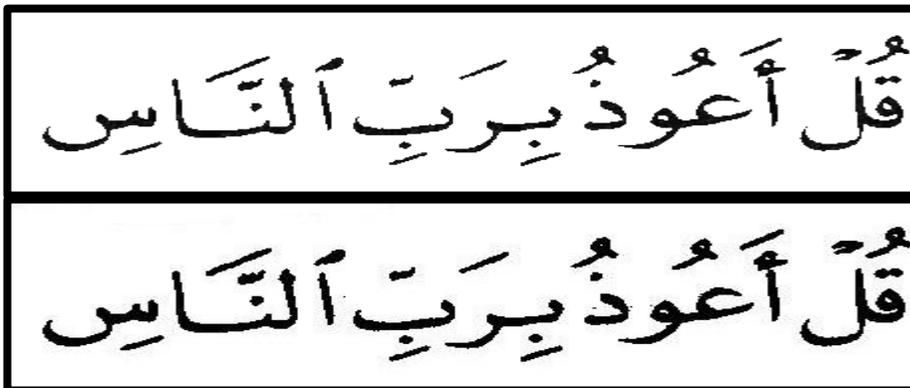


Figure 3.7 Filtered image

### 3.4.3 Sharpening:

Sharpening filters used in order to highlight fine details within an image. They based on first and second order derivatives. Since in image processing, we deal with discrete quantities, the definitions for the discrete first and second derivatives should be used. As shown in figure 3.8

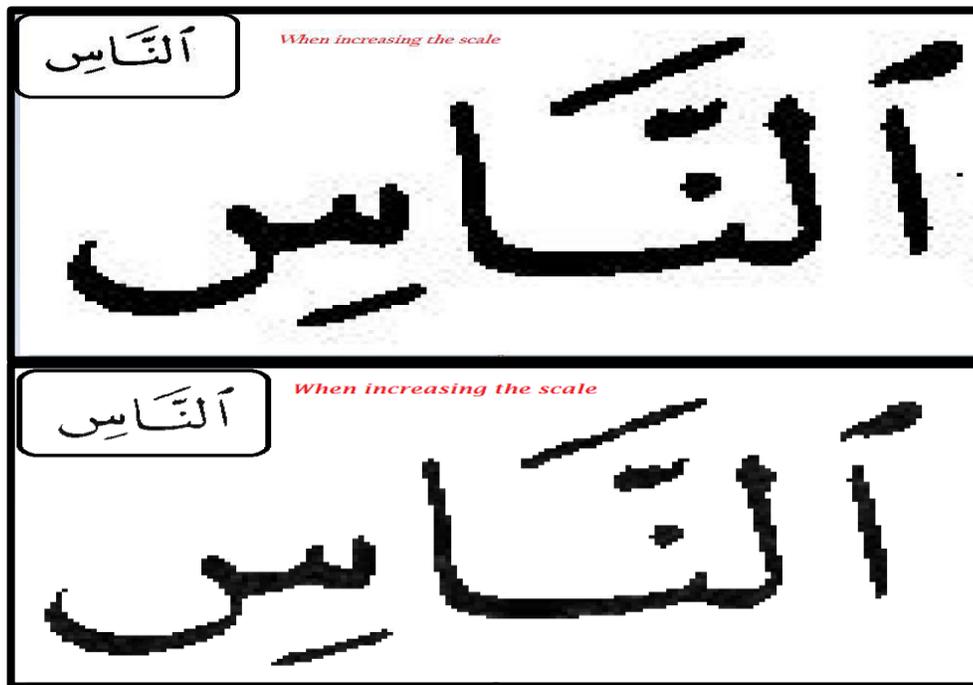


Figure 3.8 Image before and after Sharpening

### 3.4.4 Image Rotation:

A rotation is a circular movement of an object around a center (or point) of rotation.

One of the main problems we faced that the scanned image we got from scanner , wasn't at the right-angled as required , so we create a code that calculate the angle of deviation from the normal range (X - axis), then we use the output angel to rotate the image to the opposite direction, using `img.rotate(angle)`;so , we can get the image in

fixed 90 degree, because when the image is slant we can't chopping lines. As shown in figure 3.9

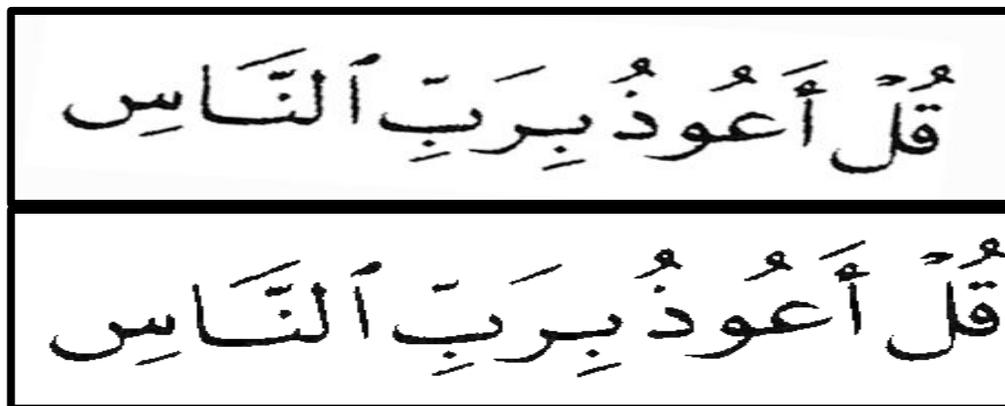


Figure 3.9 Image before and after Rotation

#### 3.4.5 Text Skeletonization:

In shape analysis, skeleton (or topological skeleton) of a shape is a thin version of that shape that is equidistant to its boundaries. The skeleton usually emphasizes geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width. Together with the distance of its points to the shape boundary, the skeleton can also serve as a representation of the shape (they contain all the information necessary to reconstruct the shape) [88].

We get the skeletons of the text by Matlab function, so we can split the text to specific words and letters in segmentation stage easily. This algorithm used for thinning binary image (so the program reads a binary image into the workspace and then performs several morphological operations on it). Binary image by definition consists of only black and white pixels. Figure 3.10 show skeletonization results.



Figure 3.10 Image before and after Skeletonization

Algorithm 1 show Pre-processing steps

---

**Algorithm 1** Pre-processing

---

**Procedure** Pre-processing (Image)

```
RGB = imread('22.png')
imshow(RGB)
I = rgb2gray(RGB)
I=imcomplement(I)
tasveer=I
figure,imshow(imcomplement(tasveer))
h = ones (2, 2)/1
rgb2 = imfilter(rgb,h)
imwrite(rgb2, 'filter.png', 'png')
```

**End procedure**

---

### 3.5 Segmentation

Segmentation can be defined as the process of dividing a word into characters. It is one of the hardest, crucial, and time-consuming phases. It represents the main challenge in any OCR system, even more than the recognition process itself. It is considered as the main source of recognition errors [55].

A poor segmentation process produces misrecognition or rejection. Arabic script may horizontally overlap as shown in Figure 3.11, so a specific segmentation method is needed.

An Arabic character has two to four different forms, which depend on its position in the word/subword. Any Arabic word segmentation algorithm will be concerned with three forms: the beginning form, the end form, and the isolated form. A particular characteristic in the appearance of the isolated and end forms of Arabic characters. Namely, the last strokes of these two forms of characters are either horizontal straight lines or upward curves. An Arabic character can possibly overlap with another character within a word if the last stroke of its isolated or end form is a horizontal straight line [36].

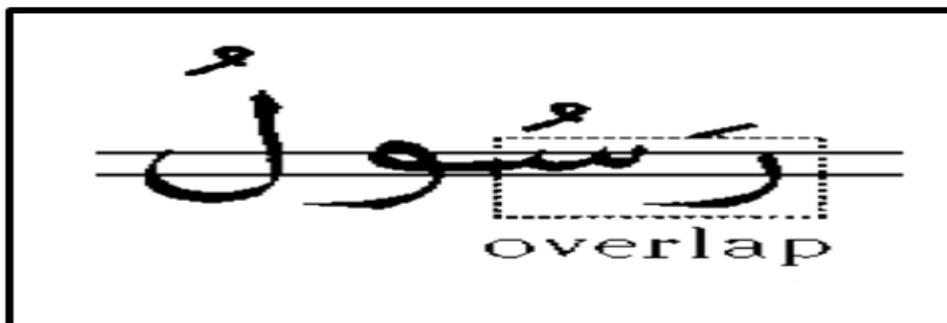


Fig. 3.11 overlaps in Arabic word [36].

In General segmentation is consists of a set of several steps as outlined in figure 3.12. In this thesis, we use these steps but we created or modified an algorithm for each step because Uthmaani font is different from other fonts.

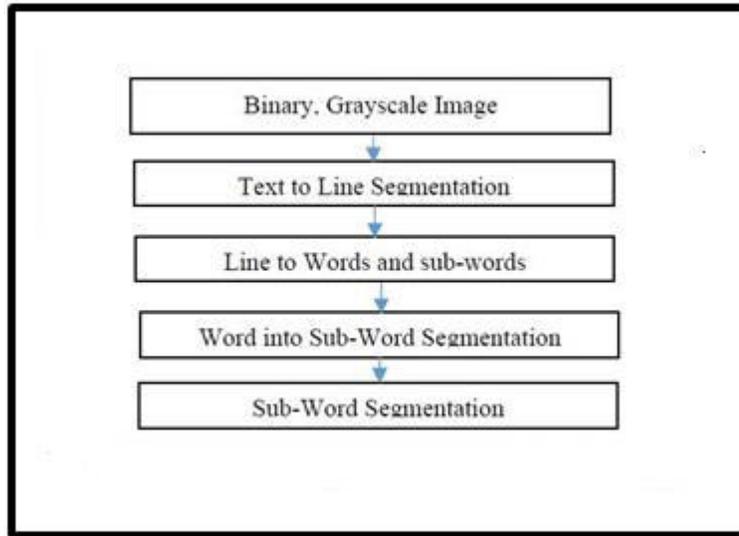


Fig. 3.12 Segmentation steps flow chart.

### 3.5.1 Text to line Segmentation:

This step divides the image into separated lines. This achieved by using horizontal projection algorithm. That is, following the rows of the array when we find the first non-empty row (the first row contains black pixels) this will be the first row of the line and when we find the first empty row, this will be the last row of the line. We repeat this process over the whole image to find all lines. Every line we find segment it into words or sub-words.

Horizontal Projection algorithm  $h(i)=\sum P(i,j) \dots\dots\dots(7)$

- Find the number of the black pixels in each row .
- Find the first row with black pixels(the start of the line)
- Find the first row with no pixels (the end of the line)
- Repeat this process to find all rows

### 3.5.1.1 Problem in segmentation of lines

In Text to line segmentation step, we modify horizontal projection algorithm, which develop by Safwa Taha, Yusra Babiker, and Mohamed Abbas [89] to avoid overlapping lines Problem in segmentation of lines for Uthmaani font in Holy-Quran. We solved this problem in step 1 in algorithm 2.

Figure 3.13 and 3.14 show the result of text to line segmentation.

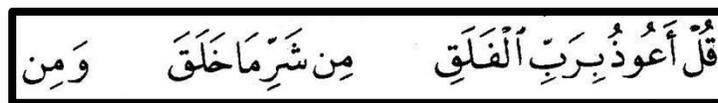


Figure 3.13 horizontal projections Line1

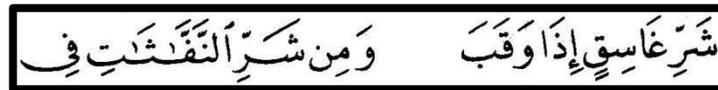


Figure 3.14 horizontal projections Line2

Algorithm 2: show modify horizontal projection algorithm to avoid overlapping lines.

---

**Algorithm 2** Text to line segmentation

---

**Procedure** horizontal projection (Image)

Step 0: **IF** the current row index is smaller than max rows index **then**

    Build up horizontal projection of this row.

**IF** the current row index value equals 0 **then**

            Store the corresponding row index in array 1.

**Else**

            Go to the next row.

            Go to step 0.

**Else**

            In array 1 search for more than 5 consecutive

indexes

**IF** find it **then**

                    Store those indexes in array 2

**End If**

**End If**

**End If**

Step 1: In Case of overlaps in **array 2**

    Find the row with minimum number of black pixels in the middle of lines (min).

    Take the first line from rows (0) to min.

        Delete black pixels from the end of the first line.

    Take the second line from min to the end of lines.

        Delete black pixels from the beginning of the second line.

**End procedure**

---

### 3.5.2 Line to words or sub-words

After finding the line, we segment it into words or sub-words by using vertical projection algorithm. By this method we tracing the array of the line, when we find the first column with a black pixels, this will be the first column of the word or the sub-word, then we keep tracing until we find the column with no black pixels, which it is the last column of line.

We repeat this process until we find the all words or sub-words in the line. The tracing is executed from right to left not from left to right because Arabic is written from right to left.

Vertical projection algorithm:

$$v(j) = \sum P(i,j) \dots\dots\dots(8)$$

- Find the number of black pixels in each column.
- Find the first column with black pixels from write.
- Find the first column with no black pixels.
- Repeat the process until find all words and sub-words in the line.

In Line to words or sub-words step, we create customer vertical projection algorithm to suit Uthmaani font in Holy-Quran. Algorithm 3 show customer vertical projection to find word and algorithm 4 find sub word.

---

**Algorithm 3** Words Segmentation

---

**Procedure** Vertical projection (Image)

Last Index=0

Text=false

Array start

Array end

**For** x=0 to Image width **do**

Count=0;

**For** y=0 to Image height **do****IF** pixel Array[y][x] =value **then**

++count

Text=true

**End If****End For****IF** count=0 and text=true **then**

start.add(lastIndex)

end.add(x)

Last Index=x

Text=false

**End If****IF** count=0 and text=false **then**

Last Index=x

**End If****End For**

Sub words (start, end)

**End procedure**

---

---

**Algorithm 4** Sub words Segmentation

---

**Procedure** Sub words (Array start, Array end)

**For** i=0 **to** start.size

        Last Index=start. get (i) **do**

            Text=false

**For** x=last Index **to** end.get(i) **do**

            Count=0

**For** y= base-10 **to** base+10

**IF** pixel Array [y][x] = value **then**

            ++count

            Text=true

**End If**

**End For**

**IF** count =0 **AND** text=true **then**

        StartPoint.add(lastIndex)

        SndPoint.add(x+5)

        Last Index=x+5

        Text=false

**End If**

**IF** count=0 **AND** text=false **then**

        Last Index=x

**End If**

**IF** text=true **AND** x=end.get(i)-2 **then**

        StartPoint.add(lastIndex)

        EndPoint.add(x+5)

        Last Index=x+5

        Text=false

**End If**

**End For**

**End procedure**

---

Figure 3.15 shows the segmentation results of this stage.

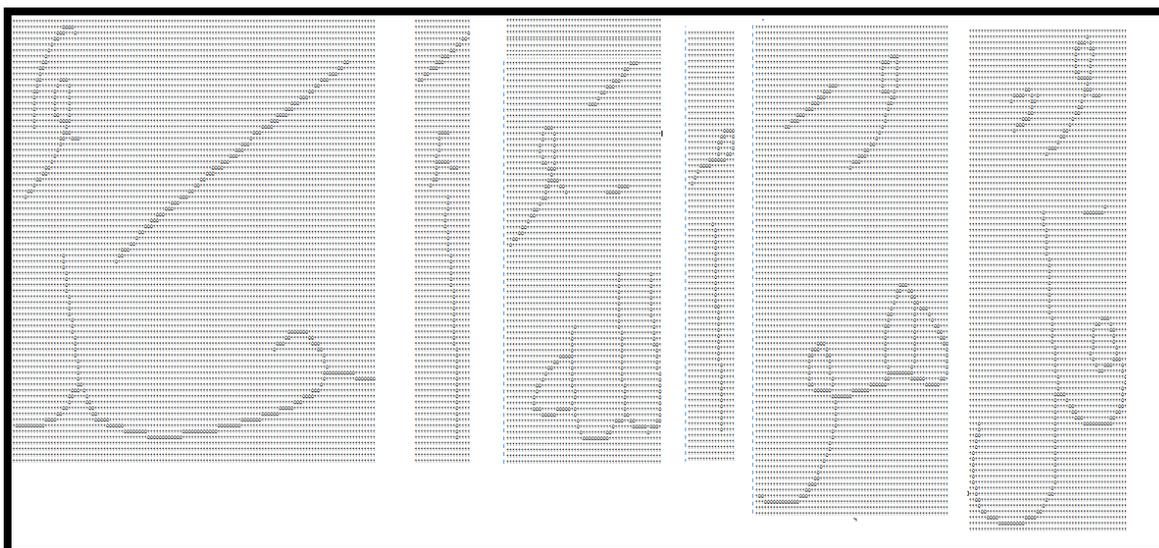


Figure 3.15 Vertical segmentation sub-word results

### 3.5.3 Segment each word or sub-word into characters.

Characters segmentation is the third level of the segmentation stage its algorithm is based on an important feature of cursive Arabic text (junction line).

Junction line is a small horizontal line that links between two consecutive characters in one sub-word. Fig 3.16 shows junction lines [89].

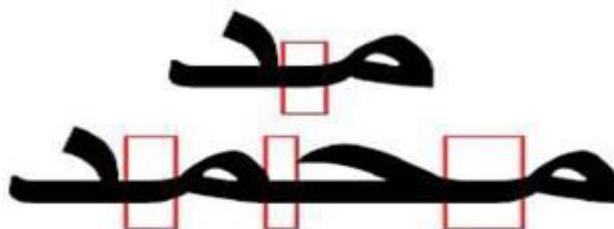


Fig. 3.16 Junction lines Example.

Since each character in the word except the first and last character placed between two junction lines, the proposed algorithm detects each junction line to extract every single character.

The extracted characters are then passed to the recognition stage which consists of two phases feature extraction and classification.

In chapter 2 of this thesis, the most common approach of OCR shown, and when studied, found that there are common problems that can be classified into two categories : Over-segmentation and Under-segmentation.

**Over-segmentation** happened to seen (س) sheen (ش), “Sad” (ص) and ”Dahd” (ض).

**Under-segmentation** mostly happened to “Alef” in the case when it is connected with other character. In our case , it happened also to “Ha” (ه) and "A'yn" (ع) in the middle of the word.

The first step in the proposed segmentation algorithm in this thesis is to find baseline, which it is the line with the most number of black pixels from full Image (not a skeleton sub-word), the baseline in full image more accurate than skeleton image. Algorithm 5 shows how to find base line.

Then find the boundaries using base line, the boundaries of baseline –the above and under lines - in the original word (before thinning), then tracing the above line to find the points when the curvature changes above it, this point will be a potential segmentation point. Figures 3.17 show the boundary example and Algorithm 5 Find base line.

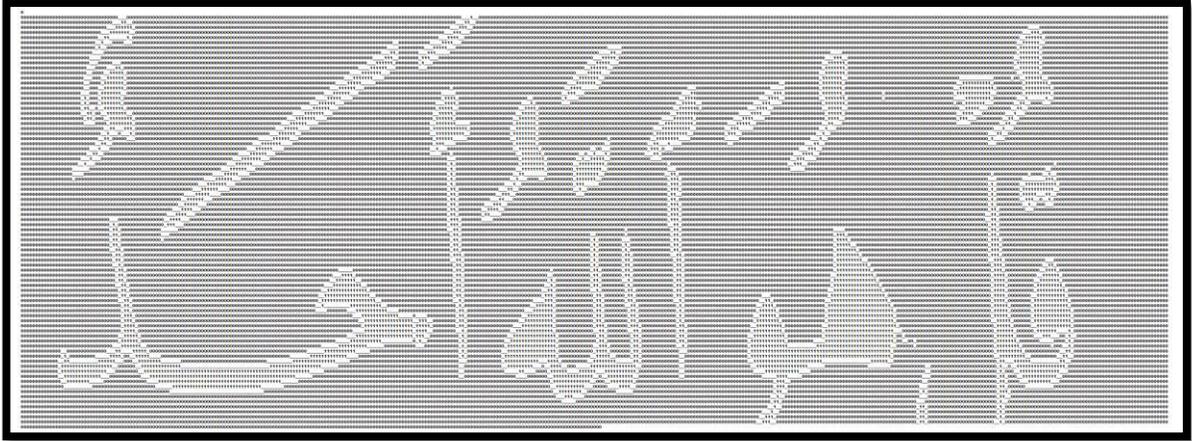


Figure 3.17 boundary example

---

**Algorithm 5** Find base line

---

**Procedure** base line ()

Count=0

Max =0

Index = -1

**For** i=0 to height **do**

Count =0

**For** j=0 to width **do**

**IF** Pixel Array[i][j]=value **then**

++ Count

**End If**

**End For**

**IF** count > Max **then**

Max=count

Index=i

**End If**

**End for**

Base line = index

**End procedure**

---

After find boundaries, the up-counter is need to determine suggested split point as figure 3.18 illustrates.

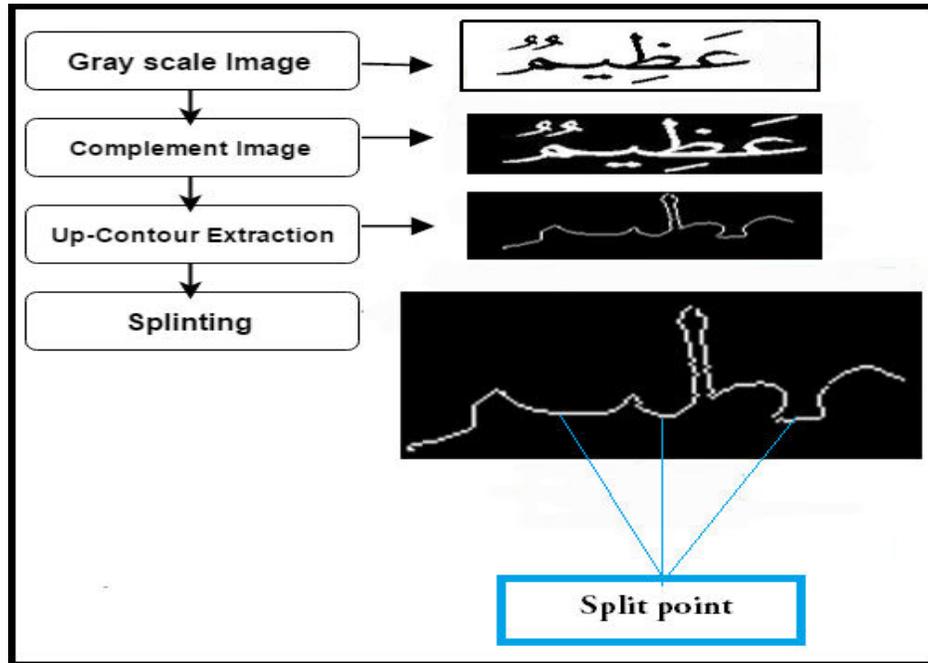


Fig. 3.18 up contour example.

**3.5.3.1 Up-Contour Extraction:** In The previous Stage (Segment each line into words or sub-words) for each sub-word array of start and end point are kept as show in pseudo code for Segment each line into words or sub-words. The start point locates in the first part of the first character in the sub-word contour, and the end point locates in the last part of the last character the sub-word contour.

To determine the first part of the first character and the last part of the last character, a threshold value first is determined according to the average length for the character, so the first part of the first character locates between the maximum column index of the sub-word contour and second column which is determined by subtracting the previous threshold value from the max column index.

In the same manner, the last part of the last character is determined but in this time, the region locates between the first column index in the contour and the second column assigned by adding the previous threshold to the first column index value. Figure 3.19 show up contour extractions



Figure 3.19 up contour extractions

**3.5.3.2 Splitting Areas extraction:** At this step, the up contour scanned to extract the splitting areas row by row then **get** the continuous regions in every scanned row and check the resulting continuous regions in each scanned row. If one of the pixels in boundary have '1' value and previous row has the value equals '0', this pixel is the suggested split area .Figure 3.20 shows suggest split area example.



Figure 3.20 suggest split area example

In Figure 3.20 the suggest split pixel given marker as '5' in red color and minimize them to choose one spilt pixel , '6 ' green pixel in figure 3.20 is spilt pixel that have a higher priority than '5' pixel because its form the high points. One of the highest points select to be spilt point, which gives the best character identify percentage.

### 3.5.3.3 The cases that the splitting areas should be ignored:

1- SAD (ص), DAD(ض), TAA(ط), and THAA (ض): SAD-DAD-IND algorithm is used, in this case the ratio between the height and the width of the transition area is determined, also the ratio between the height and the width of the of the main body of the character is determined, at the last the ration between the previous two ratios is calculated, if the ratio is between a certain threshold value **sad-dad-thres-values**, then the character satisfies the conditions and considered as SAD, DAD, TAA and THAA, refer to figure 3.21.

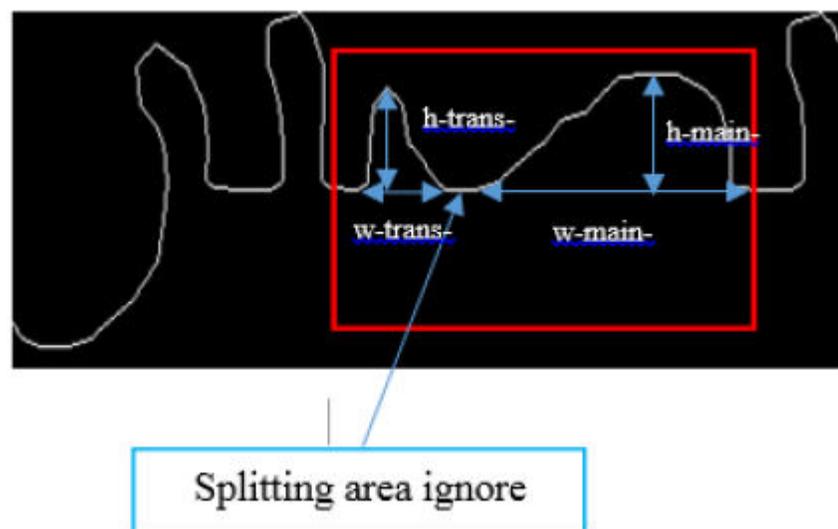


Figure 3.21 Splitting area ignore in case of SAD [71]

2- SEEN(س) SHEEN(ش) case :

As before the transition area should satisfy the ignore conditions. Then the area between the previous splitting area first reference point and the previous of the previous splitting area first reference point checked if it has diacritics points. The checking for the existence of dots in this region is done by determining the overlapping diacritics first, then the overlapped area is determined, after that, the ratio between the size of the overlapped area to the size of the diacritic (dot), if the calculated ratio is more than a threshold value, then the character in this case is sheen.

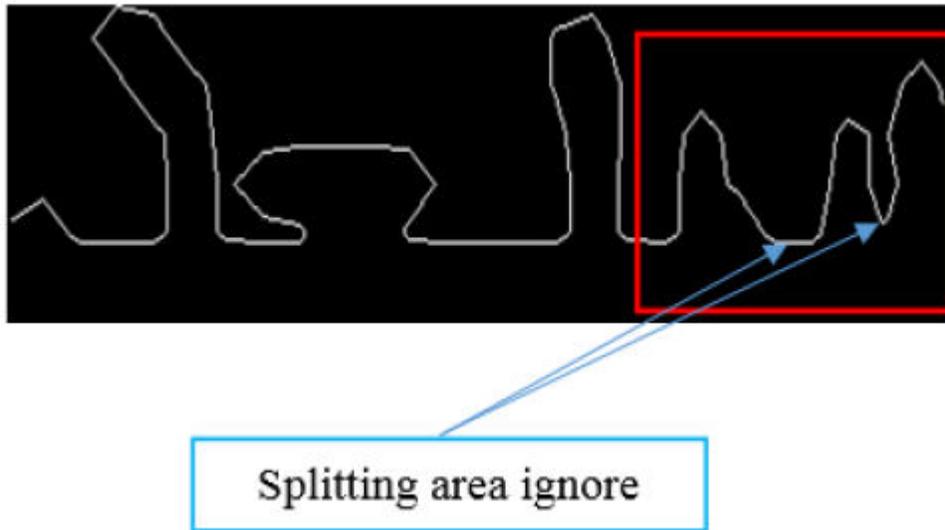


Figure 3.22 Splitting area ignore in case of SEEN [71]

For seen(س) case the number of the transition areas that stratifies the conditions of ignore are counted if they equals to three then the character seen and the previous two splitting areas should be ignored in this case, Figure 3.22 shows the splitting areas that should be ignored in case of sheen letter.

Figure 3.23 shows the segmentation each word or sub-word into characters.

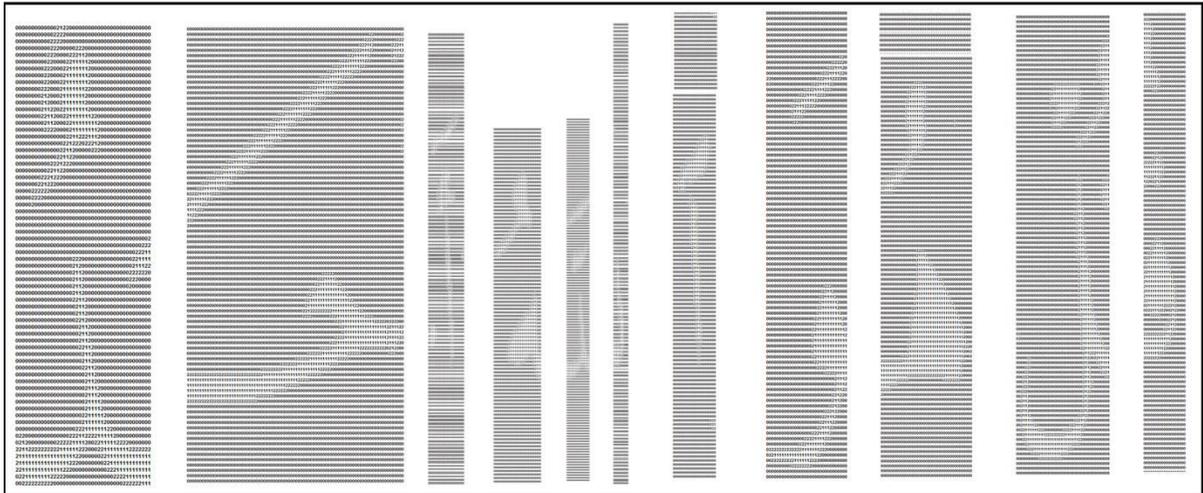


Figure 3.23 Segment each word or sub-word into characters

### 3.5.4 Extract diacritics from characters

The figure 3.23 show the result of Segment each word or sub-word into characters stage. As shown in the figure diacritics and adjustment sign connected with characters, using horizontal projection algorithm diacritics separated from the characters and keeps the index of each diacritics to pass it into recognition stage. As shown in Figure 3.24.

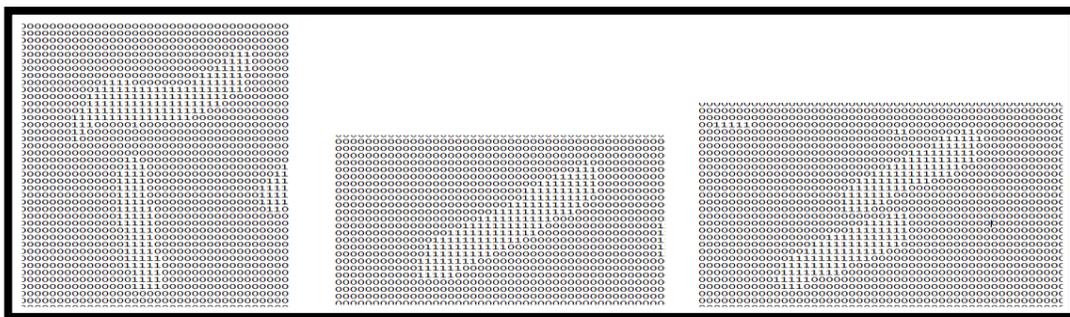


Figure 3.24 Extract diacritics

## 3.6 Recognition

Recognition is the final process, which is responsible of character determination, in this stage the segmented character checked for some features (feature extraction).

Feature extraction consists of the following features:

- 1- Height
- 2- Width
- 3- Summation of horizontal pixels
- 4- Summation of vertical pixels

In this stage, a new algorithm called "Matrix Summation" was developed. Its basic idea comes from the idea of both horizontal and vertical projection (Y- projection & X- projection) and the Freeman chain code algorithms, below is the definition of these algorithms:

### 3.6.1 X-Y projection:

Y- Projection (horizontal) is the process of extracting the character into rows, each row consists of number of pixels, and then sums these pixels to get the horizontal vector.

X- Projection (vertical V) is the column extraction of the character, each column contains number of pixels, then sum these pixels to perform the V vector.

### 3.6.2 Freeman chain code:

A technique used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. The direction of each segment coded by using numbering scheme such as the one shown in Figure 4.25 Chain codes based on this scheme are referred to as the freeman chain [89]. A unique vector relative to its boundary shape represents each character. An example is illustrated in Figure 4.26 which represents the “Kaf” letter ’ ك ’ vector:

[7,6,6,6,6,6,6,6,6,6,6,6,6,4,4,4,4,4,4,4,4,4,2,2,2,2,6,6,6,7,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2].

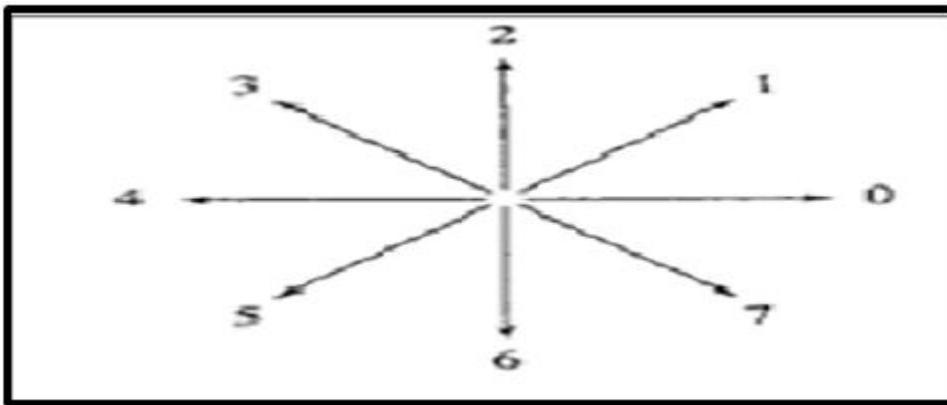


Figure 3.25 Chain codes based freeman chain [36]

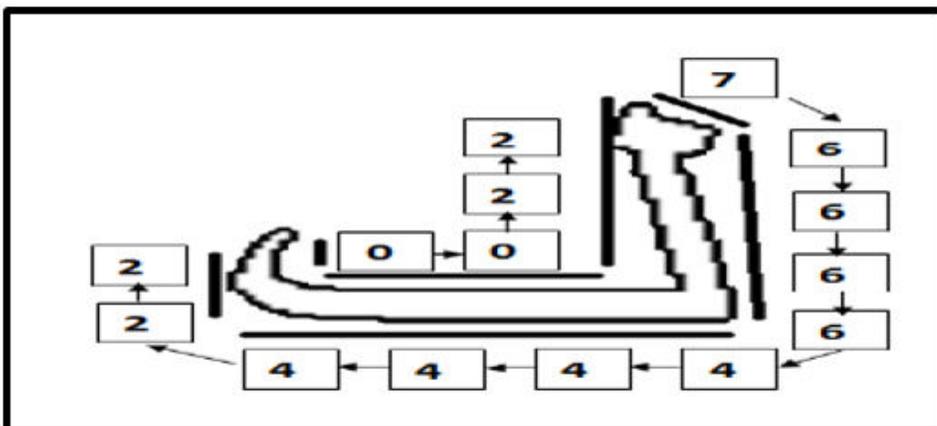


Figure 3.26 kaf letter ’ ك ’ vector [36]

The algorithm "Matrix Summation" based on the conversion the character to an array of white and black pixels ( so it is important to convert the image into white and black in the pre-processing level) , give each black pixel the value of (1) and others the value of (0) the result, as the Figure 3.27 illustrate the equivalent array of "dal" character (دال) And after summing the 1's horizontally and vertically the two vectors was obtained:

Hz : {7,10,11,6,4,5,5,4,4,5,6,17,29,29,29}

V: {3,3,4,5,6,7,8,9,9,7,6,7,7,7,6,7,6,6,6,7,7,6,5,4,4,4,3,3,1}



Figure 3.27 Dal 'د' in Binary

Then by counting the size of horizontal vector, we get the **height** of the character and by counting the size of vertical vector, we get the **width**.

### 3.6.3 Add Characters to database:

The database consists of the V & Horiz vectors for each character in its each form (isolate, initial, middle, end).

During the process of proving the quality of the algorithm and create the database Note that the same character can have different distributions of pixels in different pages and

even in different positions in the same page due to printing issue, so the algorithm, calculate the vectors on different pages and stored average of all vectors from these pages in the database as shown in Figure 3.28.

```
int[][] dbVectorHz = {{1, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1}, //ا
{2, 2, 4, 3, 4, 4, 3, 3, 4, 3, 4, 6, 8, 38, 37, 36, 32, 1, 0, 0, 0, 0, 2, 4, 5, 3, 2, 1}, //ب
{1, 4, 6, 6, 6, 5, 3, 1, 0, 0, 3, 2, 4, 5, 3, 3, 4, 3, 3, 4, 5, 7, 19, 37, 35, 33, 30, 1}, //ت
{2, 4, 4, 3, 1, 3, 5, 7, 7, 6, 3, 1, 0, 0, 2, 2, 4, 4, 3, 4, 3, 4, 3, 4, 5, 6, 9, 37, 36, 34, 30, 1}, //ث
{10, 22, 23, 18, 16, 9, 3, 3, 2, 1, 1, 2, 2, 2, 1, 1, 4, 5, 4, 2, 1, 1, 1, 2, 2, 3, 4, 5, 6, 8, 10, 21, 19, 14, 8, 4, 1}, //ج
{2, 17, 23, 23, 23, 14, 7, 5, 4, 3, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 3, 4, 5, 6, 9, 13, 22, 21, 15, 12, 1}, //ح
{2, 4, 5, 4, 1, 0, 0, 0, 0, 0, 0, 10, 22, 23, 23, 18, 12, 6, 3, 3, 4, 2, 1, 2, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 2, 2, 2, 3, 4, 5, 7, 10, 18, 13, 7, 1}, //خ
{2, 3, 3, 3, 3, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 15, 15, 15, 6, 1}, //د
{2, 3, 5, 3, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 4, 3, 3, 3, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 15, 15, 15, 14, 1}, //ذ
{1, 2, 3, 4, 4, 3, 3, 2, 2, 2, 1, 1, 1, 2, 2, 2, 3, 4, 4, 4, 5, 6, 5, 12, 11, 2, 1}, //ر
{3, 4, 4, 2, 1, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 3, 3, 2, 2, 2, 1, 1, 1, 1, 2, 3, 3, 4, 4, 4, 5, 5, 7, 13, 7, 1, 1}, //ز
{1, 1, 3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2}, //س
{1, 4, 5, 3, 2, 2, 2, 4, 4, 6, 4, 2, 2, 0, 0, 0, 0, 0, 2, 4, 5, 4, 4, 4, 4, 5, 7, 18, 17, 14, 4, 5, 3, 2, 3, 2, 5, 8, 11, 17, 21, 18, 15, 12, 2}, //ش
{3, 4, 16, 19, 20, 12, 11, 8, 8, 8, 14, 23, 30, 34, 35, 35, 1, 2, 3, 4, 5, 8, 9, 8, 13, 18, 13, 4}, //ص
{3, 3, 5, 4, 0, 0, 0, 0, 0, 0, 6, 8, 9, 12, 14, 6, 5, 6, 7, 10, 26, 30, 32, 32, 32, 1}, //ض
{2, 3, 5, 5, 5, 3, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 3, 13, 16, 19, 15, 10, 8, 7, 9, 13, 32, 35, 34}, //ط
{1, 2, 5, 5, 5, 4, 2, 2, 2, 2, 5, 5, 4, 2, 1, 1, 1, 2, 2, 2, 3, 4, 11, 14, 16, 18, 12, 9, 8, 8, 11, 35, 34, 34, 14}, //ظ
{8, 10, 9, 6, 3, 2, 2, 2, 3, 5, 9, 10, 14, 18, 14, 12, 7, 5, 3, 3, 2, 1, 2, 2, 2, 2, 2, 1, 1, 3, 3, 3, 3, 4, 4, 5, 7, 9, 22, 20, 18, 12, 3}, //ع
{1, 1, 4, 4, 4, 1, 0, 0, 0, 0, 0, 1, 8, 9, 12, 5, 3, 1, 2, 2, 3, 6, 9, 11, 15, 17, 15, 10, 7, 5, 3, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 3, 2, 3, 3, 12}, //غ
{1, 2, 5, 4, 3, 0, 0, 0, 0, 0, 1, 4, 5, 8, 10, 7, 5, 6, 6, 9, 10, 12, 11, 11, 12, 15, 39, 36, 31}, //ف
{3, 4, 7, 8, 7, 6, 5, 1, 1, 0, 0, 0, 2, 6, 7, 9, 7, 6, 6, 6, 9, 12, 12, 12, 13, 5, 4, 3, 3, 2, 3, 4, 5, 7, 8, 13, 21, 26, 24, 18, 11, 1}, //ق
{1, 3, 3, 5, 6, 5, 4, 4, 3, 4, 4, 3, 4, 3, 5, 5, 9, 10, 8, 4, 5, 6, 5, 6, 11, 3, 3, 3, 4, 5, 5, 9, 30, 30, 27, 24}, //ك
{1, 3, 4, 5, 4, 4, 3, 3, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 3, 3, 6, 16, 17, 17, 1, 1}, //ل
```

Figure 3.28 V & Horiz vectors sample

### 3.6.4 Prove "Matrix Summation" as character feature:

We calculate vertical vector, horizontal vector, height and width for all character in the last 10 page in Holy Quran, figure 3.27 show sample of character. Appendix A shows all character vectors.

When calculate vector for specific character in different positions in Quran Almost vertical vector, horizontal vector, height and width similar so in "Matrix Summation" algorithm we define error range to deal with the simple difference found in character vector in different positions. Table 3.1 show Seen (س) feature in 7 different positions.

Table 3.1 isolated Seen (س) feature in 7 different positions.

Test number	Feature	Experiment Result
1	vertical vector	{1, 1, 3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,4,2,3,3,4,4,6,7,6}
	height	28
	width	45
2	vertical vector	{1, 1, 1,3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,3,3,4,2,3,3,4,4,6,7,6}
	height	30
	width	46
3	vertical vector	{1, 1, 3, 5, 5,5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,4,2,3,3,4,4,6,7,6}
	height	28
	width	47
4	vertical vector	{1, 1, 3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,4,2,3,3,4,4,6,7,6}
	height	28
	width	45
5	vertical vector	{1, 1, 1,3, 3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2,2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,4,2,3,3,4,4,6,7,6}
	height	32
	width	45
6	vertical vector	{1, 1, 1,3,3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,4,2,3,3,4,4,6,7,6}
	height	31
	width	45
7	vertical vector	{1, 1, 3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2}
	horizontal vector	{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,3,4,3,2,3,4,3,3, 3,3,3,4,2,3,3,4,4,6,7,6}
	height	28
	width	45

### 3.6.5 Recognition Algorithm:

The algorithm 6 take the segmented character, which already represented as array, then calculates the vectors and compare them with vectors in the database.

---

**Algorithm 6** MATRIX SUMMATION

---

**Procedure** Recognition Letter (Segmented Character)

Calculate Hz & V vectors

Calculate width and height

Set Error range = 0

Search DB for the vectors

**IF** no result found **then**

    Error range ++

    Repeat Search DB for the vectors

**End If**

**IF** Error range reach the limit without finding the result **then**

    Return not found

**Else**

    Return the result character

**End If**

Rewrite the document using the recognized characters

**End procedure**

---

It is difficult to find vector identical to that in database due to Varsity of pixel distribution as mentioned, so the comparison process use a limited error range, starting with error range = 0 and search in DB, if no result found, increase the error range by 1 and repeat the search, because the minimum error will be with the same character, we still increase the error range until find the character .

To minimize number of entities on which we search, we distribute the characters on groups depending on the height and width, so if the height and width of Haá is 15 and 30 respectively, the algorithm will search only in the character group with that height and width, so the classes to search minimized and of course the search time is minimized too.

After experimentation, the result was perfect if the character was segmented perfectly, if there was a little error with segmentation, the result also good and small percentage of error will occur between similar characters such as: ت and ب , ث and ف , ي and ق

However, if the segmentation is incorrect, for example segment the same character into several parts or connect two or more characters with each other, the result will be wrong or not found.

### 3.7 Audit files

In this thesis, use optical character recognition for special Arabic font in auditing electronic files of the Qur'an. The goal of this project to build computer software tool auditing electronic files that contain pages of Qur'an and compared electronically with their assets.

The output of the previous stage image acquisition, pre-processing, segmentation, feature extraction, classification is file contain 'Sura' from Holy Qur'an in text format as shown in Figure 3.29

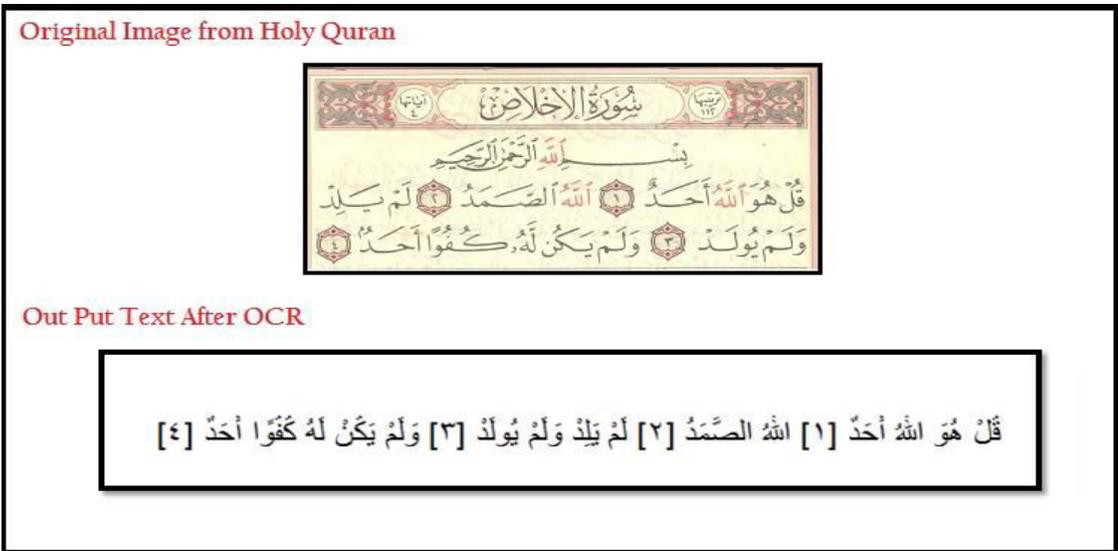


Figure 3.29 Sample System Result

To compare text output electronically with their assets we used "computer fonts" from King Fahd Group site. We use Microsoft Word file containing all pages of Holy-Quran "Uthmaani font – Hafs" as reference to be compare with the result of the OCR system as shown in figure3.30

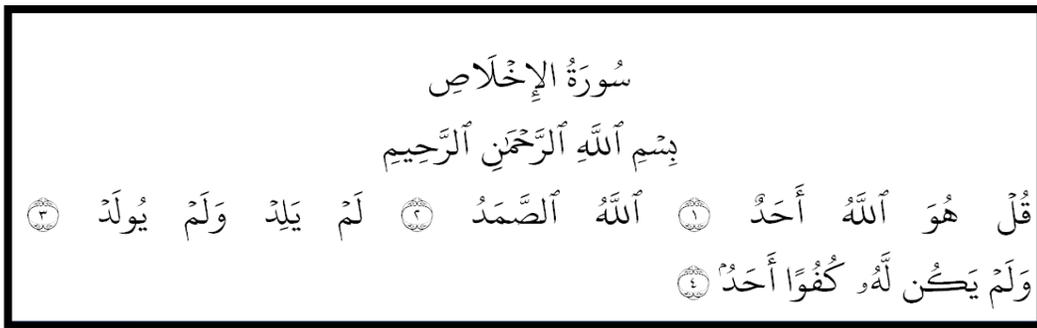


Figure 3.30 Sura from reference.

After comparing the two texts together if the program discovered any differences between them is developing a box window in red on areas of difference in the two files. As shown in figure 3.31.



Figure 3.31 Compare input with reference.

### 3.8 Auto correction

The output of the previous stage determines the location of the errors resulting from a comparison with the reference, to increase the effectiveness of the system and their applicability to the entire Quran autocorrect developed. The following steps describe the method.

- Detect error index.
- Determine error type (character, diacritics and adjustment sign).
- Based on the type of error, replace the error with correction from approved reference.

If the error in the characters, replaced the character in the matrix generated after segmented and identify stage, in the same way diacritics and adjustment sign replaced.

### 3.9 Summary

In this chapter OCR stage are detailed pre-processing, segmentation, recognition, and comparison with reference. Each part discussed with algorithms and charts in details.

In pre-processing stage, set of operations applied to image gray scale, filtered, sharpening, image rotation, image format conversion, increase the dpi to 300 and skeletonization.

In segmentation stage, segment to lines using horizontal projection, line to word and sub-word using vertical projection, sub-word to character and extract diacritics and adjustment sign from character.

In recognition stage, the segmented character checked for some features height, width, summation of horizontal pixels, summation of vertical pixels in this stage, a new algorithm called "Matrix Summation" created its basic idea comes from the idea of both horizontal and vertical projection (Y- projection & X- projection) and the Freeman chain code algorithms.

In audit stage , The output of the previous is file contain 'Sura' from Holy Qu'an in text format .To compared text output electronically with their assets we used ' computer Fonts Group Site ' from the site we use Microsoft Word file containing all pages of Holy-Quran " Uthmaani Font – Hafs" as reference to be compare with the result of the OCR . After comparing the two texts together if the program discovered any differences between them is developing a box window in red on areas of difference in the two files.

## CHAPTER 4

---

### Evaluation and Result

#### 4.1 Introduction

The goal of this thesis building a computer software tool auditing electronic files that contain pages of Qur'an and compared electronically with reference, before converting the files to the printing stage or publish the soft copy in order to reduce the time and manual effort during the audit of electronic files in the prepress stage. Reduce errors ratio that may occur during the printing phase and finally inventory errors that may appear, and documenting their location on each page.

In this chapter, evaluation and testing conducted on a sample dataset to determine the performance of the overall auditing system. Finally, the obtained result analyzed and discussed.

#### 4.2 System Evaluation:

There are no published studies and research where OCR is used to read the texts of the Quran which written in Arabic and "Uthmanic" font, But according to one of the reports issued by the King Fahd Complex for the Printing of the Holy Quran entitled "The efforts of King Fahd Complex for the Printing of the Holy Quran use the modern technologies to serve the Holy Quran".

The report addressed the number of works, programs and technical tasks concerned with the overall developed and harnessed efforts to serve the Holy Quran and its

sciences such as check electronic files tool during preparations the report mentioned that the tool had prepared and programmed by the unit of research and development computer in the complex, which is currently in use at the department concerned with the two sizes of the Quran Format, planning is under way to prepare additional copies for the tool that mentioned; to serve the rest of the other sizes. There are no details about algorithms that were adopted to develop the tool.

#### **4.3 Prove OCR System Database:**

In this thesis, we test OCR System on 23 sware from Holy Quran and build database for this sware, to prove that OCR system database it's like the reference we run system 10 time .Table 4.1 show OCR system results.

Test number	sware number	Number of error	OCR result
1	23	6	98.5%
2	23	2	99.6%
3	23	1	99.9%
4	23	0	100%
5	23	0	100%
6	23	0	100%
7	23	0	100%
8	23	0	100%
9	23	0	100%
10	23	0	100%

As shown in Table 4.1, we test OCR system on 23 sura from Holy Quran in the first result appear some error in OCR result duo to segmentation or recognition stage, after we correct the error we run system number of times to ensure that no error duo to OCR system appear in auditing system, and database for 23 sura as reference database.

#### 4.4 Auditing Result:

Testing was firstly conducted on Al-Ekhlās "سورة الإخلاص" and creates some mistakes then try the auditing system. The system detected these mistakes and drew red boxes at the mistakes' locations. Figure 4.1 illustrates, the first part, the result of the Pre- process Stage that extracted the "Sura" from page, removes frame, Detects "ayat" ends and removes it.

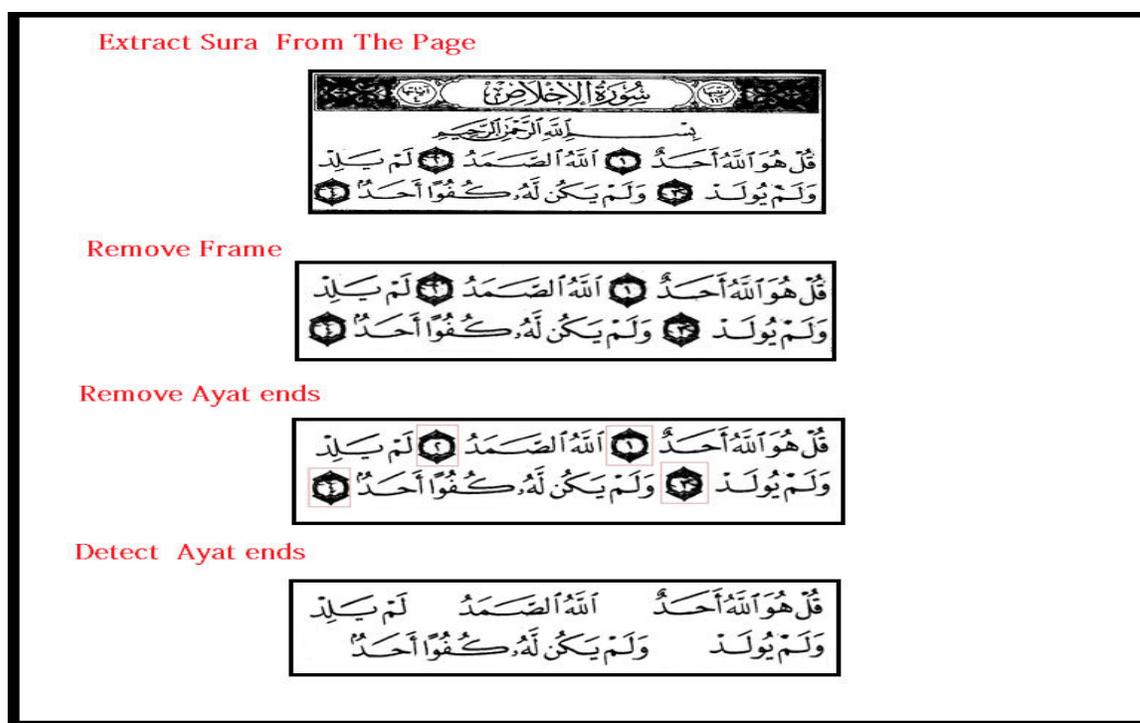


Figure 4.1 Pre- process results in Al-Ekhlās "سورة الإخلاص" part 1

Figure 4.2 shows the result of part 2, Pre- processes filtered image, inverts the image, and skeletonizes it.

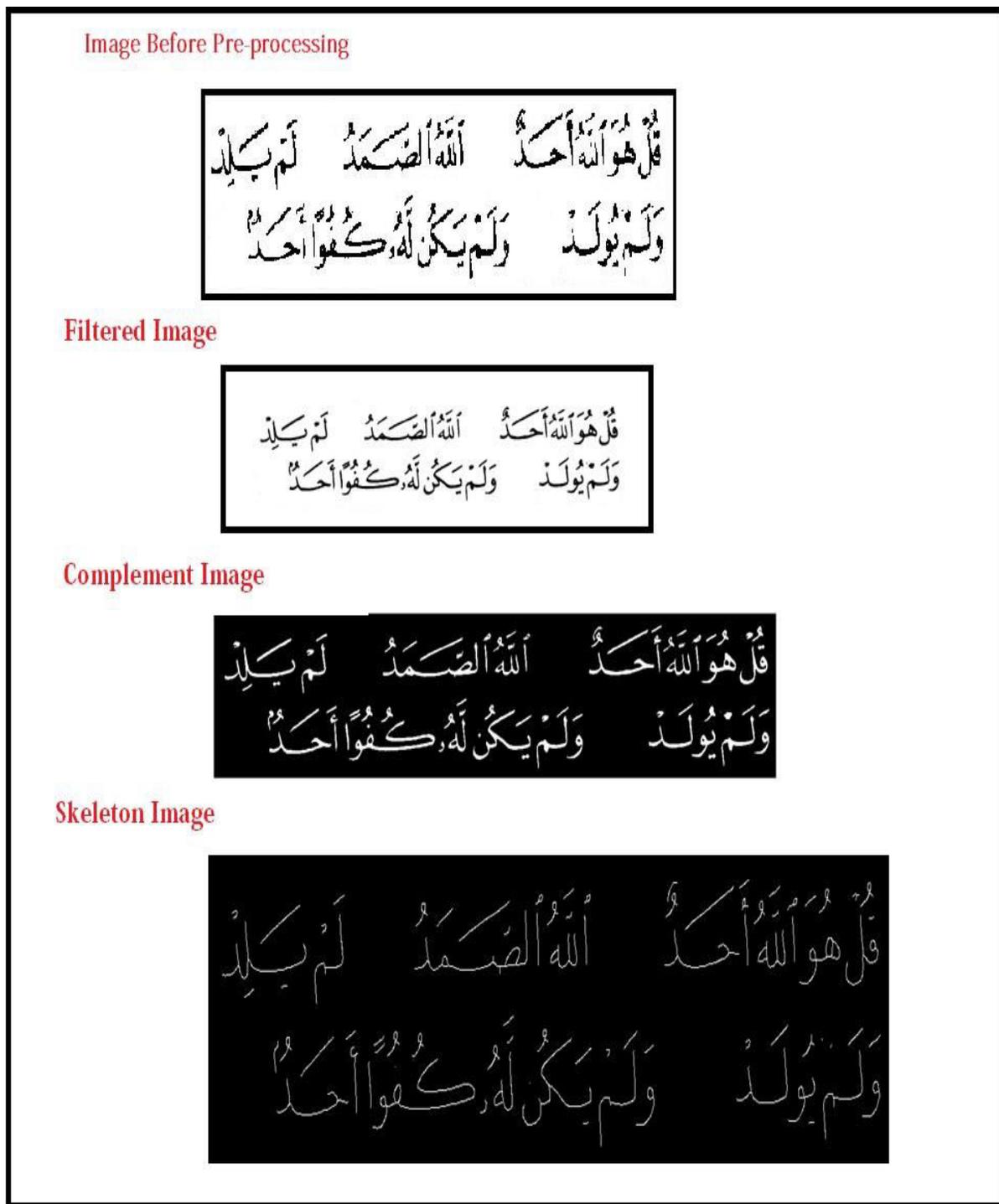


Figure 4.2 Pre- process results in Al-Ekhlav "سورة الإخلاص" part 2

Figure 4.3 demonstrates the result of the segmentation stage: splits line, splits word and sub-words, and splits characters.

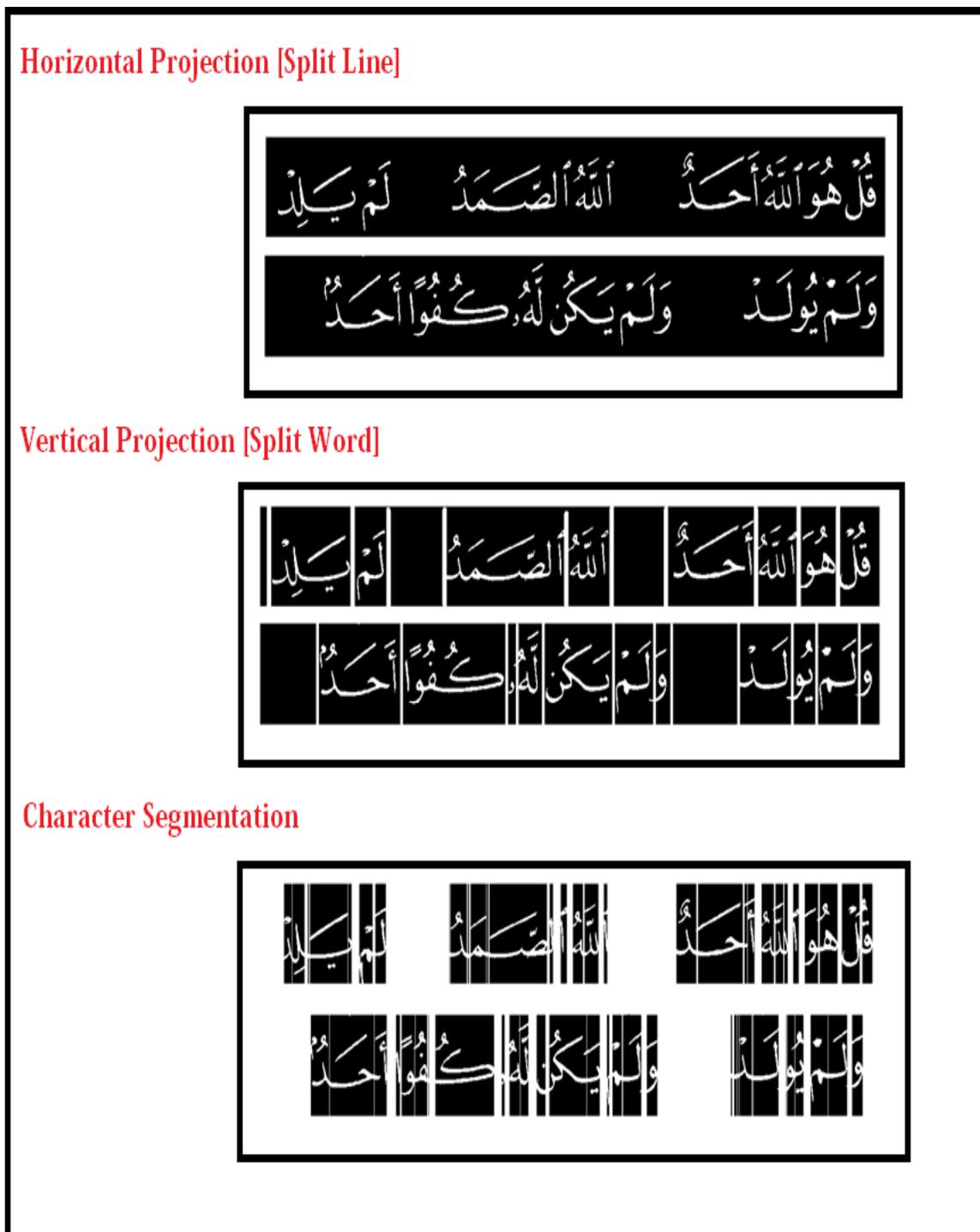


Figure 4.3 Segmentation results on Al-Ekhlās "سورة الإخلاص"

As shown in figure 4.4, the system detects three places for the presence of errors.

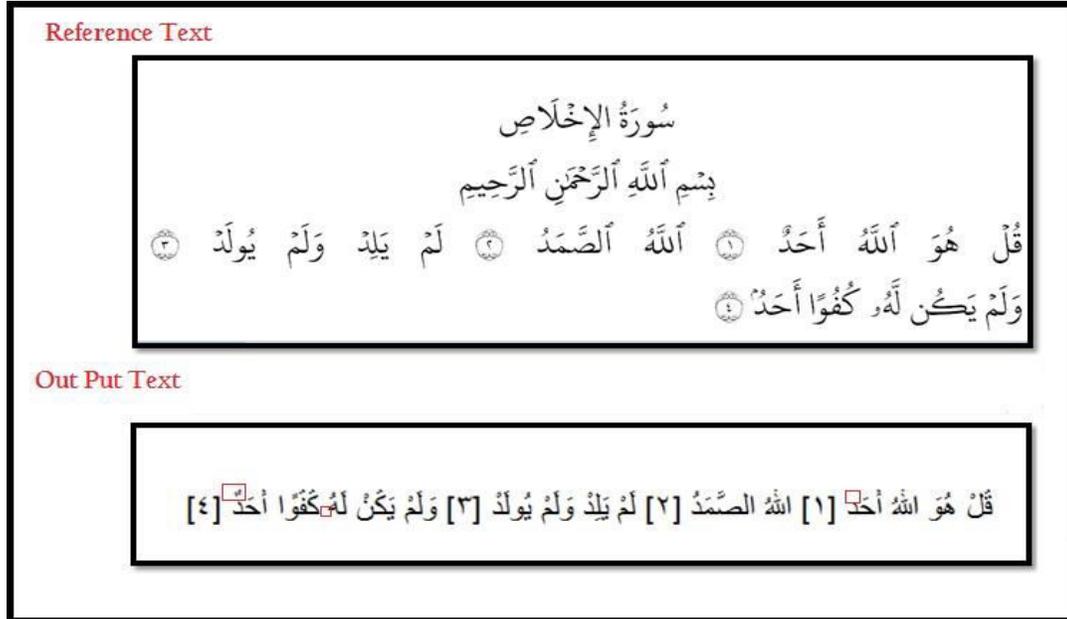


Figure 4.4 result of audit stage

In the first testing phase, the system is based on the last two pages in Holy Quran that contain 6 "Suar" "النصر", "المسد", "الكافرون", "الناس", "الفلق", "الإخلاص".

The results and the number of mistakes were highlighted and then modifies databases and algorithms after identifying the cause errors. As it is shown in the Figure 4.4, Al-Ekhlâs "سورة الإخلاص".

In the second phase, the system has been tested over the last five pages that covers "Swars" from Surat "الفارعة" to "Surat" "الناس". Figure 4.5 shows the results and the number of errors before updating database and after the development of the system.

Figure 4.5 shows the results and the number of errors before updating database for AL-HUMAZAH "سورة الهمزة".

Reference Text

سورة الهمزة

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَيَلِّ لِكُلِّ هَمْزَةٍ لَمْزَةً ①      الَّذِي جَمَعَ مَا لَا وَعَدَّدَهُ ②

يَحْسَبُ أَنَّ مَالَهُ أَخْلَدَهُ ③      كَلَّا لَيُنْبَذَنَّ فِي الْحُطَمَةِ ④

وَمَا أَدْرَاكَ مَا الْحُطَمَةُ ⑤      نَارُ اللَّهِ الْمُوقَدَةُ ⑥      الَّتِي تَطَّلِعُ

عَلَى الْأَفْعِدَةِ ⑦      إِنَّهَا عَلَيْهِمْ مُّوَصَّدَةٌ ⑧      فِي عَمَدٍ مُّمَدَّدَةٍ ⑨

Out put Text

سورة الهمزة

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَيَلِّ لِكُلِّ هَمْزَةٍ لَمْزَةً ①      الَّذِي جَمَعَ مَا لَا وَعَدَّدَهُ ②

يَحْسَبُ أَنَّ مَالَهُ أَخْلَدَهُ ③      كَلَّا لَيُنْبَذَنَّ فِي الْحُطَمَةِ ④

وَمَا أَدْرَاكَ مَا الْحُطَمَةُ ⑤      نَارُ اللَّهِ الْمُوقَدَةُ ⑥      الَّتِي تَطَّلِعُ

عَلَى الْأَفْعِدَةِ ⑦      إِنَّهَا عَلَيْهِمْ مُّوَصَّدَةٌ ⑧      فِي عَمَدٍ مُّمَدَّدَةٍ ⑨

Figure 4.5 numbers of errors before updating Database

To increase accuracy of system, modify the database and add adjustment sign as shown in Figure 4.6. Then again, audit the same image as shown in Figure 4.7 and have number of error zero

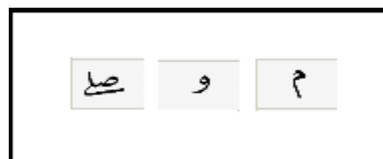


Figure 4.6 Modify database

## Input Text

وَيْلٌ لِّكُلِّ هُمَزَةٍ لُّمَزَةٍ الَّذِي جَمَعَ مَا لَا وَعَدَّدَهُ،  
يَحْسَبُ أَنَّ مَالَهُ أَخْلَدَهُ، كَلَّا لَيُنْبَذَنَّ فِي الْحُطَمَةِ  
وَمَا أَدْرَاكَ مَا الْحُطَمَةُ نَارُ اللَّهِ الْمَوْقُودَةُ الَّتِي تَطَّلِعُ  
عَلَى الْأَفْعِدَةِ إِنَّهَا عَلَيْهِمْ مُّوَصَّدَةٌ فِي عَمَدٍ مُّمَدَّدَةٍ

## Reference Text

سُورَةُ الْهُمَزَةِ  
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
وَيْلٌ لِّكُلِّ هُمَزَةٍ لُّمَزَةٍ ① الَّذِي جَمَعَ مَا لَا وَعَدَّدَهُ ②  
يَحْسَبُ أَنَّ مَالَهُ أَخْلَدَهُ ③ كَلَّا لَيُنْبَذَنَّ فِي الْحُطَمَةِ ④  
وَمَا أَدْرَاكَ مَا الْحُطَمَةُ ⑤ نَارُ اللَّهِ الْمَوْقُودَةُ ⑥ الَّتِي تَطَّلِعُ  
عَلَى الْأَفْعِدَةِ ⑦ إِنَّهَا عَلَيْهِمْ مُّوَصَّدَةٌ ⑧ فِي عَمَدٍ مُّمَدَّدَةٍ ⑨

Figure 4.7 numbers of error after updating Database

It has been continuing to increase the number of "السور" and pages that processed through the system and modifying databases to match the fence and entered with the rest of the adjustment and characters signs

In the final stage the system was examined on ten pages of the Holy Quran "حزب" that contains 23 "Sura" that starts from "سورة الشمس" to "سورة الناس".

In this approach, the system is scalable to the whole of the Holy Quran by creation a strong and comprehensive database for all characters, diacritics and adjustment signs, and auditing character segmentation to include all cases in the Quran.

When the system is examined on the “Suras” that start from the middle of the page and fragmented in two pages like surat "العاديات" borders Starting from the mark of the beginning of the current Sura up to the mark of the beginning of the following Sura, are as demonstrated in Figure 4.8

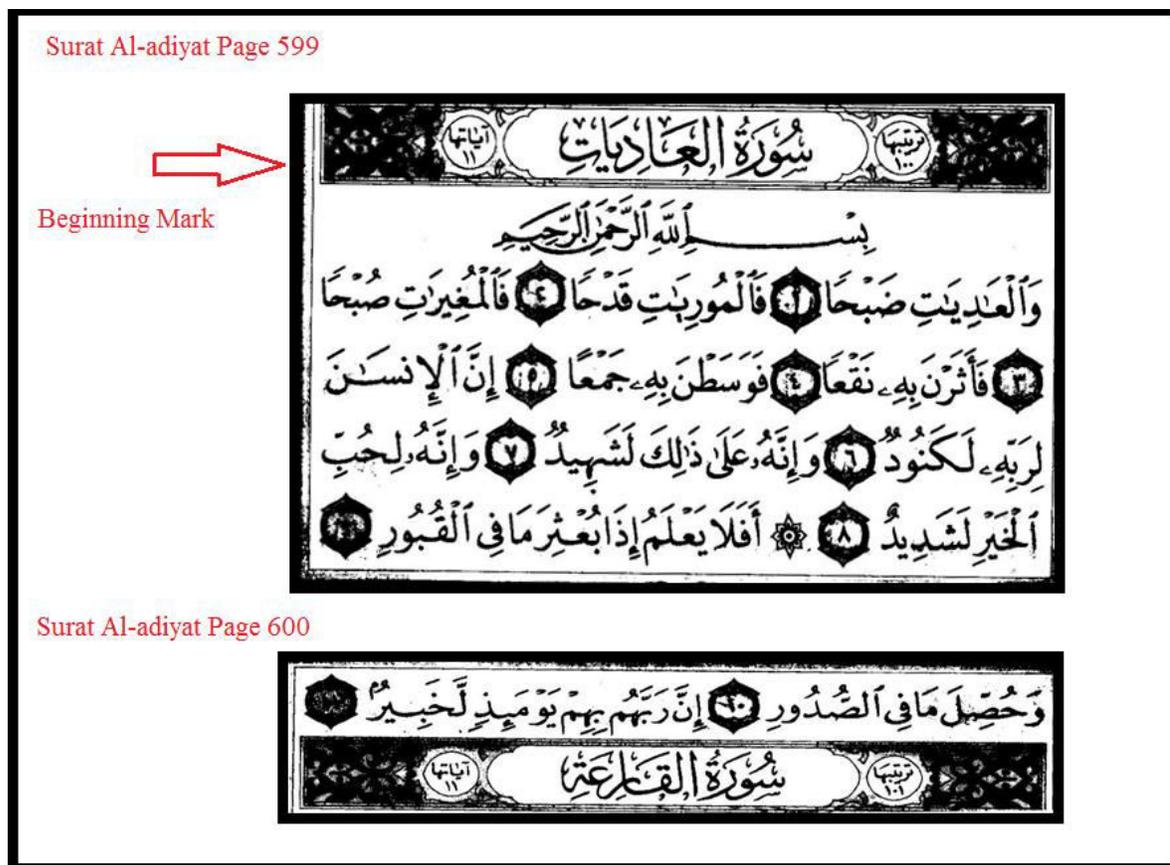


Figure 4.8 Sura in two page part1

Figure 4.9 show the result of Sura in two page after remove border

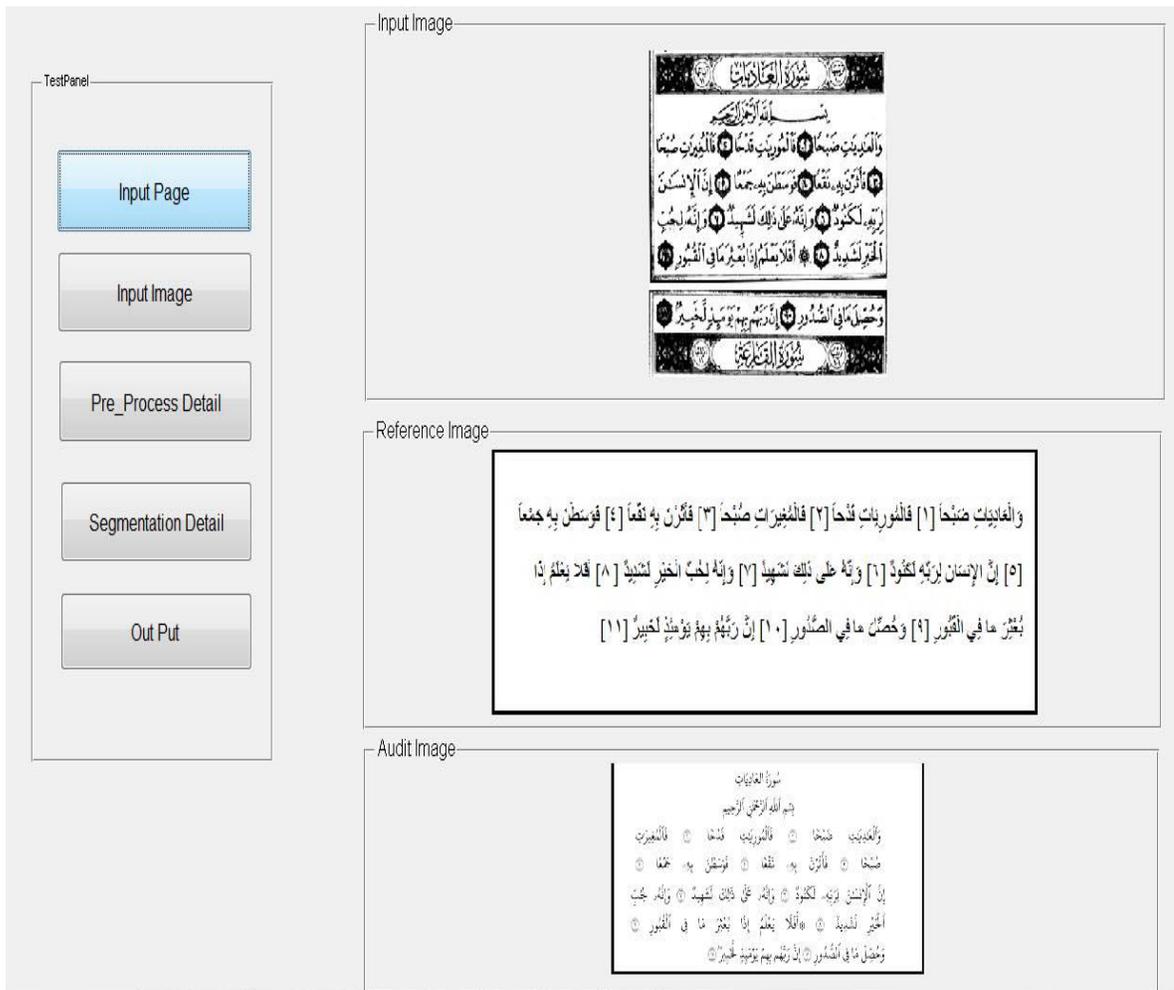


Figure 4.9 Sura in two-page part2

When the system is tested on a small-sized Quran, it has been identified correctly. This means that the font size is not a restriction as shown in Figure 4.10.

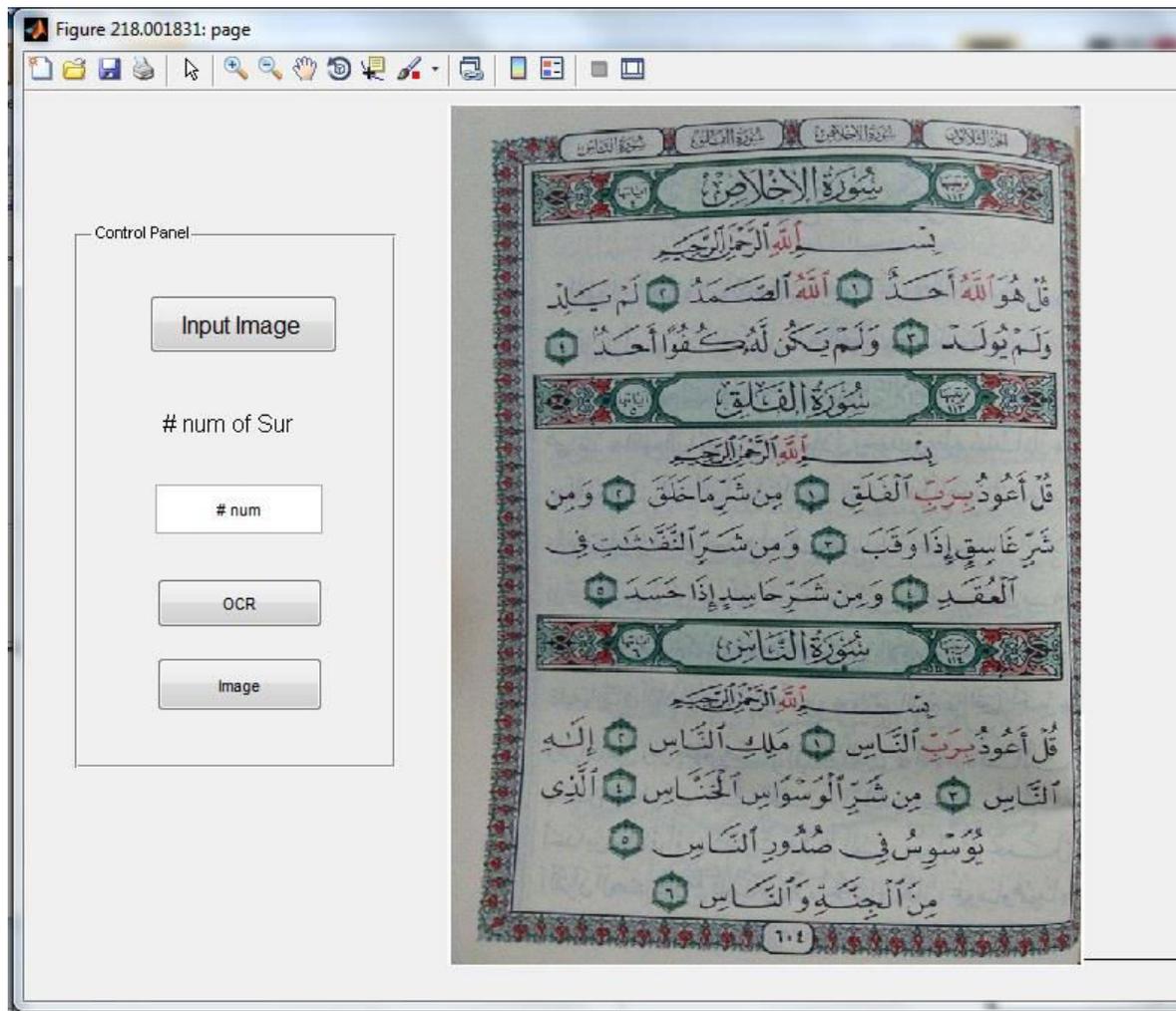


Figure 4.10 Small-sized Quran part1





This research will not deal with Time-response where it be proposed as a further work but will deal with number of errors in each page. However, we calculate the time required to processing Surat Al-Ekhlās completely, time is 22 seconds.

#### 4.6 Auto correction:

To increase the effectiveness of the system and their applicability to the entire Quran autocorrect developed after detect error index determine error type (character, diacritics and adjustment sign). In addition, based on the type of error, replace the error with correction from approved reference. If the error in the characters, replaced the character in the matrix generated after Segmented and identify stage, in the same way movements in the same way diacritics and adjustment sign replaced.

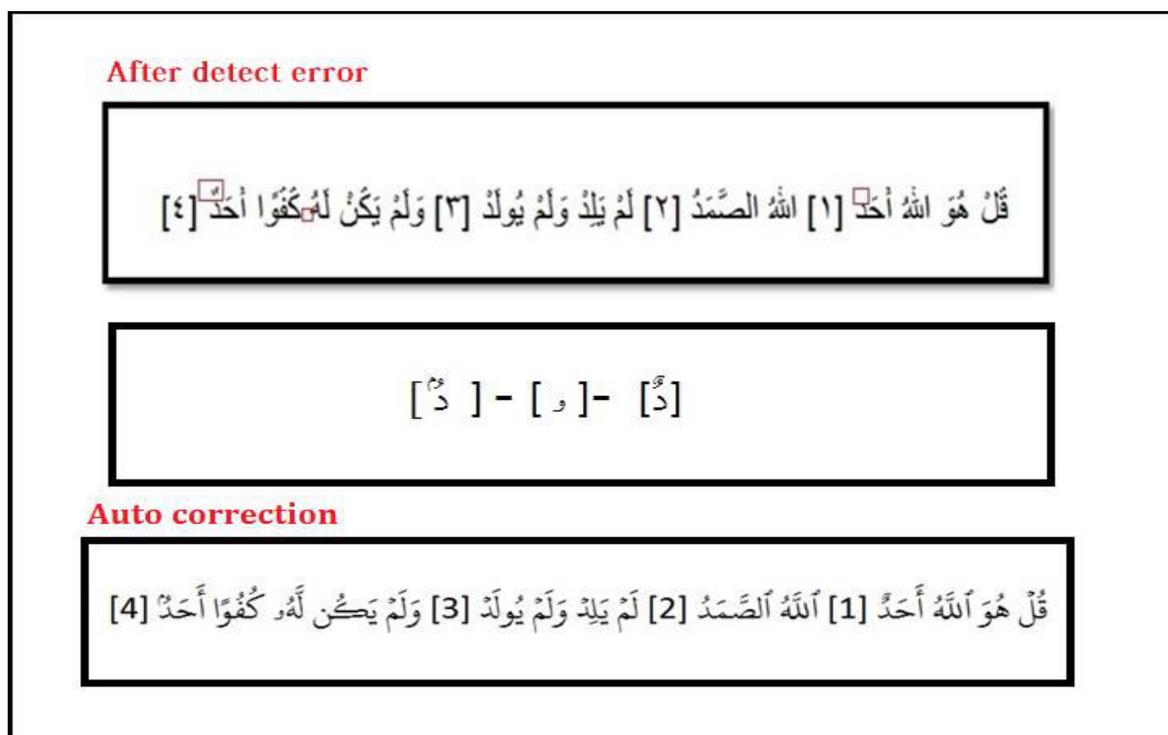


Figure 4.15 Auto correction

#### 4.7 Result Evaluation:

Table 4.2 show auditing system result.

Test number	sware number	Aya number	Number of Error [artificial]	Number of Detect Error	Percent	Note
1	1	4	1	1	100%	-
2	3	15	3	3	100%	-
3	6	29	5	4	80%	-
4	6	29	5	5	100%	Modify system
5	12	60	4	4	100%	-
6	16	98	6	5	83%	-
7	16	98	6	6	100%	Modify system
8	20	138	7	6	86%	-
9	20	138	7	7	100%	Modify system
10	24	193	9	9	100%	-
11	26	>193	5	4	80%	Add three sura out of database
12	26	>193	5	5	100%	Modify system

As shown in table 4.2 we test the auditing system at the beginning on one sura and create one error to check if system can detect it, we increase number of sura and add more than one error. In test number 3 in 6 sura we add 5 errors, system detect 4 of them. In test 5 we modify system database to detect the previous error.

In test number 11 we use 26 sura three of them out of out OCR database which build for 23 sura ,system detect 4 of 5 error in last test ,we modify system database to detect all error in previous test.

Note that the error has been corrected does not appear again.

#### **4.8 Summary:**

In this Chapter, Evaluation and testing accomplished on a “حزب” sample to determine the performance of the overall auditing system.

In result chapter, presented samples demonstrate all the steps of the AI sincerity at all stages of OCR. Sample shows the percentage increase accuracy when database development and that the system scalable, the model on the Quran from a small size, sample on the two-page surat, and displayed the database that has been built.

## Chapter 5

---

### Conclusion and Future Work

#### 5.1 Introduction

This Chapter summarizes the work and introduces the conclusion and the future work. The first section concludes the work done in this thesis. The second section summarizes the contributions added by the thesis. The obstacles that faced the progress shown in the third section. The final section covers the future work

#### 5.2 Conclusion

This thesis concentrates on Auditing electronic files of the Quran using optical character recognition for special Arabic font Uthmaani (الرسم العثماني).

Electronic files contain pages of Quran and compared electronically with their assets, before converting the files to the printing stage or publish the soft copy in order to reduce the time and manual effort during the audit of electronic files in the prepress stage, and reduce errors ratio that may occur during the electronic processing phase and finally detect errors that may appear, and documenting their location on each page electronically.

In this thesis it has been reviewed all OCR steps which use to Audit electronic file of Holy-Quran in detail, the algorithms used, output of each stage.

In Pre-process stage mention all image Modification and focus on image Skeletonization because different between Holy Quran font and printed Arabic font.

In segmentation stage, use horizontal projection, vertical projection, counter up and other algorithm to segment character.

In recognition stage, character classification using chan code and holy Quran character database. In this stage, a new algorithm called "Matrix Summation" was developed. Its basic idea comes from the idea of both horizontal and vertical projection (Y- projection & X- projection) and the Freeman chain code algorithms, using Matrix Summation we build database for all character, diacritics and adjustment sign to use in OCR system. We calculate vertical vector, horizontal vector, height and width for all character in the last 10 page in Holy Quran. Appendix A shows all character vectors.

Finally, in auditing, the outputs from previous Stage compare with reference-soft copy document - and detect error. After comparing the two texts together if the program discovered any differences between them is developing a box window in red on areas of difference in the two files. To increase the effectiveness of the system and their applicability to the entire Quran autocorrect developed. We detect error index, then determine error type (character, diacritics and adjustment sign), based on the type of error, replace the error with correction from approved reference.

If the error in the characters, replaced the character in the matrix generated after segmented and identify stage, in the same way diacritics and adjustment sign replaced.

### 5.3 Contribution

The following points summarize the main contributions in this thesis:

- Using optical character recognition to audit the electronic files of the Quran in special font "Uthmaani".
- In pre-process stage there is a skeleton for the image , due to the absence of a base line in Uthmanic Font ( الرسم العثماني ), as the other fonts , in this thesis a special algorithm build to find the base line.
- In the Segmentation stage, due to the method of drawing characters in Uthmaani font which is different from other Arabic fonts, special algorithm build for segment words and sub-words into character.
- In recognition stage , we build a new algorithm called "Matrix Summation", using Matrix Summation we build database for all character, diacritics and adjustment sign to use in OCR system. We calculate vertical vector, horizontal vector, height and width for all character in the last 10 page in Holy Quran.
- Also in the segmentation stage, the one character segmented into three levels, the character, diacritics, and signs of adjustment.
- In auditing stage, compare the output of OCR system with Printed of King Fahd Glorious Quran Printing Complex Saudi Arabia, Maddinah as a reference.

### 5.4 Obstacles

The main difficulty that faced develop automated audit electronic file of Holy Quran approach is the unavailability of database. No open source database s found for Holy Quran character. Building a database is time consuming and needs accuracy.

## **5.5 Future Work**

Although substantial progress has already been made toward the ultimate goal of auditing electronic files of Quran using optical character recognition, still many challenges exists and OCR auditing systems compared to human auditing capabilities are primitive.

The system has been tested up to 23 Surah of the Holy Quran in 10 pages And build a database of these pages contain letters, Adjust signs and diacritics.

In the next stages of the development of the system should be able to auditing all Holy Quran 114 Surah present in the 600-page through database development.

In the system, development phase to add new page now modifying the database manually and add new character and signs, in the next stages to include new character and add new page database modification must have done automatically and the system will be able to add new cases automatically.

We look forward to modify the system to be compatible with other copies of the Quran produced by multiple presses rather than copies prints in king Fahd Glorious Quran Printing Complex Saudi Arabia , Maddinah.

## Reference

---

- [1] David A. Forsyth and Jean Ponce, 'Computer Vision: A Modern Approach', Pearson Prentice Hall, NJ, USA, 2003.
- [2] Datong Chen, Kim Shearer and Hervé Bourlard, 'Video OCR for Sport Video Annotation and Retrieval ' Proceedings of then 8th International Conference on Mechatronics and Machine Vision in Practice, (M2VIP 2001), Hong Kong 27-29 August 2001 ., pp57-62
- [3] ImageFormats URL: <https://dev.havenondemand.com/docs/ImageFormats.html> viewed in 14-10-2016..
- [4] Tappert, C. C.; Suen, C. Y.; Wakahara, T. "The state of the art in online handwriting recognition".(1990).
- [5] T. Reiss, Recognizing Planar Objects Using Invariant Image Features, Springer-Verlag, 1993.
- [6] Qing Chen A, Evaluation of OCR Algorithms for Images with Different Spatial Resolutions and Noises, 2003.
- [7] Parker JR. Algorithms For Image Processing and Computer Vision. John Wiley & Sons, 1997.
- [8 ] Bahri, Z.; and Kumar, B. V. K. 1988. Generalized Synthetic Discriminant Functions. J. Opt. Soc. Am. A 5 (4) 562–571.
- [9] Hadjar, K., Ingold, R. (2003) Arabic News paper segmentation. In: Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), pp 895-899. IEEE COMPUTER SOCIETY.
- [10] M. S. Khorsheed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)", Pattern Recognition Letters 28 (2007) pp. 1563-1571.
- [11] Ejaz Raqum, Usool-o-Qavaid Khush Naveesi, Mir Muhammad Kutub Khana, Arram Bagh, Karachi, Pakistan .
- [12] Elaine Rich and Kevin Knight, 'Artificial Intelligence', 2nd Edition, McGraw Hill, NY, USA, 1991.
- [13] SOHAIL ABDUL SATTAR ," A TECHNIQUE FOR THE DESIGN AND IMPLEMENTATION OF AN OCR FOR PRINTED NASTALIQUE TEXT" . 2009.
- [14] Shafii, Mahnaz, "Optical Character Recognition of Printed Persian/Arabic Documents" (2014). Electronic Theses and Dissertations. Paper 5179.

- [15] J. Said, M. Cheriet and C. Suen, "Dynamic morphological preprocessing: a fast method for baseline extraction," ICDAR, pp. 8-12, 1996.
- [16] L. Xu, E. Oja and P. Kultanen, "A new curve detection method: Randomized Hough Transform (RHT)," Pattern Recognition Letters, vol. 11, pp. 331-338, 1990.
- [17] H. Cao, R. Prasad and P. Natarjan, "A stroke regeneration method for cleaning rule lines in handwritten document images," in Proceeding of the international workshop on multilingual OCR, pp. 1-10, New York, NY, 2009.
- [18] T. Hoang, E. Smith and S. Tabbone, "Sparsity-based edge noise removal from bilevel graphical document images," International Journal on Document Analysis and Recognition, vol. 17, no. 2, pp. 161-179, 2014.
- [19] W. Peerawit and A. Kawtrakul, "Marginal noise removal from document images using edge density," in Proceeding of fourth information and computer engineering, 2004.
- [20] F. Shafait, J. van Beusekom and D. Keysers, "Document cleanup using page frame detection," IJDAR, vol. 11, no. 2, pp. 81-96, 2008.
- [21] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," Pattern Recognition, vol. 33, no. 2, pp. 225-236, 2000.
- [22] R. Parvathi, N. Javanthi and et al., "Intuitionistic fuzzy approach to enhance text documents," in Proceedings 3rd IEEE international conf. on intel. systems, 2006.
- [23] C. Leung, H. Chan and et al., "A new approach for image enhancement applied to low-contrast low-illumination documents," Pattern Recognition Letters, vol. 26, no. 6, pp. 769-778, 2005.
- [24] S. Nomura, K. Yamanaka and et al., "Morphological preprocessing method to thresholding degraded word images," Pattern Recognition Letters, vol. 30, no. 8, pp. 729-744, 2009.
- [25] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Itrans. Syst., vol. 9, no. 1, pp. 62-66, 1979.
- [26] G. Lazzara and T. Géraud, "Efficient multiscale Sauvola's binarization," International Journal on Document Analysis and Recognition, vol. 17, no. 2, pp. 105-123, 2014.
- [27] W. Niblack, An introduction to digital image processing, Englewood Cliffs: Prentice Hall, N. J., pp. 115-116, 1995.

- [28] A. Benouareth, . A. Ennaji and M. Sellami, Semi, "Semi-continuous HMMs with explicit state duration for Arabic word modeling and recognition," *Pattern Recognition Letters*, vol. 29, pp. 1742-1752, 2008.
- [29] A. Vinciarell, "Application of information retrieval techniques to single writer documents," *Pattern Recognition Letters*, vol. 22, pp. 1043-1050, 2001.
- [30] S. Mozaffari, K. Faez and et al., "Lexican reduction using dots for off-line Farsi/Arabic handwritten word recognition," *Pattern Recognition Letters*, pp. 724-734, 2008.
- [31] H. Al-Youse and S. S. Udpa, "Recognition of Arabic character," *IEEE Trans.Pattern. Anal. Mach. Intell.* , vol. 14, no. 8, pp. 853-857, 1992.
- [32] S. Nasrollah and A. Ebrahimi , "Printed Persian Subword Recognition Using Wavelet Packet Descriptors," *Journal of Engineering (Open Access)*, vol. Article ID 465469, 2013.
- [33] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. info.theory*, vol. 8, pp. 179-187, 1962.
- [34] C. H. Teh and R. T. Chin, "Invariant image recognition by Zernike moments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 10, no. 4, 1988.
- [35] URL:[http://en.wikipedia.org/wiki/List\\_of\\_languages\\_by\\_writing\\_system](http://en.wikipedia.org/wiki/List_of_languages_by_writing_system) viewed in 1/10/2016
- [36] A. Cheung, M. Bennamoun\*, N.W. Bergmann "An Arabic optical character recognition system using recognition-based segmentation" 2001 .
- [37] DRAFT. "A survey of modern optical character recognition techniques ", 2004
- [38] Hadjar, K., Ingold, R. (2003) Arabic News paper segmentation. In: Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), pp 895-899. IEEE COMPUTER SOCIETY.
- [39] Majid M. Altuwaijri and Magdy A. Bayoumi, "Arabic Text Recognition Using Neural Networks", IEEE International Symposium on Circuits and Systems, ISCAS'94, London, UK, 30 May- 02 June,1994, Vol. 6, pp. 415-418.
- [40] Albadr B, Haralick R, Segmentation-free approach to text recognition with application to Arabic text, *International Journal on Document Analysis and Recognition*, (1998) 1: 147-166.
- [41] M. Pechwitz and V. Ma"rgner, "HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database," Proc. Int'l Conf. Document Analysis and Recognition, pp. 890-894, 2003.

- [42] S. Alma'adeed, C. Higgins, and D. Elliman, "Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach," Proc. 16th Int'l Conf. Pattern Recognition, vol. 3, pp. 481-484, 2002.
- [43] S. Alma'adeed, C. Higgins, and D. Elliman, "Off-Line Recognition of Handwritten Arabic Words Using Multiple Hidden Markov Models," Knowledge-Based Systems, vol. 17, pp. 75-79, 2004.
- [44] N. Farah, L. Souici, L. Farah, and M. Sellami, "Arabic Words Recognition with Classifiers Combination: An Application to Literal Amounts," Proc. Artificial Intelligence: Methodology, Systems, and Applications, pp. 420-429, 2004.
- [45] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling," Proc. Int'l Conf. Document Analysis and Recognition, pp. 893-897, 2005.
- [46] Hamid, A. and Haraty, R., "A Neuro-Heuristic Approach for Segmenting Handwritten Arabic Text", ACS/IEEE International Conference on Computer Systems and Applications, Beirut, Lebanon, 25-06-2001 – 29-06-2001, pp: 110-113.
- [47] M. S. Khorsheed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)", Pattern Recognition Letters 28 (2007) pp. 1563-1571.
- [48] Elaine Rich and Kevin Knight, 'Artificial Intelligence', 2nd Edition, McGraw Hill, NY, USA, 1991.
- [49] Albadr B, Haralick R, Segmentation-free approach to text recognition with application to Arabic text, International Journal on Document Analysis and Recognition, (1998) 1: 147-166.
- [50] I. Bazzi, R. Schwartz, and J. Makhoul, "An Omnifont Open-Vocabulary OCR System for English and Arabic," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, pp. 495-504, 1999.
- [51] Obaid AM, Dobrowiecki TP. Heuristic Approach to the Recognition of Printed Arabic Script.
- [52] R. Ramsis, S.S. El-Dabi, and A. Kamel, Arabic Character Recognition System, IBM Kuwait Scientific Centre, report No. KSC027, 1988.
- [53] Khorsheed MS, Clocksin WF. Spectral features for Arabic word recognition. The IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'2000, Istanbul, Turkey, June 5-9, 2000, pp. 3574-3577
- [54] Bouslama F, Kishibe H. Fuzzy logic in the recognition of printed Arabic text. IEEE Transactions on 1999: 1150-1154.

- [55] Zidouri A, Sarfraz M, Shahab SA, Jafri SM. Adaptive dissection based subword segmentation of printed Arabic text. *IEEE Transactions on* 2005: 239-243.
- [56] Motawa D, Amin A, Sabourin R, Segmentation of Arabic Cursive Script. *Proceedings of the 4th International Conference on Document Analysis and Recognition*, 1997: 625 – 628.
- [57] Tolba M, Shaddad E. On the automatic reading of printed arabic characters. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Los Angeles, CA, 1990; 496–498
- [58] Al-Yousefi H, Udpa S. Recognition of Arabic characters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1992; 14(8): 853-857.
- [59] Amin A, Mari J. Machine recognition and correction of printed Arabic text. *IEEE Transactions on Man, Machine and Cybernetics* 1989; 19(5): 1300-1306.
- [60] Alemami S, Usher M. On-line recognition of handwritten Arabic characters. *IEEE Trans Pattern Analysis and Machine Intelligence* 1990; 12(7): 704–710
- [61] A. Zahour, B. Taconet, and A. Faure, "Machine Recognition of Arabic Cursive Writing", in *From Pixels to Features III: Frontiers in Handwriting Recognition*, ed. S. Impedovo and J.C. Simon. Amsterdam: Elsevier Science Publishers B.V., 1992, pp. 289-296.
- [62] S. I. Abuhaiba, "Recognition of Off-Line Handwritten Cursive Text," Ph.D. thesis, Department of Electronic and Electrical Engineering, Loughborough University, Loughborough, U. K., 1996.
- [63] Haraty, R. and Ghaddar, C. Neuro-Classification for Handwritten Arabic Text. *Proceedings ACS/IEEE International Conference on Computer Systems and Applications*, 2003.
- [64] Amin, A. Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming. *Pattern Recognition Letters*, vol. 24, pp. 3187-3196, 2003.
- [65] A. Alaei, P. Nagabhushan, U. Pal, "A new two-stage scheme for the recognition of Persian handwritten characters," in *Proc. of 12th ICFHR*, pp.130-135, 2010.
- [66] A. A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network," *Pattern Recognition*, vol. 43, no. 7, pp. 2582-2589, July 2010.
- [67] D. V. Sharma, P. Jhaji, "Recognition of isolated handwritten characters in Gurmukhi script," *International Journal of Computer Applications*, vol. 4, no. 8, pp. 9-17, 2010.
- [68] M. Kumar, M. K. Jindal, R. K. Sharma, "k-NN based offline handwritten Gurmukhi character recognition," in *Proc. of ICIP*, pp. 1-4, 2011.

- [69] Pal & Chaudhuri, 2004 "OCR System: A Literature Survey " , 1993
- [70] "جهود مجمع الملك فهد لطباعة المصحف الشريف في استخدام التقنيات المعاصرة , علي بن عبد الله برناوي " عام 1430 هجري. لخدمة القرآن الكريم. "
- [71] Khader Mohammada, Muna Ayyeshb, Aziz Qaroushc ,Iyad Tumar "Printed Arabic Optical Character Segmentation" 2015.
- [72]. M. Sarfraz, S. N. Nawaz, A. Al-Khuraidly, "Offline Arabic Text Recognition System", Proc. of Int. Conf. on Geometric Modelling and Graphics (GMAG'03),2003.
- [73] Haraty, R. and Ghaddar, C. Neuro-Classification for Handwritten Arabic Text. Proceedings ACS/IEEE International Conference on Computer Systems and Applications,2003.
- [74] Amin, A. Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming. Pattern Recognition Letters, vol. 24, pp. 3187-3196, 2003.
- [75] A. Alaei, P. Nagabhushan, U. Pal, "A new two-stage scheme for the recognition of Persian handwritten characters," in Proc. of 12th ICFHR, pp.130-135, 2010.
- [76] A. A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network," Pattern Recognition, vol. 43, no. 7, pp. 2582-2589, July 2010.
- [77] D. V. Sharma, P. Jhajj, "Recognition of isolated handwritten characters in Gurmukhi script," International Journal of Computer Applications, vol. 4, no. 8, pp. 9-17, 2010.
- [78] Niyazi Kilic, Pelin Gorgel, Osman N. Ucan, Ahmet Kala.;"MULTIFONT OTTOMAN CHARACTER RECOGNITION USING SUPPORT VECTOR MACHINE" ISCCSP 2008.
- [79] Ozturk, A.; Gunes, S.; Özbay, Y." Multifont Ottoman character recognition"Electronics, Circuits and Systems, 2000. ICECS 2000. The 7th IEEE International Conference on  
Year: 2000, Volume: 2
- [80] G. Tleaa , M. Kader, A. Badran, Arabic letters recognition based on BP and perceptron and compare the performance of them , Iraqi Journal of Statistical Sciences.
- [81] C.P. Sumathi1, T. Santhanam , G.G. Devi , A SURVEY ON VARIOUS APPROACHES OF TEXT EXTRACTION IN IMAGES ,International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.4, August 2012
- [82] Mathworks website ,URL: <http://www.mathworks.com/help/images/ref/im2bw.html> , Documentation Center , viewed in 14-10-2016.

- [83] Mathworks website, URL: <http://www.mathworks.com/help/images/ref/imfilter.html> , Documentation Center , viewed in 14-10-2016
- [84] Mathworks website, URL: [http://www.mathworks.com/matlabcentral/newsreader/view\\_thread/44630](http://www.mathworks.com/matlabcentral/newsreader/view_thread/44630) , Matlab central ,viewed in 14-10-2016.
- [85] Graphics Software,URL:<http://graphicssoft.about.com/od/aboutgraphics/a/bitmapvector.htm> , Vector and Bitmap Images, viewed in 14-10-2016.
- [87] Plsn website , URL : <http://www.plsn.com/current-issue/39-video-digerati/5137-the-benefits-of-png.html> , The Benefits of PNG, viewed in 14-10-2016.
- [88] Wekibidia website , URL: [http://en.wikipedia.org/wiki/Topological\\_skeleton](http://en.wikipedia.org/wiki/Topological_skeleton) , Topological skeleton, viewed in 14-10-2016.
- [89] Safwa Taha, Yusra Babiker, and Mohamed Abbas, Optical Character Recognition of Arabic Printed Text. IEEE student Conference on research and development,2012.

## Appendix A

---

### "Matrix Summation" as character features

#### Horizontal vector:

{1,2,2,2,2,2,3,3,3,2,2,2,2,2,2,3,3,3,3,2,2,2,2,2,2,2,1,1},//ا  
{2,2,4,3,4,4,3,3,4,3,4,6,8,38,37,36,32,1,0,0,0,0,2,4,5,3,2,1},//ب  
{1,4,6,6,6,5,3,1,0,0,3,2,4,5,3,3,4,3,3,4,5,7,19,37,35,33,30,1},//ت  
{2,4,4,3,1,3,5,7,7,6,3,1,0,0,2,2,4,4,3,4,3,4,3,4,5,6,9,37,36,34,30,1},//ث  
{10,22,23,18,16,9,3,3,2,1,1,2,2,2,1,1,4,5,5,4,2,1,1,1,2,2,3,4,5,6,8,10,21,19,14,8,4,1},//ج  
{2,17,23,23,14,7,5,4,3,2,3,1,1,1,2,1,1,1,1,1,1,1,1,2,3,3,4,5,6,9,13,22,21,15,12,1},//ح  
{2,4,5,4,1,0,0,0,0,0,10,22,23,23,18,12,6,3,3,4,2,1,2,1,1,1,1,0,0,1,1,1,1,2,2,2,3,4,5,7,10,18,21,18,13,7,1},//خ  
{2,3,3,3,3,2,2,2,1,1,2,1,1,1,1,1,1,15,15,15,6,1},//د  
{2,3,5,3,2,0,0,0,0,0,0,2,3,4,3,3,3,2,2,1,2,1,1,1,1,1,1,2,15,15,15,14,1},//ذ  
{1,2,3,4,4,3,3,3,2,2,2,1,1,1,2,2,2,2,3,4,4,4,5,6,5,12,11,2,1},//ر  
{3,4,4,2,1,0,0,0,0,0,0,1,2,3,4,4,3,3,2,2,2,1,1,1,1,2,3,3,3,4,4,4,5,5,7,13,7,1,1},//ز  
{1, 1, 3, 5, 5, 4, 3, 3, 4, 5, 9, 19, 18, 15, 6, 4, 2, 2, 4, 4, 6, 8, 14, 22, 18, 10, 2, 2},//س  
{1,4,5,3,2,2,2,4,4,6,4,2,2,0,0,0,0,0,2,4,5,4,4,4,4,5,7,18,17,14,4,5,3,2,3,2,5,8,11,17,21,18,15,12,2},//ش  
{3,4,16,19,20,12,11,8,8,8,14,23,30,34,35,35,1,2,3,4,5,8,9,8,13,18,13,4},//ص  
{3,3,5,4,0,0,0,0,0,0,6,8,9,12,14,6,5,6,7,10,26,30,32,32,32,1},//ض  
{2,3,5,5,5,3,2,1,1,2,1,1,1,2,2,2,2,1,1,1,1,2,3,13,16,19,15,10,8,7,9,13,32,35,34},//ط  
{1,2,5,5,5,4,2,2,2,2,5,5,5,4,2,1,1,1,2,2,2,3,4,11,14,16,18,12,9,8,8,11,35,34,34,14},//ظ  
{8,10,9,6,3,2,2,2,3,5,9,10,14,18,14,12,7,5,3,3,2,1,2,2,2,2,2,2,1,1,3,3,3,3,4,4,5,7,9,22,20,18,12,3},//ع  
{1,1,4,4,4,1,0,0,0,0,0,0,1,8,9,12,5,3,1,2,2,3,6,9,11,15,17,15,10,7,5,3,2,2,2,2,1,1,1,1,1,2,2,2,2,3,2,3,3,4,6,9,22,20,16,12},//غ  
{1,2,5,4,3,0,0,0,0,0,0,1,4,5,8,10,7,5,6,6,9,10,12,11,11,12,15,39,36,31},//ف  
{3,4,7,8,7,6,5,1,1,0,0,0,0,2,6,7,9,7,6,6,6,9,12,12,12,13,5,4,3,3,2,3,4,5,7,8,13,21,26,24,18,11,1},//ق  
{1,3,3,5,6,5,4,4,3,4,4,3,4,3,5,5,9,10,8,4,5,6,5,6,11,3,3,3,4,5,5,9,30,30,27,24},//ك  
{1,3,4,5,4,4,3,3,2,2,2,2,1,2,2,2,2,2,1,2,2,2,2,2,1,2,2,3,3,6,16,17,17,1,1},//ل  
{5,7,7,10,9,5,4,4,5,10,13,16,16,6,3,3,2,2,2,2,1,2,2,2,1,1,2,2,2,1,1},//م  
{1,0,0,0,0,1,5,5,5,3,0,0,0,0,1,1,2,3,3,2,3,3,2,3,3,3,3,4,3,3,5,5,6,9,12,22,20,17,14,6},//ن  
{1,3,7,8,8,5,4,2,3,3,4,5,8,11,10,8,4},//ه  
{4,6,7,8,4,4,3,3,6,9,8,7,5,2,2,2,2,3,3,3,3,4,5,7,10,11,7},//و  
{1,10,13,14,13,10,9,9,11,12,10,6,5,7,10,24,22,19,14,6,3,2,4,7,4},//ي  
{3,3,6,9,7,4,2,2,0,0,0,0,1,3,6,7,8,9,8,5,3,3,4,5,12,11,9,8,3},//هـ

{3,5,5,8,9,2,3,4,6,8,13,14,13,8,6,3,1,1},//ء  
 {1,4,4,2,3,4,4,0,0,1,1,2,2,3,3,3,3,3,3,3,2,2,3,3,3,3,3,3,3,3,2,2,2,3,2,2,1,1,1},//أ  
  
 {5,7,11,11,12,11,11,10,9,8,8,4,4,4,3,4,4,3,5,3,4,4,4,2,2,2,4,6,4,4,6,6,12,13,11,9,4},//لا  
 {3,7,10,11,10,10,9,8,8,8,7,4,4,3,4,4,4,2,4,4,4,4,3,4,4,5,6,6,5,8,8,11,10,9,6,0,0,0,2,3,4,4,4,3,2},//لأ  
 {3,3,1,2,5,4,2,0,0,3,5,10,12,12,11,12,9,9,9,9,7,5,4,5,4,4,4,3,3,2,5,3,3,3,3,4,5,3,4,5,7,12,12,11,7},//لأ  
  
 {1,2,3,3,4,3,4,2,2,2,1,1,14,13,14,7,1,1,0,0,0,0,1,3,5,5,5,1,1},//إ  
 {3,4,5,4,2,2,0,0,0,0,1,1,3,4,4,3,3,3,3,3,3,13,13,13},//ت  
 {2,4,5,4,1,5,4,7,9,10,5,4,3,0,0,0,0,1,2,3,4,4,4,3,2,2,1,2,1,19,19,20,19,18},//ث  
 {1,3,9,10,14,11,7,7,5,7,6,10,26,35,35,35,31,0,0,0,1,3,4,4,4,3},//ج  
 {7,10,11,6,4,5,5,4,4,5,6,17,29,29,29},//ح  
 {2,5,3,1,0,0,0,0,0,0,0,5,8,12,8,6,5,4,4,3,3,4,7,29,29,29},//خ  
 {1,3,2,2,2,1,1,2,4,7,7,6},//س  
 {4,4,4,3,4,2,5,7,10,4,3,3,0,0,0,0,1,2,3,2,3,4,4,4,7,14,18,18,17,13},//ش  
 {7,11,12,9,4,6,4,8,11,16,47,42},//ص  
 {2,3,3,0,0,0,0,0,0,1,7,11,14,11,6,5,6,8,11,13,35,48,41},//ض  
 {1,1,5,5,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,9,13,9,8,6,5,4,6,8,20,32,32},//ط  
 {2,3,5,4,3,2,1,1,2,4,6,4,2,1,1,1,1,1,1,1,2,7,12,15,11,8,6,5,5,6,10,32,32,32},//ظ  
 {6,12,14,15,11,3,2,3,2,2,4,8,12,21,24,21,17,14},//ع  
 {1,2,3,2,0,0,0,0,1,1,8,12,14,4,1,1,1,1,2,2,5,9,13,19,17,10},//غ  
 {1,2,4,4,5,3,1,0,0,0,1,2,6,8,8,9,6,6,6,7,9,11,10,10,4,2,2,2,2,20,20,20,19,18},//ف  
 {2,3,6,7,7,6,5,2,0,0,0,1,3,7,8,8,10,5,5,6,7,9,11,10,9,2,1,1,2,2,4,16,15,16},//ق  
  
 {3,5,8,5,4,2,2,1,2,1,3,4,5,6,7,7,4,4,3,3,3,2,2,2,2,1,2,2,2,21,22,6,2},//ك  
 {1,2,2,4,5,5,4,4,3,3,2,1,1,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2,2,3,3,9,9,9},//ل  
 {3,5,2,3,4,4,4,5,9,13,20,17,15,2},//م  
 {1,4,6,3,0,0,0,0,0,0,2,2,3,4,4,3,2,2,2,1,1,1,1,20,20,20,19},//ن  
 {2,3,5,5,6,6,9,11,7,6,8,7,5,8,11,11,12,20,19,14,8,5},//ه  
 {1,1,3,3,2,3,2,2,1,1,1,2,13,13,13,5,2,1,0,0,0,3,4,4,5,4,2,2},//و  
 {1,3,4,2,3,4,3,0,0,0,0,1,1,3,4,4,4,3,2,2,1,1,1,2,2,14,14,14,13,1},//ز  
 {1,2,2,4,4,4,3,2,2,2,2,1,2,3,2,2,2,2,2,2,2,2,2,3,2,3,2,2,2},//ح  
 {1,2,3,3,3,4,3,2,3,2,3,19,21,20,20,20,4,0,0,0,0,1,3,5,4},//ط

{2,3,5,10,6,5,2,1,0,0,0,0,1,4,4,4,3,2,3,2,2,2,2,15,15,15,14,7} //ث//  
 {1,5,5,5,3,3,4,6,8,11,6,4,2,0,0,0,0,2,3,3,4,4,4,2,2,2,1,1,3,7,14,14,14,14} //ثذ/  
 {3,8,10,12,7,6,6,5,6,5,6,18,30,29,31,0,0,0,0,0,2,4,4,2} //جـ/  
 {5,8,12,13,11,7,7,6,7,7,6,25,25,25,17} //حـ/  
 {2,5,4,1,0,0,0,0,0,0,1,7,9,12,9,7,6,5,6,5,5,7,28,28,26} //خـ/  
 {1,3,4,2,2,1,1,1,0,1,1,1,1,1,1,1,1,2,13,13,13} //دـ/  
 {2,5,2,0,0,0,0,0,0,0,1,1,2,2,1,1,1,1,0,1,1,1,1,1,1,1,1,2,2,13,13,11} //ذـ/  
 {1,1,2,2,2,1,1,1,1,1,1,1,1,1,3,1,3,2,3,3,3,3,3,6,8,11,6} //رـ/  
 {3,4,3,1,0,0,0,0,0,0,0,1,1,2,3,2,1,1,1,1,0,1,2,0,1,1,3,1,3,2,3,3,3,5,3,7,7,12,7} //زـ/  
 {1,1,2,2,3,3,3,3,4,4,6,9,13,20,20,18,11} //سـ/  
 {2,4,3,2,1,4,4,2,4,1,0,0,0,0,0,0,1,1,1,1,2,2,3,5,13,15,29,29,14} //شـ/  
 {7,9,12,14,10,8,5,6,10,16,33,35,35,35,16} //صـ/  
 {1,2,3,5,4,3,0,0,0,0,0,1,8,11,14,15,11,9,5,5,10,14,26,37,37,37} //ضـ/  
 {2,4,5,5,4,3,2,2,1,1,2,2,2,1,2,2,2,2,2,7,11,14,16,17,11,8,6,6,12,33,33,33,33} //ظـ/  
 {1,3,5,5,4,3,2,1,3,4,6,3,1,1,1,1,1,1,1,1,1,10,12,15,8,11,6,5,3,8,18,32,32,25} //طـ/  
 {8,11,13,12,11,10,7,7,6,11,32,35,31,27,14,4} //عـ/  
 {1,3,5,5,4,2,0,0,0,0,0,1,5,12,12,13,12,12,7,6,7,10,12,26,26,23,17,1} //غـ/  
 {1,3,6,4,2,1,0,0,0,0,3,5,7,7,5,6,6,11,12,12,12,11,10,2,2,2,2,14,14,14,14,1} //فـ/  
 {2,5,4,7,3,2,0,0,0,0,0,2,2,5,7,8,6,5,4,4,4,4,6,7,6,7,7,14,18,18,2} //قـ/  
 {2,4,7,9,8,6,5,3,2,2,1,2,4,4,8,6,7,7,5,4,4,4,3,3,3,2,1,3,1,3,23,22,23,18} //كـ/  
 {2,3,5,6,5,5,4,3,3,3,3,2,2,2,3,2,2,2,2,2,3,3,2,2,2,1,11,11,11,11,11} //لـ/  
 {2,4,5,7,5,4,5,7,10,17,27,26,24,17,2,1} //مـ/  
 {2,4,4,5,3,1,0,0,0,0,1,1,2,3,4,4,3,2,1,1,1,1,2,9,15,15,15,15} //نـ/  
 {2,3,5,6,7,8,4,2,3,3,4,2,5,14,22,22,22,20,6,5,3,4,4,4,6,6,8,6,5} //هـ/  
 {3,5,7,6,3,3,2,2,3,7,7,8,5,1,1,1,2,3,3,2,2,4,5,5,6,11,8,2} //وـ/  
 {1,2,3,4,4,3,3,2,2,2,1,2,2,13,13,13,12,5,0,0,0,1,4,6,6,8,4,4,3} //زـ/  
 {2,3,9,7,5,5,1,0,0,0,0,1,1,2,3,3,4,6,7,5,6,5,6,9,12,14,15,12,12,1,2,2,1} //حـ/  
 {1,4,6,8,9,10,10,11,11,7,4,3,2,3,3,2,1,1,0,0,0,0,1,2,3,3,3,5,5,6,7,9,10,12,7} //طـ/  
  
 {1,1,2,3,4,5,3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2,2,3,3,8,7,6,2} //لـ/  
 {2,3,4,5,4,4,5,5,4,5,6,5,8,10,38,37,35,33,0,0,0,0,0,2,5,6,4,2} //بـ/  
 {2,4,5,8,8,5,4,1,0,0,2,3,5,5,4,3,4,3,4,4,5,6,10,36,37,36,34,29} //تـ/  
 {3,5,5,3,1,3,68,9,6,5,2,1,0,1,2,3,5,5,4,3,4,4,4,4,4,7,10,37,35,34,31} //ثـ/

{19,23,24,25,20,9,8,5,4,5,6,12,12,12,10,5,1,3,5,7,6,3,2,2,3,3,3,4,5,6,8,11,15,21,18,14,6} //ج

{14,22,24,24,25,12,7,6,5,6,6,7,10,11,9,8,1,1,2,2,1,2,2,2,3,3,3,5,6,8,9,14,23,19,15,8} //ح

{4,6,5,4,0,0,0,0,0,0,0,22,24,24,25,13,9,7,5,6,4,5,12,10,10,9,3,1,1,1,1,1,1,3,2,2,3,3,4,5,7,9,11,23,21,16,13} //خ

{2,3,4,5,4,4,2,2,2,2,2,2,2,1,1,1,3,16,16,15,15,10} //ـ

{2,4,6,4,3,0,0,0,0,0,0,1,2,3,4,5,4,3,2,2,2,1,2,2,2,2,1,1,2,14,16,16,15,15} //ذ

{1,3,3,4,4,5,3,3,2,1,1,1,1,1,2,2,2,3,4,4,3,5,5,6,7,13,15,11,5} //ر

{1,5,6,6,3,0,0,0,0,0,0,0,2,2,4,5,5,4,4,4,3,3,2,2,2,2,2,3,4,3,4,4,5,6,6,7,11,15,12,8} //ز

{1,3,4,8,7,6,6,4,6,7,10,19,19,19,16,2,4,3,4,6,7,10,13,24,21,18,14,6} //س

{2,3,6,5,3,1,4,6,8,8,5,4,1,0,0,0,0,0,2,4,6,7,7,8,7,6,5,6,14,20,19,20,16,3,3,2,4,5,7,10,16,23,20,18,13,4} //ش

{4,11,14,18,21,17,12,12,9,9,13,22,35,35,35,3,3,3,4,5,8,11,15,22,20,17,11} //ص

{1,2,5,5,2,1,0,0,0,0,0,7,13,16,19,21,13,11,9,10,10,13,35,35,34,35,7,2,2,3,5,6,9,11,23,21,18,15,9} //ض

{1,2,6,6,6,4,3,2,2,2,2,2,2,2,3,2,2,2,2,2,5,9,15,17,18,15,11,10,7,8,16,35,35,35,35} //ط

{1,2,5,6,6,5,4,2,2,3,6,7,7,4,3,2,2,2,2,2,2,9,14,16,18,15,11,9,8,9,12,35,35,35,35} //ظ

{4,7,9,10,11,11,8,6,7,6,8,10,12,14,13,12,9,1,2,2,2,1,1,2,2,3,3,4,3,4,6,6,9,12,20,18,15,11} //ع

{3,5,5,3,0,0,0,0,0,0,0,3,8,10,12,12,12,8,7,7,7,9,12,12,15,15,11,10,2,2,2,1,1,2,2,2,2,3,4,4,5,7,10,14,21,18,15,10} //غ

{3,5,5,4,2,0,0,0,0,0,0,3,5,9,9,9,7,6,6,7,7,11,13,12,14,42,41,39,37,25} //ف

{2,3,6,7,8,5,3,1,0,0,0,0,0,3,6,7,9,9,7,6,5,7,11,13,13,12,8,2,2,2,2,3,3,5,5,7,9,14,22,25,22,18,10} //ق

{1,2,4,6,6,5,5,3,4,4,4,4,4,4,11,11,7,4,4,5,6,6,13,3,3,4,5,4,6,10,31,31,28,26} //ك

{1,3,4,5,5,4,3,2,2,2,1,2,2,2,2,2,2,2,3,3,4,3,3,2,2,3,4,5,5,6,7,6,8,9,21,19,17,14,8} //ل

{4,8,8,10,8,6,5,5,6,12,14,16,17,6,4,2,2,2,2,1,2,2,2,2,1,2,2,2,1,1,1} //ـ

{2,5,5,3,2,0,0,0,0,1,2,3,4,3,3,2,2,2,1,2,2,2,2,4,5,6,5,10,13,25,24,21,17,9} //ن

{2,2,2,2,2,4,7,8,4,6,5,6,14,14,14,14,10,2,3,3,2,2,2,1} //ـ

{3,5,7,8,8,5,4,4,5,8,10,9,8,5,2,2,2,3,4,4,4,5,5,6,10,13,10,2} //و

{1,2,13,17,18,18,12,8,9,12,13,13,9,7,6,8,19,20,16,11,2,2,4,9,9} //ي

{2,2,4,3,5,5,4,3,0,0,0,1,2,2,2,2,2,4,5,6,8,6,4,5,7,8,11,14,12,11,3,3,2,2,3,2,1} //ـ

{1,12,14,17,16,8,8,10,12,13,11,6,7,7,12,25,23,21,17,11} //ى

{1,2,6,7,6,5,2,2,2,1,2,2,2,1} //٤

{4,6,6,5,3,3} //.

/\*

{1,2,2,3,2,1,1,1,2,3,7,7,7,5,3} //1س

{1,1,2,2,2,2,2,10,10,9,7,5} //2س

{1,2,2,3,3,4,4,4,2,2,2,3,3,4,3,3,3,4,4,4,8,8,13,17,22,17,15,6} //3س

\*/

Vertical vector:

{13,25,20,8} //ا

{13,12,10,7,4,4,4,4,4,4,4,4,4,4,4,4,7,9,9,7,5,4,4,4,4,4,4,4,5,4,4,4,6,5,5,7,7} //ب

{14,12,9,6,4,4,4,4,4,4,4,4,4,4,7,7,8,6,5,0,6,9,10,9,8,5,5,5,4,4,5,4,5,3,5,6,9,6} //ت

{12,11,10,7,4,4,4,4,4,4,4,4,4,6,9,7,7,6,5,9,12,12,10,7,4,4,4,4,5,4,4,5,4,5,5,6,7,6} //ث

{5,6,6,8,8,9,10,12,12,11,14,10,12,9,11,11,13,8,10,7,9,9,9,10,10,8,10} //ج

{5,6,5,8,8,8,8,9,9,10,11,11,12,13,11,14,12,10,12,13,12,11,10,10,8,10} //ح

{6,7,8,7,9,9,10,10,11,16,15,16,15,14,12,11,12,10,12,9,10,10,9,7,6} //خ

{11,8,6,6,6,6,6,3,4,4,4,5,3,4} //د

{12,9,8,8,8,7,8,5,8,9,8,5,5,4,4} //ذ

{11,14,10,11,9,5,3,3,4,3,3,2,2,3,2,1,2,1,1,0,2,1,1,2,1} //ر

{12,12,11,11,7,6,9,7,5,4,2,3,2,2,2,3,1,1,0,2,1,2,0,1,1} //ز

{8,10,7,3,4,5,4,6,5,5,3,4,4,3,3,3,3,6,10,9,8,4,3,3,3,4,3,2,3,4,3,3,3,3,4,2,3,3,4,4,6,7,6} //س

{6,9,5,3,7,5,8,8,5,8,11,10,11,6,4,3,4,6,9,11,8,4,4,4,3,4,4,4,5,5,4,4,5,5,4,4,5,5,4,4,5,5,9} //ش

{13,14,15,12,8,7,8,6,6,6,7,7,8,5,8,8,9,6,8,7,7,9,8,6,6,7,5,5,5,5,5,6,11,10,11,7,4,3,5,3,2,3,3,4,2,2,3,3,3,3,2,3,4,3,3,4,4,6,9} //ص

{4,5,4,5,5,5,5,6,7,11,12,10,9,6,6,5,6,7,7,8,9,9,10,10,10,10,11,12,11,9,7} //ض

{11,13,12,11,8,9,6,6,6,9,7,7,8,7,8,6,8,6,7,8,24,23,17,12,12,5,5,6,4,4,4,4,4,4} //ط

{10,13,14,14,12,9,9,10,12,11,13,10,8,6,6,8,7,8,8,13,26,22,14,11,9,7,5,4,4,4,4,4,4,5,2} //ظ

{0,2,17,23,20,15,17,19,16,15,12,13,12,10,9,10,7,8,9,8,8,8,7,6,3,2,1,1} //ع

{1,2,2,3,7,9,7,8,9,9,7,8,10,10,12,15,16,17,18,16,16,14,12,14,17,16,5} //غ

{3,16,18,13,12,12,13,17,15,15,6,3,4,3,4,4,5,5,9} //ف

{0,6,12,11,4,4,5,4,4,5,4,4,4,4,4,8,9,11,10,14,14,18,15,17,23,17,19,16,15,16,6} //ق

{13,23,20,18,13,9,4,4,5,10,12,13,11,12,12,11,9,5,5,5,4,5,4,4,4,4,4,5,8,10} //ك

{5,6,3,3,3,3,3,3,3,6,8,13,20,22,14,8} //ل

{4,8,9,10,10,9,9,9,8,7,8,6,3,9,13,12,5} //م

{0,7,11,3,4,4,4,4,4,5,5,8,10,9,9,7,4,5,5,5,4,4,5,8,12,17,13,1} //ن

{5,8,8,8,8,6,8,8,9,7,7,7,4} //ه

{10,14,14,12,13,12,11,14,12,4,4,3,3,2,1,1,1,1,1,1,1} //و

{0,6,10,7,7,6,5,4,4,5,5,7,10,9,4,5,7,11,11,11,11,14,12,13,15,13,10,4,4,4,5,3,3} //ي

{0,3,4,8,12,11,9,13,14,14,13,10,9,7,8,6} //ب

{0,5,7,9,12,9,10,10,9,9,7,5,5,4,4,4,3} //ة

{0,7,16,24,32,17,6} //أ

{0,1,5,7,8,8,8,7,7,5,5,9,8,9,9,10,12,10,11,11,9,12,18,18,15,6,6} //لا  
 {0,2,6,8,8,8,7,6,5,3,12,14,15,15,12,11,8,10,10,13,14,15,17,9,6,5} //لا  
 {0,8,13,13,11,10,9,7,6,4,7,8,9,8,11,9,9,8,10,11,14,12,16,17,12,6,4} //لا  
  
 {3,3,3,3,4,6,9,8,9,11,12,10,12,6} //ب  
 {3,3,7,5,7,4,3,6,9,18,16,9,3,0} //ب  
 {5,5,5,5,5,5,8,14,14,13,12,10,10,10,14,16,15,14,2} //ب  
 {4,6,4,4,4,5,5,6,4,4,5,6,8,8,9,9,14,17,14,16,11,9,11,11,10,12,8,9,9,5,7,4,4,5,4} //ج  
 {3,3,4,5,6,7,8,9,9,7,6,7,7,7,6,7,6,6,6,7,7,6,5,4,4,4,3,3,1} //ح  
 {3,4,4,4,5,6,7,8,6,6,7,7,7,8,10,10,10,8,7,6,6,5,4,4,3,3,3,3,3} //خ  
 {4,5,4,5,7,10,10} //س  
 {5,5,4,6,12,13,12,11,7,14,13,10,7,8,6,8,12,10} //ش  
 {8,10,11,8,7,7,6,6,6,6,5,5,6,4,4,5,3,4,4,3,4,6,5,5,4,4,4,3,3,3,3,3,3,5,4,5,7,5,3,3,3,3,3,3,3} //ص  
 {4,9,11,9,6,7,7,6,6,6,7,5,7,5,6,4,6,5,5,5,8,6,8,5,6,4,4,4,2,2,2,2,2,3,3,3,5,7,4,3,1,2,2,2,2,4,1} //ض  
 {8,10,11,8,5,4,4,4,5,3,3,4,3,4,4,4,4,5,2,2,14,7,10,6,5,3,3,3,3,3,4} //ط  
 {8,10,11,9,7,7,6,6,7,11,9,8,7,5,6,4,5,4,4,5,17,23,10,11,8,4,5,3,3,3,3,4} //ظ  
 {6,4,4,5,5,12,15,13,11,9,9,9,8,9,8,8,6,8,7,6,4,4,4,3,3,2} //ع  
 {1,2,3,1,1,2,2,4,3,4,4,5,5,8,7,10,8,9,9,11,12,13,3,3,4} //غ  
 {4,5,5,5,5,5,5,5,10,15,14,20,19,20,16,18,16,15,18,8} //ف  
 {3,3,5,9,16,15,13,11,13,18,19,21,18,16,13,3} //ق  
 {3,5,6,9,8,8,5,7,7,6,8,8,5,7,8,9,10,9,7,2,3} //ك  
 {3,3,3,11,13,16,26,21,9,0} //ل  
 {3,8,7,7,5,5,7,4,6,6,7,6,4,4,5,4,4,3,3,3,4,1} //م  
 {5,5,5,5,5,5,5,5,5,5,6,6,8,8,11,12,11,12,12} //ن  
 {5,5,4,4,5,10,10,8,9,13,13,12,15,16,12,10,9,10,8,7,3} //ه  
 {4,4,8,9,8,5,3,3,5,7,12,14,11} //ي  
 {4,4,4,4,4,9,10,7,6,8,8,11,13,15} //ذ  
  
 {0,3,6,13,24,16,6,4} //ل  
 {5,6,5,6,5,6,5,5,6,6,5,5,5,8,8,8,8,11,11,13,12} //ب  
 {4,5,5,7,8,8,7,7,6,7,12,14,14,17,10} //ب  
 {5,6,7,11,14,13,13,10,7,9,15,16,18,13} //ب

{1,2,4,4,5,6,7,7,8,9,8,7,7,7,8,10,10,11,11,7,7,7,6,5,4,4,4,4,3},//ح  
 {5,5,5,5,8,7,8,9,11,10,10,10,9,11,11,10,10,10,9,8,7,7,6},//ح  
 {1,1,4,5,5,6,8,8,9,10,9,9,8,11,12,12,11,10,8,7,7,6,5,4,4,3,3,3},//خ  
 {2,12,5,5,7,8,7,7,4,4,4,4,3,3},//ح  
 {1,13,3,4,4,4,5,7,5,5,5,3,2,4},//خ  
 {1,9,6,7,9,4,3,3,3,3,3,3,2,2,2,1,1,1,0,1,1},//ز  
 {2,7,6,7,9,5,4,8,7,6,4,4,3,3,2,2,2,1,1,1,0,1,0,0,1},//ز  
 {3,4,4,4,4,4,4,5,5,11,8,6,4,4,6,17,15,5,6},//س  
 {10,7,5,6,7,8,7,5,8,6,7,6,5,3,3,3,3,4,5,8,4,2,3,3,3,5},//ش  
 {5,4,5,5,5,5,6,6,6,6,7,7,8,8,8,6,7,6,6,6,7,7,8,8,8,8,9,10,11,12,11,10},//ص  
 {4,5,3,3,3,3,3,4,4,5,5,6,10,11,10,10,7,4,5,6,7,8,7,8,8,8,8,10,11,14,12,13,11,6},//ض  
 {5,5,5,5,5,5,6,11,14,20,25,20,8,8,7,8,8,8,8,8,10,10,10,11,12,12,14,13,12,7},//ط  
 {1,6,10,12,10,8,7,6,7,8,9,10,6,6,5,6,5,5,6,5,9,25,16,11,10,5,4,3,3,2,2,3,3},//ظ  
 {6,5,6,4,4,4,3,3,4,5,6,7,6,8,11,11,11,11,12,14,9,10,7,4,5,4,4,5,5,5,6,4,4,5,6,6},//ع  
 {4,3,3,3,4,9,9,12,15,18,18,13,12,12,13,12,9,5,4,4,4,4,4,4,5,4,3,4},//غ  
 {4,4,10,13,15,16,18,20,18,18,16,14,17,13,0},//غ  
 {3,4,4,3,2,2,4,7,11,14,9,12,14,15,17,18,15,5},//خ  
 {5,7,7,11,11,11,11,10,10,10,11,11,11,12,10,12,11,12,10,6,6,3,2},//ح  
 {5,5,5,5,7,12,17,22,26,23,12},//ط  
 {5,4,4,4,4,4,5,3,3,3,4,5,5,6,4,8,9,8,8,9,10,10,10,8,6},//ح  
 {5,5,5,5,4,4,6,9,9,11,11,11,14,6},//ح  
 {4,4,4,4,6,11,11,9,11,12,14,16,16,20,22,18,6,5,4,4,5,5},//ه  
 {1,14,12,12,11,10,10,11,11,5,3,4,3,2,2,2,1,1,1},//و  
 {5,5,7,9,10,7,5,7,8,11,15,19,15},//ز  
 {0,4,8,10,13,13,11,9,9,11,13,17,15,12,7},//س  
 {0,5,7,10,9,11,10,12,10,7,8,7,9,8,8,10,9,7,9,5,4,4},//س  
  
 {2,2,5,5,4,6,9,25,25,8,4,2},//ل  
 {12,16,12,10,5,4,4,4,4,4,4,4,4,4,4,6,8,9,8,7,5,4,4,4,4,4,4,5,5,5,4,6,8,11,5},//ب  
 {11,16,12,10,6,5,5,5,5,5,5,5,4,7,9,10,9,8,6,6,8,9,9,7,5,5,5,4,5,5,5,4,5,5,8,10},//ت  
 {3,15,11,10,7,4,4,4,4,4,4,4,4,6,7,9,8,8,6,6,12,13,13,11,7,4,4,4,4,5,4,5,4,4,5,10,4},//ث  
 {4,4,5,11,12,13,12,13,16,18,10,12,14,15,16,16,15,12,12,11,11,12,11,12,12,11,13,13,9},//ج  
 {1,4,5,10,10,11,12,13,14,17,13,11,12,12,13,12,12,12,13,12,12,14,13,12,9,11,16,9},//ح

{4,6,10,10,13,13,14,16,18,10,10,11,15,15,16,16,13,12,11,10,11,11,12,11,10,11,13,11,1} //تج

{12,12,10,9,9,10,10,9,5,4,5,5,4,4,5,3} //ـد

{12,13,8,8,8,9,10,8,8,9,8,5,4,5,3} //ـذ

{15,13,13,11,10,4,5,5,4,5,4,4,4,3,2,2,2,2,2,1,1} //ـر

{12,17,14,14,11,11,9,8,10,8,6,5,4,4,3,3,3,3,2,2,1,1,1,1,1} //ـز

{12,9,5,4,5,5,6,11,7,5,5,4,4,4,4,4,12,14,11,10,4,4,4,4,4,5,4,4,4,5,5,5,5,4,4,4,5,5,6,8,10,9} //سح

{11,11,8,6,9,9,10,12,14,8,12,15,12,10,5,4,5,9,18,12,10,5,4,4,5,4,4,5,5,4,5,4,4,5,5,5,4,5,5,4,6,7,8,10} //ش

{12,13,14,15,12,11,11,9,9,8,8,8,8,7,7,7,7,6,7,6,7,7,6,6,6,5,5,5,5,5,13,13,10,9,4,5,4,5,4,5,4,4,4,4,4,4,4,4,4,5,4,5,5,6,7,12,5} //ص

{11,13,15,14,10,10,10,9,9,9,8,8,7,8,7,7,6,6,8,10,11,11,8,6,5,5,5,4,4,4,4,6,14,13,10,8,4,4,5,4,5,4,4,5,5,5,5,5,5,5,4,4,5,5,6,8,9} //ض

{9,12,13,13,13,10,11,9,9,8,8,7,7,8,8,7,6,7,7,7,21,31,23,16,12,8,5,5,5,5,4,4,4,4,4} //ط

{9,12,13,14,11,10,9,8,10,12,12,11,10,8,7,7,7,7,7,24,30,21,15,12,8,6,5,4,4,4,4,4,4,4} //ظ

{1,6,7,8,8,9,9,10,14,18,19,19,17,18,17,16,12,11,9,9,8,11,11,12,5} //ع

{1,6,7,7,8,8,9,13,15,19,19,22,22,22,20,19,14,13,12,8,10,10,10,13,8} //غ

{15,17,13,13,14,15,15,20,20,15,7,5,5,5,5,5,5,5,5,5,5,5,5,4,4,4,4,4,4,4,4,4,4,5,4,5,4,5,5,8,7} //ف

{13,15,16,16,19,19,18,17,16,15,16,14,11,9,7,4,4,4,4,4,4,5,4,4,5,5,4,4,7,10,9} //ق

{9,20,26,22,16,9,4,4,4,4,11,13,11,11,9,8,11,9,7,5,5,5,5,4,5,5,4,4,4,7,8} //ك

{0,7,13,10,6,7,4,4,5,5,5,5,5,5,4,4,4,4,4,9,14,20,22,20,6,4} //ل

{6,7,9,10,12,11,9,9,8,8,8,6,6,10,12,13,6} //م

{16,15,11,7,4,5,5,5,5,4,4,6,9,10,9,7,5,5,5,5,4,4,5,5,6,7,11,3} //ن

{0,1,5,6,8,8,7,8,7,7,6,6,13,17,11,9,8,6} //هـ

{11,17,14,14,15,14,12,14,14,9,4,3,4,3,2,2,2,2,1,2,1,1} //و

{0,6,11,13,9,6,4,4,3,4,4,7,9,8,6,5,5,11,13,11,12,12,11,11,12,11,10,7,4,4,4,4,4,4,4} //ي

{0,5,4,12,10,12,9,7,7,11,9,15,20,11,8,7,6,3} //ـة

{0,7,10,10,7,6,5,5,5,5,5,5,5,5,6,10,11,12,12,13,13,13,10,12,11,9,4,4,4,4,4,4} //ى

{0,1,5,5,6,7,10,7} //،

{0,4,6,6,5,3,3} //،،

/\*

{4,5,4,4,8,10,12} //1س

{5,5,5,5,4,4,4,3,11,9} //2س

{0,10,11,6,5,6,4,4,5,3,4,6,4,4,4,4,4,4,4,4,4,3,5,5,8,11,12,13,5} //3س

\*/