CONTENT-BASED RETRIEVAL OF ARABIC HISTORICAL MANUSCRIPTS USING LATENT SEMANTIC INDEXING

BY

MOHAMMAD HUSNI NAJIB YAHIA

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

December, 2011

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Mohammad Husni Najib Yahia** under the direction of his thesis advisors and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Thesis Committee

Dr. Wasfi G. Al-Khatib
(Chairman)

Dr. Moataz Ahmed
(Member)

Dr. Husni Al-Muhtaseb
(Member)

Dr. Adel Ahmed
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

31/1/12
Date

Affectionately dedicated

To my mother:
for her love and prayers

To the memory of my father who passed on a love of reading and respect for education

To my beloved wife:
for her endless support

To my two children Osama and Omar:
for their love

And to my sisters and brothers:
for their encouragement

ACKNOWLEDGEMENT

In the name of Allah, the Beneficent, Most Merciful

All praise is due to Allah alone, who is the First and the Last, and He has full knowledge of everything.

May Allah's peace and blessings be unto His slave-servant Mohammed (peace be upon him) and unto all his brethren prophets and messengers of Allah, and unto all their followers to the Day of Judgment.

I wish to express my heartfelt thanks and obligation to my advisor, Dr. Wasfi Al-Khatib who guided me with keen interest and rendered all possible help inspite of his research, teaching schedules, department deputies, and family responsibilities. I found no suitable words to thank him.

Thanks also are due to my thesis committee members Dr. Moataz Ahmed and Dr. Husni Al-Muhtaseb for their invaluable comments and criticism.

I would like to thank all faculty members who guided and showed me the path of learning, research and knowledge throughout my graduate studies.

I also extend my thanks to my friends: Mohammed Al-Qadhi, Ashraf Al-Shaikh Khalil, Imad Jaha, Mohammed Amro, and Rashad Othman for their support.

I wish to express my thanks to Dr. Essam Al-khatib for providing me several historical Arabic manuscript images which I have used some of them in this research work.

Also, I am thankful for KFUPM for providing research facilities such as providing access to the high reputable journals.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Algorithms

ABSRACT

Large archives of historical Arabic manuscripts cannot be manually searched because of the difficulty of manual indexing construction and also cannot be automatically searched since they were stored in their original image forms. Optical character recognition (OCR) techniques are available, but due to the characteristics of the historical Arabic manuscripts and some contents such as figures and drawings of manuscripts, OCR techniques may not yield satisfactory results. Due to these reasons, we suggest in this research work, employing content-based image retrieval (CBIR) techniques and latent semantic indexing (LSI) approach to index these manuscripts and to make them better accessible to the public. Our system consists of five main modules: 1) image pre-processing, 2) image segmentation, 3) feature extraction, 4) LSI index construction, and 5) the query matching. Additionally, we present a novel feature called Circular Polar Grid (CPG). We have acquired two manuscripts, on which our system was tested. We evaluated the efficiency and accuracy of our developed system using precision and recall. The experiments were carried out, independently, on these manuscripts. The results showed that our system was able to retrieve relevant words. Additionally, the circular polar grid features showed superiority over other feature sets.

خلاصة الرسالة

الاسم: محمد حسني نجيب يحيى

عنوان الرسالة: استرجاع المحتوى القائم على الصور للمخطوطات العربية التاريخية باستخدام فهرسة الدلالات الكامنة

مجال التخصص: علوم الحاسب والمعلومات

تاريخ التخرج: ديسمبر، ٢٠١١

تعتبر عملية البحث اليدوي في المخطوطات العربية من الأمور الصعب تنفيذها وذلك نظرا لضخامة المخزون من هذه المخطوطات مما يؤدي لصعوبة انشاء الفهارس يدويا. كما ان البحث التلقائي لن يجدي نفعا نظرا لوجود الكثير من هذه المخطوطات مخزنة بالأساس كصور. وعلى الرغم من توفر تقنية التعرف الضوئي على الحرف (OCR) فإن بعضا من خصائص المخطوطات العربية التاريخية وبعض محتوياتها كالأرقام والرسومات تعيق تقنيات التعرف الضوئي على الحرف من أداء المهمة وبالتالي الحصول على نتائج مرضية. نتيجة لهذه الأسباب، فإننا نقدم في هذا العمل البحثي توظيف تقنيات المحتوى القائم على استرجاع الصور (CBIR) ومبادئ فهرسة الدلالات الكامنة (LSI) نهجا لفهرسة هذه المخطوطات وجعلها أفضل وصولا للعامة. يتكون النظام المطور من خمس وحدات رئيسية هي : ١) التجهيزات المسبقة، ٢) تجزئة الصور، ٣) استخراج السمات، ٤) بناء فهرس الدلالات الكامنة و ٥) تنفيذ و مطابقة الاستعلامات. بالإضافة إلى ذلك، قمنا في هذا العمل البحثي بتقديم سمة جديدة وهي الشبكة الدائرية القطبية Circular Polar Grids. أما فيما يتعلق باختبار النظام، قمنا بالحصول على مخطوطتين عربيتين ومن ثم تنفيذ الوحدات الرئيسية على تلك المخطوطتين. تم تقييم كفاءة ودقة النظام المطور باستخدام المقياسي: الدقة Precision والاسترجاع Recall. أظهرت النتائج أن نظامنا قادر على استرجاع الكلمات المرتبطة ذات الصلة. بالإضافة إلى ذلك، أظهرت السمة الجديدة، الشبكة الدائرية القطبية، تفوقها على مجموعات السمات الأخرى المستخدمة في النظام.

CHAPTER 1

INTRODUCTION

Generally, we define a manuscript as a small or large handwritten book which was written by the author himself or by human copiers before the printing era began (Gacek, 2009). Arabic historical manuscripts represent an important source of knowledge that needs to be preserved. Most of these manuscripts are archived and stored in scattered libraries and universities world-wide. Even the ones that were digitized and stored electronically cannot be automatically searched since they were stored in their original image forms. Due to the nature of these manuscripts, applying optical character recognition techniques to automatically recognize text is currently infeasible.. It is said that the number of Arabic historical manuscripts which were written in the golden age of the Islamic civilization topped 15 million. Due to several reasons such as wars and natural factors (e.g. humidity and pests), there was a great loss of this heritage. However, the remainder of these manuscripts does not exceed 3.25 million titles.

The topics of these manuscripts cover different sciences such as Islamic Jurisprudence, Linguistics, History, Geography, Sociology, Philosophy, Mathematics, and Astronomy. Preserving these manuscripts began earlier by keeping them in parlors or archives, storing them in utensils and repairing lacerated paper sheets. In contrast, with the

availability of current technologies such as digitization and image processing techniques, it is possible to digitize them and provide better and easier ways of access.

In this chapter, we present the motivation of this research work in Section 1.1. The concepts of content-based image retrieval are presented in Section 1.2. Problem definition, research objectives, thesis contributions are presented, respectively, in Section 1.3, Section 1.4, and Section 1.5

1.1. Motivation

Many Islamic researchers and literates prefer to deal with Arabic historical manuscripts in their original image forms. There are several reasons for this. Firstly, many of these manuscripts may contain marginal notes, figures, tables, and signs which cannot be recognized by traditional OCR techniques. Secondly, converting the manuscripts into text may lead to the loss of significant information. Thirdly, these manuscripts, in their original forms, represent a significant value as an Islamic heritage. Thus, these reasons encourage us to investigate techniques other than OCR in order to deal with these manuscripts without affecting their contents.

1.1.1. Content-Based Image Retrieval (CBIR) Concepts in Arabic Historical Manuscript Retrieval

In Arabic literature and books, including Arabic manuscripts, there are distinct words and phrases called keywords, which can be used to make a specialized search on a topic

or a specific paragraph. What distinguishes these keywords is that they are frequently repeated in the pages of a book and a particular topic may be searched or inferred through these keywords. For example, in the books of prophetic tradition (الشريف الحديث), a tradition text may be searched using few words, including the names of the narrators of the tradition such as Abu Huraira (أبو هريرة) , or the names of the collectors of these traditions such as Al-Bukhaari (البخاري) and Muslim (مسلم). Moreover, most of the texts of the traditions begin with the words: The Messenger of Allah peace be upon him said (قال رسول الله صلى الله عليه وسلم), and ends with the name of the narrator. Hence, the beginnings of the tradition texts and their endings can be searched. In the field of Arabic literature and poetry, specific verses by Al-Mutannabi (المتنبي) (an Arab poet) can be searched just by searching the word (المتنبي). Moreover, praise (المدح), satire (الهجاء), or pride (الفخر) poetry can be searched using expressions related to them. As for Arabic Grammar books, there are words such as verb (فعل), subject (فاعل), noun (اسم), and adjective (صفة) that show the function and the structure of the Arabic sentence. As for the books of jurisprudence and legal transactions, we find words such as duties (واجبات), provisions (الشروط), hated (مكروه), desirable (مستحب) which can be used to search terms and provisions of a particular topic. For example, some terms and conditions of prayer may be searched by searching the word prayer (الصلاة) or to be more specific by searching for words such as provisions of prayer (شروط الصلاة). On the other hand, ablution subject and its terms can be searched through the expression purity chapter (أحكام الطهارة). Figure 1, Figure 2 and Figure 3 show some manuscripts images and some keywords within them.

1.1.2. Why do we Need CBIR Techniques Instead of OCR Techniques?

Traditional Optical Character Recognition (OCR) techniques that usually recognize words character by character, fail when applied to Arabic historical manuscripts because of the manuscripts, themselves, and the Arabic script characteristics. The quality of historical manuscripts is often degraded due to faded ink, stained and lacerated paper, and other factors, Figure 4 depicts some of these factors as follows: A) Lacerated paper, B) Faded ink, C) Perforated paper, and D)Scribbles. On the other hand, the complex nature of Arabic handwritten scripts hampers OCR systems from showing encouraging results.

1.1.3. Why Arabic Historical Manuscript Script is Hard to Recognize?

Handwritten Arabic text recognition systems have low performance due to the characteristics of the Arabic language. Arabic language has 28 letters and more than half of them can assume 4 different shapes depending on their position (i.e. initial, medial, final, or isolated) within the word. For example, the letter "seen" (س) takes the following forms : (سـ) at the beginning of the word, (ـسـ) in the middle of the word, (ـس) at the end of the word and (س) as an isolated letter.

A word may consist of one or more subwords. This adds another difficulty to the recognition process. For example, the word (جامعة) which means university, consists of

two parts: (جا) and (معة). Some letters have the same bodies with different dot positions as in “ba” (ب), “ta” (ت) and “tha” (ث).

Combinations of some letters may form a new shape, which is a vertical stacking of two or more characters. For example, the word (لا) consists of the letter (ل) and the letter (ا), is very difficult to separate. Writers prolong letters for decorative reasons or for justifying text. These prolongings can cause vertical overlaps with neighboring letters. As shown in Figure 5, some of these difficulties are: A) Two different words have been written in such a way, i.e. by prolonging letters, as to be impossible to segment. B) and D) Overlapping of two letters. C) Combinations of some characters form new shapes which are vertical stackings of two letters. E) Different letters with the same shapes but using of dots related to letters will distinguish them. F) Two different shapes of which depend on the position of the letter within a word. G) a word consists of three subwords.

Figure 1: Some keywords within a grammar manuscript page

Figure 2: Some keywords within the holy Quran manuscript pages

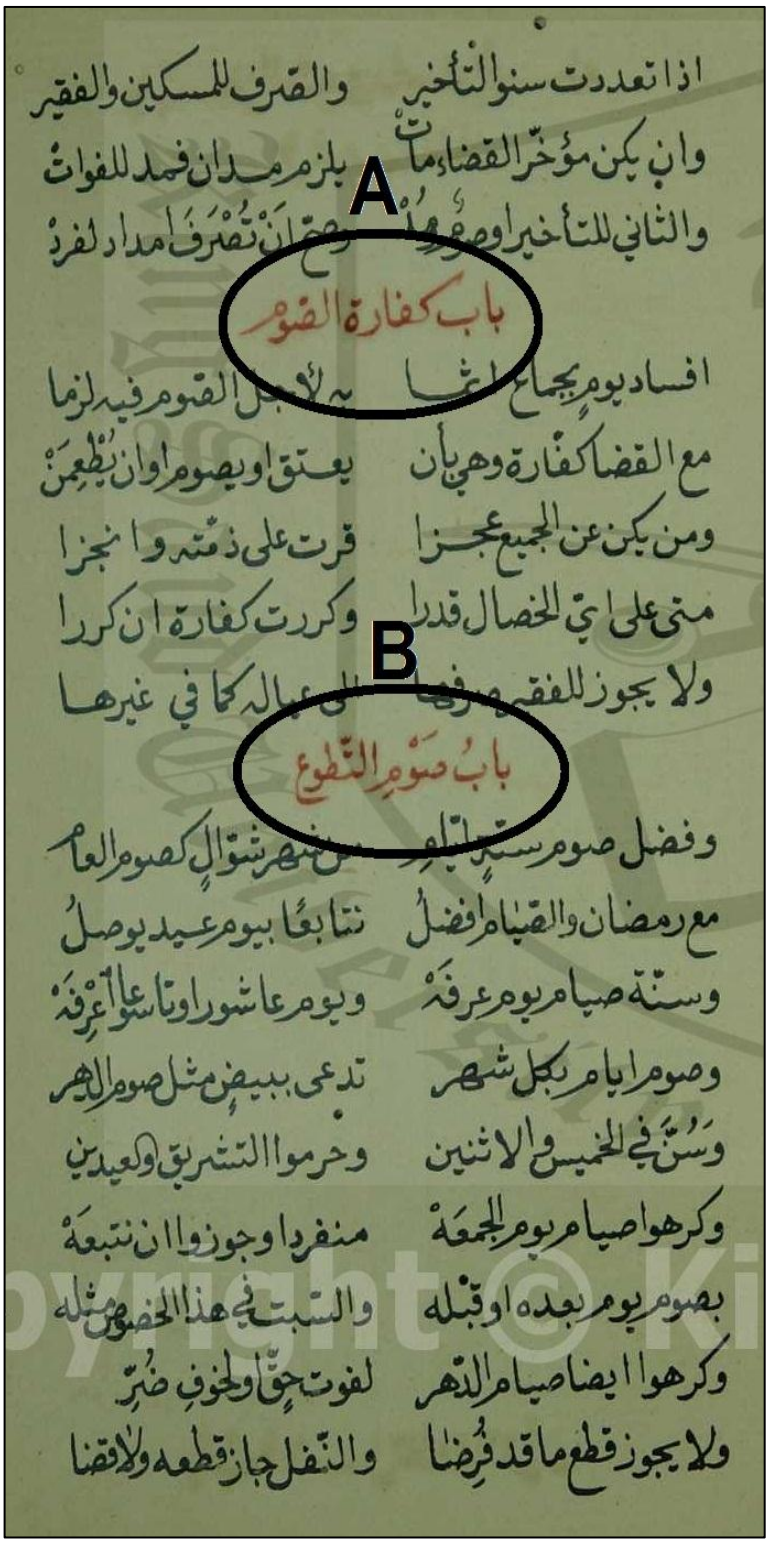

Figure 3: Some indexing terms as in A and B which are about fasting within Arabic literature manuscript pages

Figure 4: Some difficulties and challenges appearing in Arabic historical manuscripts. A) Lacerated paper, B) Faded ink, C) Perforated paper, and D) Scribbles

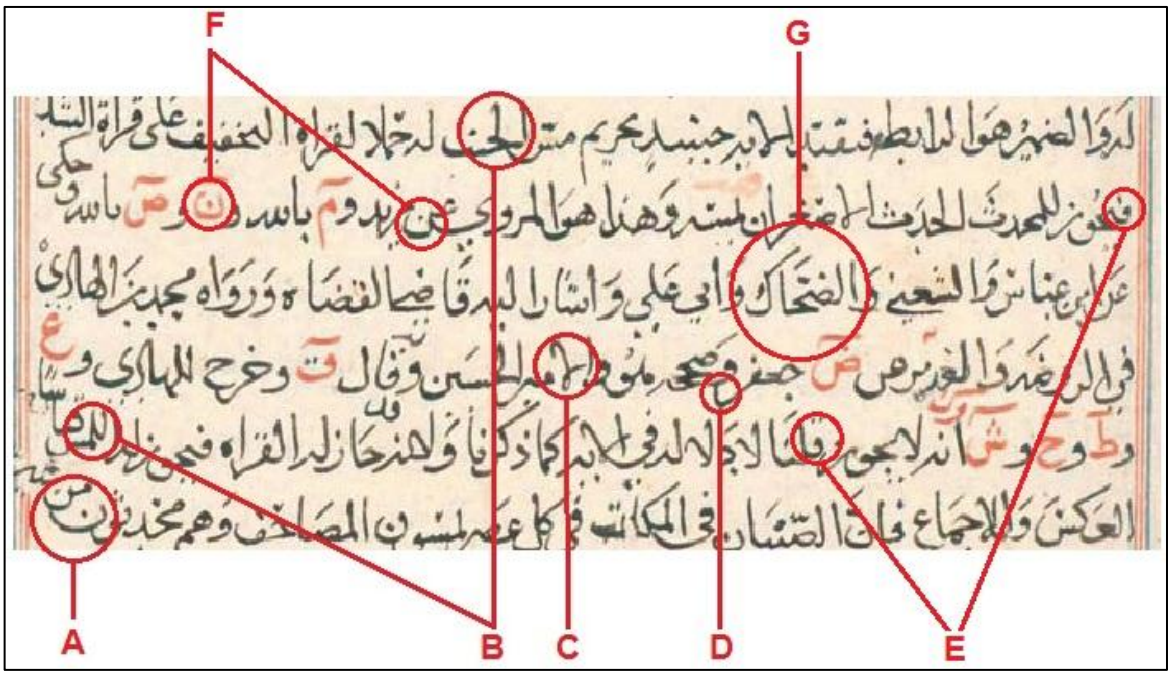

Figure 5: Difficulties and challenges in handwritten Arabic scripts. A) Prolonging letters B) and D) Overlapping letters. C) Vertical stackings letters. E) Different letters with the same shape. F) Different shapes of the same letter G) a word consists of subwords

1.2. Content-Based Image Retrieval (CBIR)

Because of the huge and fast growth of digital image databases, the development of efficient search and retrieval mechanisms has become more needed. Content-Based Image Retrieval (CBIR) has appeared and has been dedicated to tackle such difficulties (e.g. the quality of the original images). However, most image retrieval schemes have moved towards the CBIR technique (Li & Drew, 2003), without regard to accompanying textual information. This technique uses low-level features (e.g. color, shape and texture) to index and retrieve images from databases.

Generally, CBIR systems retrieve images from a database by comparing the features that were automatically extracted from the images themselves. Content-based image retrieval depends on the concepts of query by image example (QBE). It involves a matching process that matches the features of a query image with the features of images stored in the database. The system then determines the stored images whose feature values match those of the query most closely.

Furthermore, the structure of the content-based image retrieval system as shown in (Xiaohong & Jinhua., 2008) involves the following four parts in system realization: (1) Data collection, (2) Buildup of the feature database, (3) Search in the database and (4) Order arrangement and deal with the results of the retrieval. A wide range of possible applications for CBIR technology has been identified in (Gudivada & Raghavan, 1995) and (Eakins & Graham, 1999). These areas include:

- Home entertainment
- Crime prevention
- Geographical information and remote sensing systems
- Intellectual property
- Medical diagnosis
- The military
- Architectural and engineering design
- Fashion and interior design
- Journalism and advertising
- Cultural heritage

- Education and training
- Web searching

1.3. Problem Definition

Manual indexing of Arabic historical manuscripts is generally a very expensive task, and could raises human errors. While applying OCR techniques to automatically recognize text is currently infeasible, we will approach the problem of manuscripts indexing and retrieval by automatic indexing of Arabic historical manuscripts through word spotting, using "Text Image" similarity of keywords. This similarity will be computed using latent semantic indexing.

1.4. Research Objectives

The main objective of this research work is to develop a system using content-based image retrieval (CBIR) and Latent Semantic Indexing (LSI) techniques in order to support the automatic indexing of Arabic historical manuscripts. Such indexing will support the efficient querying and retrieval of manuscripts. In order to achieve this objective, the following steps have to be carried out:

- Preprocessing of the manuscripts pages.
- Segmentation and feature extraction.
- Indexing using LSI.

1.5. Thesis Contributions

The major contributions of this research work are:

- Applying the techniques of content-based image retrieval (CBIR) and the concepts of latent semantic indexing (LSI) to indexing historical Arabic manuscripts.
- Presenting circular polar grid features as a novel feature set.
- Developing a prototype system for retrieval and indexing of historical Arabic manuscripts based on the above contributions.

1.6. Thesis Outline

This thesis is organized as follows: Chapter 1 provides an introduction of the CBIR techniques and historical manuscripts indexing techniques. In chapter 2, we concentrate on literature survey about previous work related to manuscripts indexing. Additionally, feature sets and similarity measures are discussed. In chapter 3, we describe LSI mechanism and how to apply it on image retrieval. Chapter 4 describes the prototype system. Chapter 5 concentrates on the experimental evaluation while conclusions and future work are discussed in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

2.1. Manuscripts Indexing

In this section, we review the research works that deal with indexing historical manuscripts. We will concentrate, particularly in section 2.1, on historical Arabic manuscripts. While section 2.2 describes images features and similarity measures that used, generally, in image retrieval. The work of (Tomai, Zhang, & Govindaraju, 2002) has shown the difficulty of historical handwritten manuscript recognition.

Authors of (Al-Khatib, Shahab, & Mahmoud., 2007) proposed a computer aided retrieval and indexing system for Arabic historical manuscripts. The proposed system used different combinations of five feature categories: projection profiles, concentric circle features, angular line features, Hu's moment and geometric features. Several similarity measures such as Euclidean distance, Manhattan distance, and angular separation have been tested. The developed system has shown encouraging results with recognition rate equal to 76%.

In (Saykol, Sinop, Gdkbay, Ulusoy, & Cetin, 2004), a framework for content-based retrieval of historical documents in the Ottoman Empire archives was proposed. The

documents were stored as textual images, these images were compressed by constructing a library of symbols occurring in a document, and the symbols in the original image were then replaced with pointers into the codebook to obtain a compressed representation of the image. Features based on angular and distance span of shapes and features of wavelet domains were used to extract the symbols. Additionally, Euclidean distance and histogram intersection techniques were tested in order to evaluate similarities and differences

In (Aghbari & Brook., 2009), the classification of Arabic historical manuscripts was performed in several processes including segmentation process, features extraction process, and classification process. Several structural features (e.g. projection profile and upper and lower profiles) and statistical features were extracted. A neural network was used to learn and classify the input vectors into word classes. The authors showed that their technique was robust to different styles and the average accuracy was about 89%.

Authors of (Kane, Lehman, & Partridge., 2001) used a well-known word-image matching technique, called word spotting, for retrieval of George Washington's manuscripts documents. They used only geometric features (e.g. length, area, aspect ratio, number of ascenders & descenders, and number of black-to-white transitions). Sum of squared differences, Euclidean distance mapping, XOR algorithm, and Scott & Longuet Higgins were formulas used to test the developed system. The results of their

experiments showed that this technique (i.e. word spotting) performed better and was faster than competing matching with average precision equal to 73%.

Authors of (Rath & Manmatha., 2003) developed the word spotting technique which matched and grouped the words into clusters. Each cluster contained all instances of the same word. The words in a collection were matched as images by using dynamic time warping (DTW) which compared features of segmented word images. By pointing the required clusters, an index that points to the locations where words occur within the manuscript images can be built automatically. Features such as Gaussian smoothing and Gaussian derivatives were extracted from the input image. They claimed that DTW was much faster than the other techniques with average precision equal to 72%

For (Borbinha, Gil, & G. Pedrosa, 2006), the indexing process was "performed from a holistic perspective, by taking a whole word as a single recognizable symbol". Then, Image segmentation was executed using a development library. Words were compared based on their visual representation as a single unit. The data resulting from this process can be used to build valuable word indexes for the digitized pages. These indexes, kept in XML, can later be read by automated indexing tools. In order to carry out these tasks, the authors developed several tools such as ContentE, WIKI and SECO.

Table 1 summarizes some of these discussed literatures

Table 1: A summary of related work done in manuscripts indexing

Reference (Manuscripts type)	Feature Sets	Classifier/Formulas	Results	Strengths	Weakness
Al-Khatib, Shahab, & Mahmoud. 2007 (Arabic Manuscripts)	• Angular Lines • Concentric Circles • Projection Profiles • Hu's Moment • Geometric Features	• Euclidean Distance • Manhattan Distance • Angular Separation	76.5% (Recognition Rate)	• Concepts of "word book" • Relevance feed back • Different combination of feature sets and similarity measures • Classifiers are fast in their running and simple in their computations	• Classifiers are limited robustness to noise and variations in size • Number of query words is small (5) • Number of pages is small (8 Pages)
Saykol, Sinop, Gdkbay, Ulusoy, & Cetin. 2004 (Historical documents in the Ottoman Empire archives)	• Angular span • Distance span • Wavelet domain	• Euclidean Distance • Histogram Intersection Technique	86% (Average Accuracy)	• Allowing to process gray level images without a binarization process • Applicable to many other scripts	Classifiers are limited robustness to noise and variations in size
Aghbari & Brook. 2009 (Historical Arabic handwritten document)	• Structural features • Statistical features	Neural Network	89.3% (Average Accuracy)	• Improving the segmentation by "word stretching" • Relevance feedback	The NN is trained using a training dataset (200 words) which are selected manually (Supervised learning)
Kane, Lehman, & Partridge. 2001 (George Washington's manuscripts documents)	• Geometric features (length, area, aspect ratio, number of ascenders & descenders, number of black-to-white transitions)	• Sum of Squared Differences • Euclidean Distance Mapping • XOR algorithm • Scott & Longuet Higgins (SLH)	73% (Average Precision)	Most of the classifiers with the exception of SLH are fast in their running and simple in their computations	• Most of these classifiers are limited robustness to noise and variations in size • SLH requires more time processing.
Rath & Manmatha. 2003 (George Washington's manuscripts documents)	• Projection Profile • Word profile • Background/Ink Transitions	• Dynamic Time Wrapping (DTW)	72% (Average Precision)	DTW is much faster than other techniques	Using a manuscript of a good quality (65%)

2.2. Image Features and Similarity Measures

In the OCR process, features sets, similarity measures and relevance feedback are considered as main factors. According to (Ahmad & Grosky, 2003) , image feature can be described as physical features, logical features, local features and global features. There are many low-level features such as color, texture, and shape (Li & Drew, 2003), used in CBIR systems. Combining some percentage value of two features, namely color-texture features and color-shape features, or taking the union of these two features is proposed by (S. Zakariya, R. Ali, and N. Ahmad, 2010). Image retrieval based on dominant colors in the foreground image is proposed by (Krishnan, Banu, & Christiyana., 2007). The color histogram (Pecenovic, 1997) remains a popular primarily feature due to its simplicity. For (Shahab, Al-Khatib, & Mahmoud., 2006), matching on words is proposed by using different combinations of five features: angular lines features, concentric circle features, projection profiles, Hu's moment and geometric features. Segmented regions are used by (Zhuang & Wang, 2010), color layout and edge histogram are used by (Smeulders, Worring, S.Santini, Gupta, & Jain, 2000). Features such as texture descriptors and block correlation are used by (Pecenovic, 1997).

Particularly, several features are used in the literatures in the field of cursive handwriting recognition. Features such as ratio of height to width, position of the center of gravity of the image inside its bounding box, total number of black pixels, and average number of transitions between black and white pixels computed along different

directions are used by (Knerr, Augustin, Baret, & Price., 1998). While an overall description of the shape, curvature and histogram of the stroke direction are used by (Bunke, Roth, & Schukat-Talamazzini., 1995), (AlKhateeb, Ren, Jiang, & Al-Muhtaseb., 2011) extracted several structure-like features of Arabic scripts including a number of connected regions, a number of connected regions (dots) below the baseline, and a number of connected regions above the baseline.

Regarding to similarity measures, CBIR systems use these measures to quantify the similarity between a pair of images. Euclidean distance, Manhattan distance, and angular separation are used by (Shahab, Al-Khatib, & Mahmoud., 2006). Minkowski-form distance, cosine distance, histogram intersection, quadratic distance, Mahalanobis distance and χ^2 statistics are described and evaluated by (Zhang & Lu, 2003). Most of these distance measures are used to compute the difference between two histograms with the same number of bins. While Statistical classification methods which group images into semantically meaningful categories, using low-level features in order to achieve classification accuracy and to enhance retrieving, are applied by (Shahab, Al-Khatib, & Mahmoud., 2006) and (Zhu & Huang., 2003). Structural classification is used by (Vailaya, Figueiredo, Jain, & Zhang, 2001).

(Hare & Lewis, 2005) presented methods for using vector-space and latent semantic indexing retrieval models in combination with an invariant image representation based on local descriptors such as Lowe's SIFT of salient regions. They show that these descriptors can be quantized in visual terms and these terms are used as a basis for

indexing. For (Rahman, Desai, & Bhattacharya., 2006), an image retrieval framework with scalable image representation and inverted file-based indexing by incorporating automatically generated visual keywords is presented. “A codebook of visual keywords is implemented adopting a self-organizing map (SOM)-based vector quantization on the feature space of segmented image regions”. To reduce the dimensionality of the sparse feature vector, latent semantic indexing techniques are applied. A similarity matching function is proposed by exploiting the correlation between visual keywords. A prototype latent semantic indexing of image retrieval system is implemented by (Pecenovic, 1997). Features such as color histograms, texture descriptors and block correlation are extracted from images and represent terms in term-by-document matrix, while images themselves represent columns documents. Techniques of singular value decomposition (SVD) and term weighting also applied on this matrix.

The concepts of relevance feedback, which enhance the retrieval process in order to get better results by allowing a user to interact with the retrieval algorithm and by providing the information about which images are relevant to the query image, is applied by (Shahab, Al-Khatib, & Mahmoud., 2006), (Zhu & Huang., 2003), (Vailaya, Figueiredo, Jain, & Zhang, 2001) and (Pecenovic, 1997).

CHAPTER 3

LATENT SEMANTIC INDEXING AND IMAGE RETRIEVAL

The main reason for employing Latent Semantic Indexing (LSI) is its ability to carry out similarity matches using thresholds that are largely independent from the underlying features used. This chapter presents some mathematical background which is presented in section 3.1. Explanations about how LSI works in its original form in text retrieval and how to apply LSI to image retrieval are presented in section 3.2. Query matching and its mathematical background are presented in section 3.3.

3.1. Mathematical Background

Latent Semantic Indexing (LSI) model is used to exploit geometric relationships between document and term vectors in order to explain both similarities and differences. LSI attempts to project very high dimensional document and query vectors into a low dimensional space. This uses a *Singular Value Decomposition* (SVD) of the term-by-document matrix. Let us define SVD

Definition 3.1.1 *The singular value decomposition (SVD) of any matrix* $A^{m\times n}$ *of rank* $r \leq q = min(m,n)$*, denoted by SVD(A), is defined as:*

$$A = U\sum V^T \tag{3.1}$$

Where $U^{m\times q}$ and $V^{q\times n}$ are orthogonal matrices

The first r columns of U and V are called the left and right singular vectors respectively. The elements σ_i are the nonnegative square roots of the n eigenvalues of AA^T or AA^T.

In other words, U is $m \times m$ the orthogonal matrix whose columns define the left singular vectors of A, V is the $n \times n$ orthogonal matrix whose columns define the right singular vectors of A, and $\sum$ is the $m \times n$ diagonal matrix containing the singular values $\sigma_1 \geq \sigma_1 \geq \cdots \geq \sigma_{min(m,n)}$ of A in order along its diagonal. The components matrices of the matrix A are illustrated in Figure 6: Component matrices of the SVD

$m > n$

$$\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \bullet & 0 & 0 \\ 0 & \bullet & 0 \\ 0 & 0 & \bullet \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\sum} \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{V^T}$$

$n > m$

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \bullet & 0 & 0 & 0 & 0 \\ 0 & \bullet & 0 & 0 & 0 \\ 0 & 0 & \bullet & 0 & 0 \end{bmatrix}}_{\sum} \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{V^T}$$

Figure 6: Component matrices of the SVD

The following theorem is used to construct an approximation of A and estimate its quality.

Theorem 3.1.1 (Eckart and Young)

Let the SVD of A be given by Def. (3.1.1) with

$$\sigma_1 > \sigma_2 > \cdots > \sigma_{r+1} = \cdots \sigma_q = 0$$

And define the truncated SVD approximation A_k of A as

$$A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T = U_k \sum_k V_k^T, \qquad (3.2)$$

Then

$$\min_{rank(B)=k} \|A - B\|_F^2 = \|A - A_k\|_F^2 = \sum_{i=k+1}^{q} \sigma_i^2 \qquad (3.3)$$

Theorem 3.1.3 states that the best rank k approximation of A with respect to the Frobenius norm is A_k as defined by (3.2).

3.2. Constructing of the LSI Index

In text retrieval (TR), a document collection comprised of n documents which are indexed by m terms, can be represented as an $m \times n$ term-by-document matrix A, where n columns represent the n documents and m rows represent the m terms. Thus the matrix element could be the weighted frequency at which term i occurs in document j (Berry, Dumais, & O'Brien, 1995). Here, in our system, we use all subword

images and their feature values as a document collection. Thus, n columns represent the images and m rows represent the feature values.

3.2.1. LSI in Image Retrieval

In this section, we take an overview about how to apply LSI to image retrieval. Two challenges have emerged: 1) what are the documents? and 2) what are the terms? Here, we replaced the documents and the terms which are used in TR, respectively, by subword images and feature values. Thus, the columns of the term-by-document matrix will represent subword images and the rows represent feature values. Figure 7 demonstrates how a 8×3 term-by-document matrix is constructed from a small collection of subwords of the image of the word prophet (رسول). Angular line features and subword images were chosen as terms and columns respectively for indexing purposes. Another problem arose related to the feature value itself. In applying LSI to TR, the occurrence of the terms is easy to imagine and compute. In contrast, the notion of occurrence is even harder to imagine for image features which have numeric values, i.e. counter of black pixels within an area. To address this problem, we will apply the concept of feature occurrences. Finally, what is the suitable number of features that makes LSI work well? In order to tackle this challenge, we will test different combinations of feature sets in order to get satisfactory results.

$$A = \begin{pmatrix} 0.22 & 0.34 & 0.08 \\ 0.17 & 0.05 & 0.34 \\ 0 & 0.14 & 0 \\ 0 & 0.10 & 0 \\ 0.12 & 0.11 & 0.23 \\ 0.27 & 0.20 & 0.12 \\ 0.07 & 0 & 0.12 \\ 0.12 & 0.02 & 0.09 \end{pmatrix}$$

Figure 7: The construction of a term-by-document matrix A

3.2.2. Feature Normalizations

Feature normalization techniques aim at normalized extracted features in such a way that the normalized features are better suited for classification. For example, normalized features are suitable for a word written in different sizes. We use a simple technique to carry out the features normalization. Firstly, we count the number of black pixels within a given area. Secondly, this counter is divided by the count of all black pixels within the word image. We apply these steps for all regions within a given word image. Table 2 depicts these normalized features.

Table 2: Normalized features

Feature types	*f1*	*f2*	*f3*	*f4*
Features Counter	237	902	795	556
Normalized Features	0.0949	0.3609	0.3179	0.2234

3.2.3. Feature Occurrences

In text retrieval (TR), a document collection comprised of n documents which are indexed by m terms, can be represented as a $m \times n$ term-by-document matrix A. Where n columns represent the n documents and m rows represent the m terms. Thus the matrix element is a weighted frequency at which term i occurs in document j. Here, in our system, we use all word/subword images and their feature values as a document collection. Thus, n columns represent the images and m rows represent the feature values. The major problem of applying LSI to image retrieval is that terms must be replaced by image features (Pecenovic, 1997). In image feature values, there is no existence of occurrence, thus, we have to convert these feature values into feature occurrences. In order to do that, we apply the following algorithm (Pecenovic, 1997).

1. *Pick a reprehensive training set of document for each feature.*
2. *Compute the mean μ_f and the standard deviation σ_f if the features f's value across the training set.*
3. *For each image j in the collection define the "occurrence count" O_{fj} of feature f as*
4.

$$O_{fj} = \begin{cases} \left\lceil \frac{val_{fj} - \mu_f}{\sigma_f} \right\rceil & if\ val_{fj} > \mu_f \\ 0 & otherwise \end{cases} \quad (3.4)$$

Where val_{fj} is the value of the feature f in image j

Table 3 includes these feature occurrence values of the rectangular region features for a given word. While Figure 8 and Figure 9 illustrate this idea.

Table 3: Feature occurrences of the normalized rectangular region features

f1	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*
0	0	2	0	2	1	2	0	0

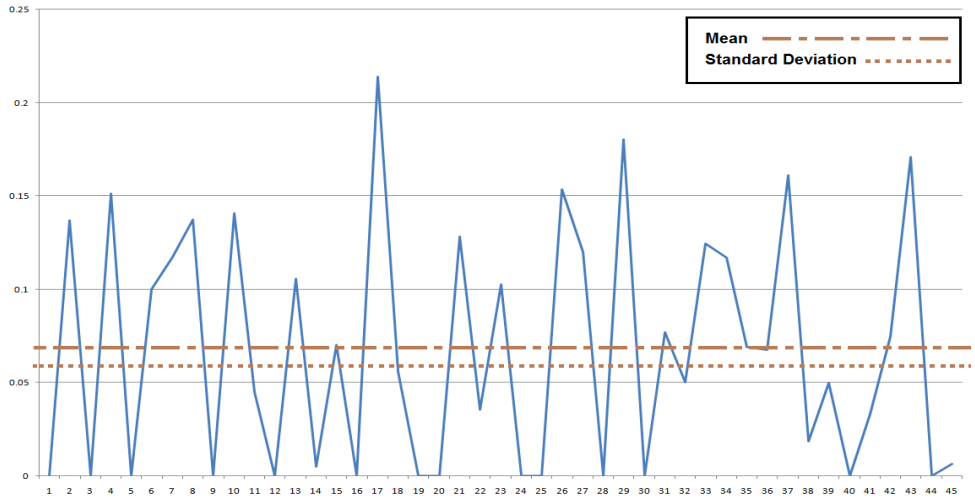

Figure 8: The histogram of the first 50 normalized values of the rectangular region features

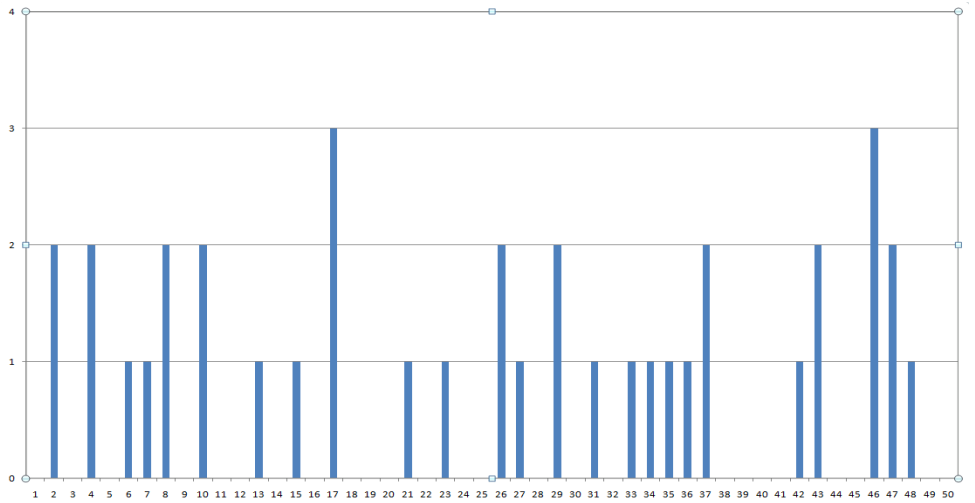

Figure 9: The result of applying (11.1) to the 50 normalized values of the rectangular region features

3.2.4. Term Weighting

The goal of term weighting is to condition the data in order to improve retrieval performance. What we mean by conditioning data is to describe the relative frequency (i.e. the importance) of a term i in a document j, which is called local weighting l_{ij}, and within the entire collection of documents which is called global weighting g_i. Recall that a collection of n documents indexed by m terms can be represented as $m \times n$ term-by-document matrix $A = [a_{ij}]$. Where each element a_{ij} is defined as:

$$a_{ij} = l_{ij} g_i \tag{3.5}$$

There are many local functions and global functions that can be used to compute term weighting. Many empirical studies report that the Log Entropy weighting work well with many data sets. Thus, each entry a_{ij} of A is computed as:

$$a_{ij} = g_i \log(f_{ij} + 1) \tag{3.6}$$

where f_{ij} is the frequency that term i appears in document j

While g_i can be computed as

$$g_i = 1 + \sum_j \frac{p_{ij} \log\ (p_{ij})}{\log n} \tag{3.7}$$

where

$$p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}} \tag{3.8}$$

3.3. Query Matching

Query matching in the vector space model can be viewed as a search in the column space of the matrix *A* or can be formulated using the component matrices of the SVD. Suppose we have a query vector q and n columns of the reduced-rank matrix A_k. $cos\,\theta$ can be used as a distance measure to retrieve relevant document. Computations of this measure are as follows:

$$\cos\theta_j = \frac{\left(A_k e_j\right)^T q}{\left\|A_k e_j\right\|_2 \|q\|_2} = \frac{\left(U_k \sum_k V_k^T e_j\right)^T q}{\left\|U_k \sum_k V_k^T e_j\right\|_2 \|q\|_2} = \frac{\left(U_k \sum_k V_k^T e_j\right)^T q}{\left\|U_k \sum_k V_k^T e_j\right\|_2 \|q\|_2} \tag{3.9}$$

For j=1,2,...,n

For the scaled document vector $s_j = \sum_k V_k^T e_j$, the formula in equation (3.9) can be simplified to

$$\cos\theta_j = \frac{s_j^T (U_k^T q)}{\left\|s_j\right\|_2 \|q\|_2}, \qquad for\ j = 1,2,\cdots,n. \tag{3.10}$$

CHAPTER 4

SYSTEM FOR INDEXING HISTORICAL ARABIC MANUSCRIPTS

The description of the developed system will be presented in this chapter. In this research work, we develop a system that provides retrieval and indexing facilities of Arabic historical manuscripts. In this system, a user can specify a query word and the system will search and retrieve occurrences of the query word found in the database. All of the retrieved words will be presented to the user to select the relevant words. Moreover, searching process will have several options (e.g. controlling threshold values, selecting feature types, and selecting feature set). The prototype system consists of five main modules: 1) image pre-processing module, 2) image segmentation module, 3) feature extraction module, 4) LSI index construction module, and 5) the query matching module. Each one of these modules consists of several tasks. Each one of these modules consists of several tasks. The architecture of the system is shown in Figure 10Figure 10. The following sections describe each model.

4.1. Preprocessing

In this stage, two necessary operations are carried out on manuscript images in order to represent them in a way which can be measured for enhancing the efficiency of the developed system. These operations include: i) Binarization and ii) Noise removal and

smoothing. These two processes are very important because the efficiency of later processes, such as segmentation and retrieval, depend on them. Other operations such as skew correction and curvature effect removal are not applied in the developed system.

4.1.1. Binarization

Most features used in the developed system depend, entirely, on the count of black pixels within an area. Thus, binary, i.e. black and white, images, as a source of features, was a suitable choice in our developed system. Many Arabic manuscripts that were stored as images are colored images. Converting these colored images into binary images, i.e. binarization, is done by two steps: 1) converting RGB color space to a gray-scale image, and 2) converting a gray-scale image to a binary image. These two steps are shown in Figure 11. The aim of these conversions is to simplify the amount of information in the colored images. Although a binary image contains less information than a colored image, the majority of important, feature related information is maintained, such as regions, black and white pixels, connected parts and so on.

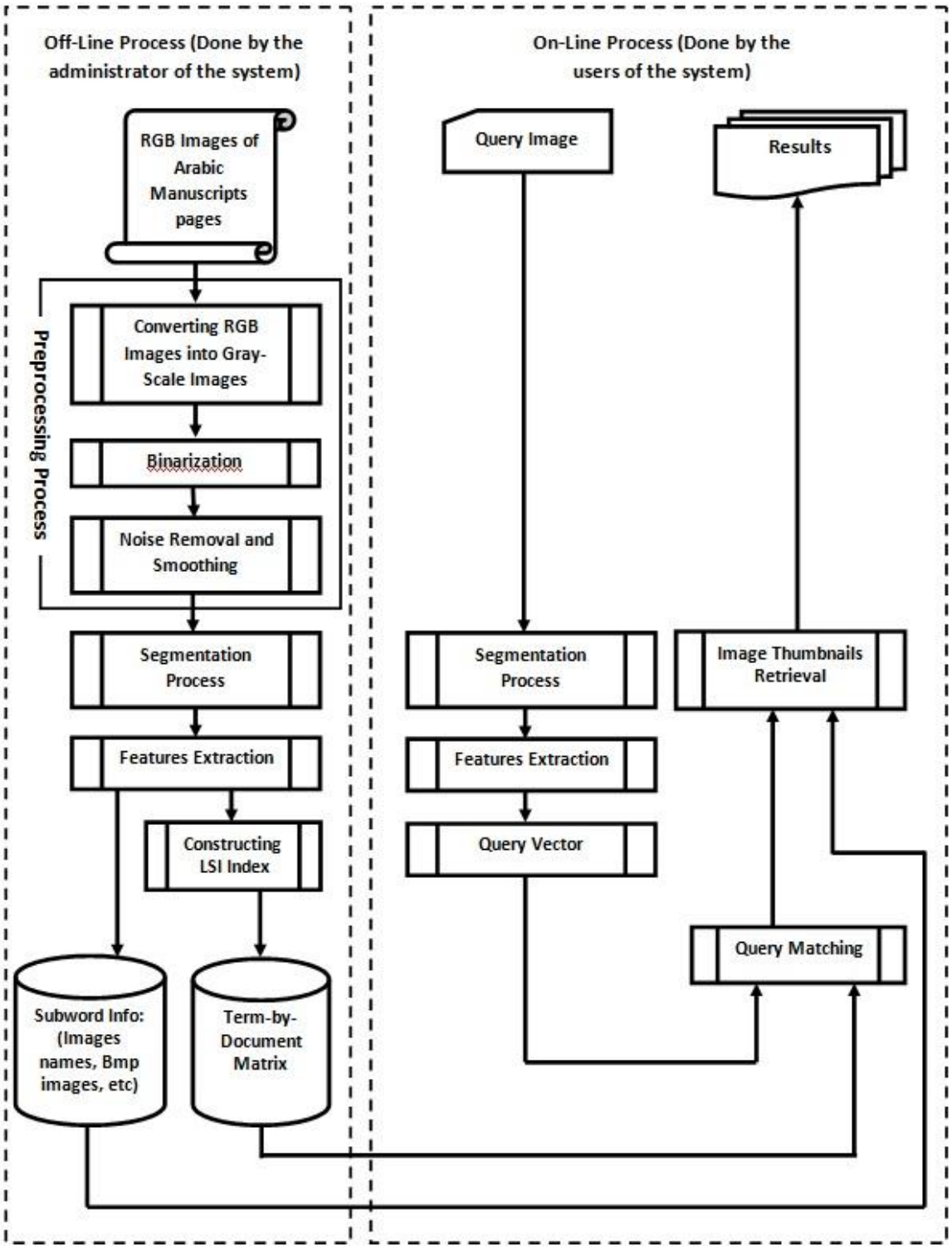

Figure 10: Architecture of the system

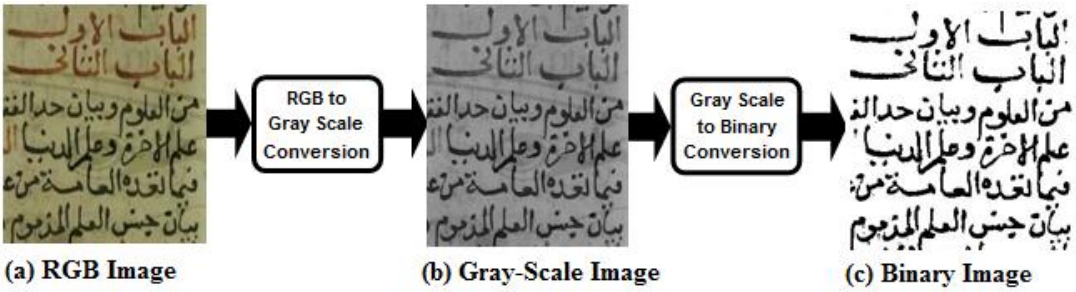

Figure 11: . Binarization steps

4.1.2. RGB to Gray-Scale Image Conversion

We can convert an image from RGB color space to a grey-scale image using a simple transformation by eliminating the hue and saturation information related to a pixel while retaining the luminance of a pixel. This conversion is implemented by using the following transformation :

$$I_{gray-scale}(n,m) = \alpha I_{color}(n,m,R) + \beta I_{color}(n,m,G) + \gamma I_{color}(n,m,B) \qquad (4.1)$$

Where $I_{gray-scale}$ is a gray-scale image, I_{color} is a RGB color image, (n,m) are the coordinates of an individual pixel within a gray-scale image or a color image. R, G, and B are color channels for red, green and blue colors respectively.

In other words, the gray-scale image is a weighted sum of the red, green and blue color channels. While $(\alpha = 0.2989, \beta = .5870 \; and \; \gamma = .1140)$ represents the weighting coefficients, the intensity level of the image pixels ranges from 0 to 255 where 0 value

represents black pixel and 255 represents white pixel. Figure 12 shows two images, the RGB image is shown in (a) and the resulted gray-scale image is shown in (b).

4.1.3. Gray-Scale to Binary Image Conversion

The binarization process replaces all pixels in the input image, i.e. gray-scale image, with luminance greater than an intensity threshold with the value 1 (white pixels) and replaces all other pixels with the value 0 (black pixel). The following equation can be applied in order to create a binary image from a gray-scale image.

$$I_{binary}(n,m) = \begin{cases} 1, & if\ I_{gray-scale}(n,m) > T \\ 0, & otherwise \end{cases} \quad (4.2)$$

Where T is the intensity threshold, $I_{gray-scale}$ is a gray-scale image, I_{binary} is a binary image and (n,m) are the coordinates of an individual pixel within a gray-scale image and a binary image.

Figure 13 depicts (a) a gray scale image and (b) a binary image. There are many methods used to compute or to choose the intensity threshold. In the very simplest method, the appropriate threshold can be chosen manually by applying the concepts of trial and error as shown in Figure 14, in this figure, (a) represents an image of an Arabic word lying on a dark background. (b) This Arabic word is successfully identified because of the threshold was chosen manually as in (c). This approach is not applicable for our developed system. In contrast, this threshold can be computed automatically

which is suitable for the developed system because many processing tasks require full automation. The common method to compute it automatically is proposed by (Otsu, 1979). This method selects the threshold which results in the tightest clustering of the two groups represented by the foreground and background pixels.

(a) (b)

Figure 12: RGB to gray scale conversion.

(a) (b)

Figure 13: Gray scale to binary conversion.

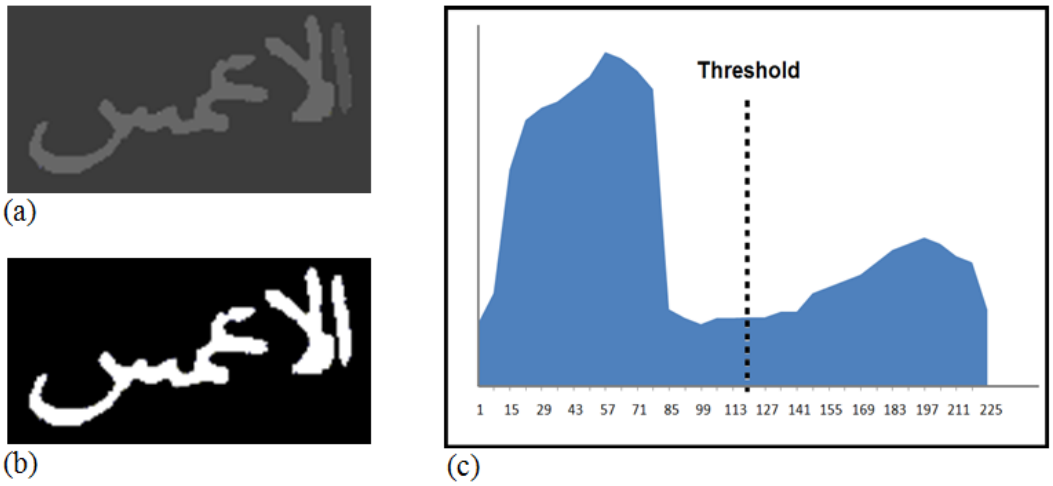

Figure 14: Computing threshold manually.

4.2. Smoothing and Noise Removal

The smoothing module is used for blurring and noise reduction. These two operations are used in removal of small objects which are not part of the text or are irrelevant areas of images. In order to remove these irrelevant pixels and fill the gap between connected components, we implement the smoothing algorithm proposed by Mahmoud (Mahmoud, 1994). This algorithm is based on a statistical decision criterion. Given a binary image of an Arabic subword, the algorithm modifies each pixel according to its initial value and to those of its neighbors. The rules are stated as follows.

$If\ P_0 = 0\ then$

$$\acute{P}_0 = \begin{cases} 0, & if \sum_{i=1}^{8} P_i < T \\ 1, & otherwise \end{cases} \quad (4.3)$$

$else$

$$\acute{P}_0 = \begin{cases} 1, & if\ p_i + P_{i+1} = 2\ for\ at\ least\ one\ i = 1,2,\dots,8 \\ 0, & otherwise \end{cases} \quad (4.4)$$

Where P_0 is the current pixel value, $\acute{P}_0$ the new pixel value and T is the threshold.

A threshold of 5 was found, experimentally, to yield acceptable results (Mahmoud, 1994). The current pixel and its neighbors are shown in Figure 15 while Figure 16 shows two images, a noise image is shown in (a) and a resulted image after applying smoothing and noise removal is shown in (b).

P_4	P_3	P_2
P_5	P_0	P_1
P_6	P_7	P_8

Figure 15: The current pixel P_0 and its neighbors

(a) (b)

Figure 16: Applying smoothing and noise removal on two images

4.3. Segmentation

The objective of this process is to segment all of the manuscript pages, i.e. images, into connected components (i.e. words/subwords). In order to achieve this objective, two modules are implemented. The former one is to segment the whole text within a page into lines. The later one is to segment these resulted lines into connected components.

4.3.1. Segment Manuscript Image to Line Images

To segment a manuscript image into line images, we depend on a well-defined method that enables us to detect a baseline of the Arabic scripts. Arabic baseline is very important in our developed system because it is used to estimate the location of peaks and to indicate the number of lines within a page in order to segment the whole page to lines. There are several methods, such as horizontal projection, word skeleton, and word counter representation, used in detecting baseline. We implement a method which is based on horizontal projection.

The horizontal projection method to detect the Arabic baseline works by reducing the 2D of data to 1D based on the pixels of the subword image, and the longest pink which will implement in the straight line will be the Arabic text baseline. Figure 17 visualizes the horizontal projection method for detecting the Arabic baseline. This figure contains: (a) an Arabic manuscript binary image. (b) Horizontal projection of this image. (c) The

baseline of the first line. We compute the horizontal projection profile by using the following equation :

$$P_i = \sum Img(i,j) \tag{4.5}$$

Where $P(i,j)$ *is the horizontal projection of the image for row* i*, and the* $Img(i,j)$ *is the pixel value at* (i,j)

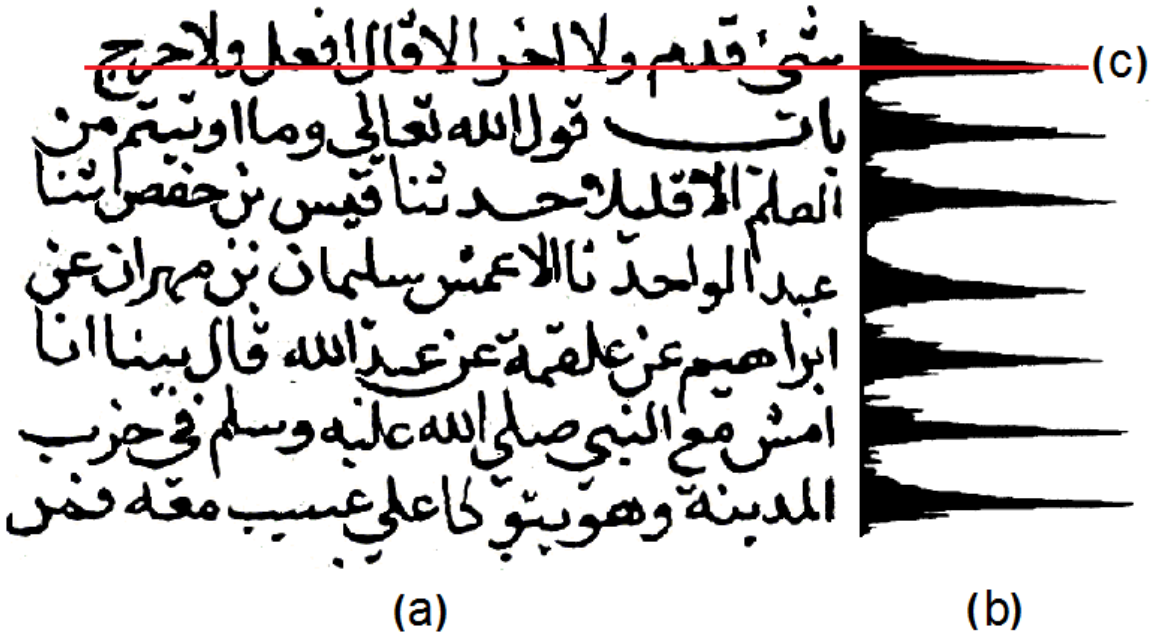

Figure 17: Detecting the baselines of the manuscript rows

4.3.2. Segment Line Images to Subword Images

In this module, a line image is segmented to connected-component images by applying the connected component algorithm (Solomn & Breckon, 2011). Additionally, each

subword is tagged with specific information such as page number and line number for storage and retrieval purposes.

4.3.2.1. Connected Components

Concepts of connected component labeling are used to detect connected regions, i.e. connected parts of the subword images, within a segmented line. A connected region consists of any group of connected pixels. For example, as shown in Figure 18, the Arabic word (الرسول), means prophet, has four connected parts c1, c2, c3, and c4. Two definitions of connection between pixels are commonly used which are 4-connection and 8-connection. Here, we are interested in 8-connection which is defined as: If a given foreground pixel has at least one neighboring foreground pixel to the north, south, east, west, north-east, north-west, south-east or south-west of itself, then this neighboring foreground pixel can be considered as part of the same object. These simple concepts are illustrated in Figure 19.

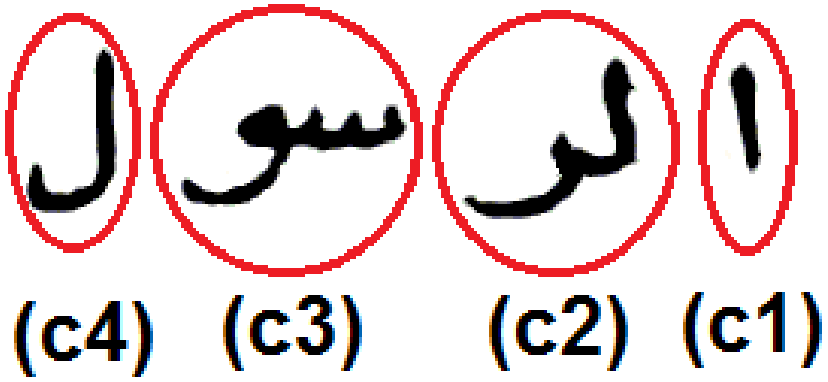

Figure 18: 4-connected components in the word prophet

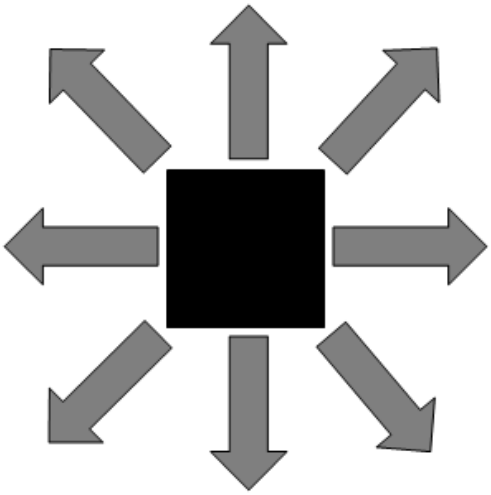

Figure 19: 8-connected pixels

4.3.2.2. Mathematics of Connected Components

(Woods, Gonzalez, & Eddins, 2004) defined mathematical representation of the concepts of connection components as follows:

A pixel p at coordinates (x, y) has two horizontal and two vertical neighbors, whose coordinates are $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ and $(x, y - 1)$. This set of 4-*neighbors* of p, denoted $N_4(p)$, is shown in Figure 20. The four diagonal neighbors of p have coordinates $(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1)$ and $(x - 1, y - 1)$. Figure 21 shows these neighbors, which are denoted $N_D(p)$. The union of $N_4(p)$ and $N_D(p)$ in Figure 22 is the 8-nieghbors of p, denoted $N_8(p)$. Two pixels p and q are said to be *8-adjacent* if $q \in N_8(p)$. Figure 23 shows a small region which is extracted from a binary line image.

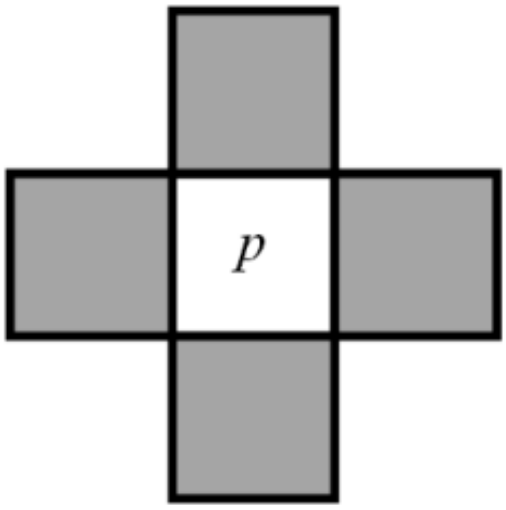

Figure 20: Pixel *p* and its 4-neighbors, $N_4(p)$

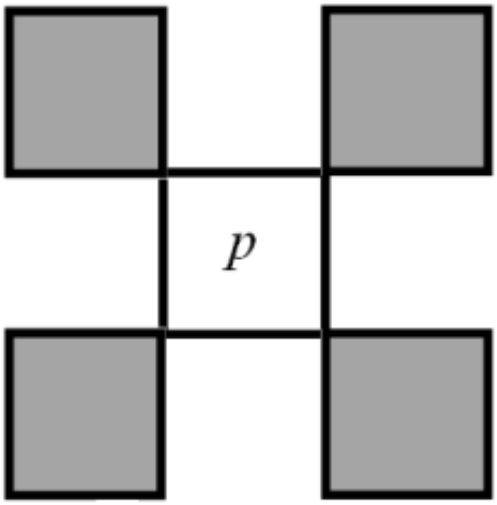

Figure 21: Pixel *p* and its diagonal neighbors, $N_D(p)$

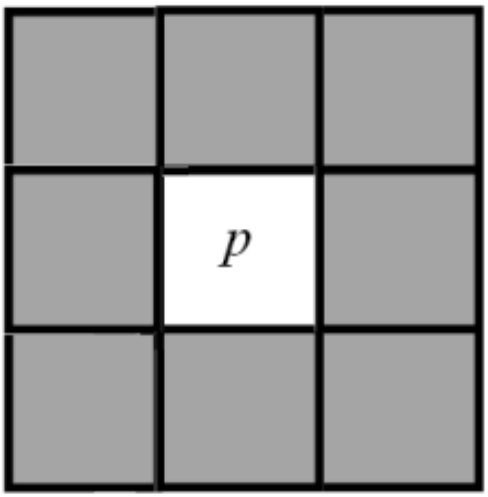

Figure 22: Pixel *p* and its 8-neighbors, $N_8(p)$

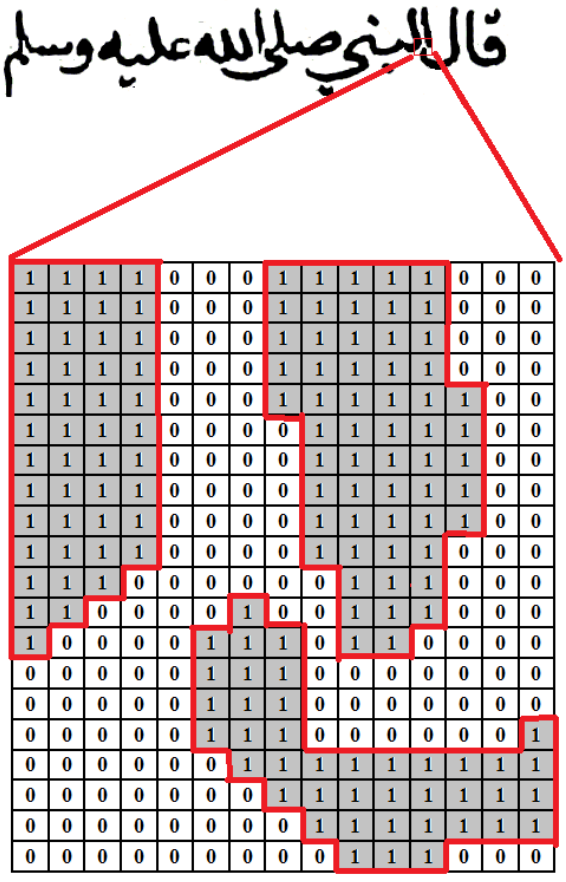

Figure 23: The shaded foreground pixels are 8-connected pixels

4.4. Features Extraction

The objective of this process is to extract the essential attributes which describe each word/subword resulted from the previous process i.e. the segmentation process. Generally, feature extraction is considered as a difficult problem in pattern recognition systems. This difficulty lies in what are the proper features to be extracted. Here, we use four sets of low-level features: 1) Concentric circle features, 2) angular line features, 3) rectangular region features, and 4) circular polar grid features. These features are represented as numerical values in order to construct LSI indexing and to carry out efficient searching and retrieval.

4.4.1. Concentric Circle Features

These features, which are four, are computed by counting the number of black pixels contained between concentric circles centered at centroid with radius $1 \times r$, $2 \times r$, …, $(n-1) \times r$, $n \times r$, where n is the number of concentric circles. Then, the values are normalized by dividing the count of black pixels within each region by the count of black pixels within the subword image. In practice, the following steps are carried out to compute these features:

Firstly, we compute centroid of the image I, $Ctr(xCtr, yCtr)$, using Algorithm 1

Algorithm 1: Computing the centroid of an Image

Input: Image *I*

Output: Centroid of an image $\boldsymbol{Ctr}(\boldsymbol{xCtr}, \boldsymbol{yCtr})$

Read Image I

for r←0 to I.allRows-1

  for c←0 to I.allCols-1

    if I(r,c)=BlackPixel

      xCtr.row ← *xctr.row* + *(I.allRows-1)* - *r*

      yCtr.col ← *yctr.col* + *c*

      area ← *area* + 1

    end if

  end for

end for

xCtr← *xCtr / area*

yCtr← *yCtr / area*

Ctr(xCtr,yCtr)

Secondly, the distance between the centroid and the original pixel $Org(0,0)$ is computed using the following equation:

$$Dis = \sqrt{xCtr^2 + yCtr^2} \tag{4.6}$$

Thirdly, the radius of the n concentric circles can be calculated using the following equation:

$$r_i = \frac{Dis}{n} \times i \tag{4.7}$$

Where $i = 1, 2, \ldots, n$ *and* n *is the number of concentric circles.*

Fourthly, to compute the count of black pixels $p(x, y)$ within a concentric circle the Algorithm 2 is designed for each black pixels

Algorithm 2: Computing the number of black pixels within concentric circles

Input: $Ctr(xCtr, yCtr)$, $r[1:n]$, n

Output: Count of black pixels, count[1:4], within concentric circles

count[1:4] ← 0

for each pixel p(x,y) in image I

  if $\sqrt{(x - xCtr)^2 + (y - yCtr)^2} < r_1$

    count(1) ← count(1) + 1

  end if

  for i←1 to n-1

    if $r_i \leq \sqrt{(x - xCtr)^2 + (y - yCtr)^2} < r_{i+1}$

      count(i+1) ← count(i+1) + 1

    end if

  end for

end for

Finally, the normalized values of the these features, i.e. concentric circle features are computed using the following equation:

$$NorConC = \frac{count_i}{\sum_i^n count_i} \tag{4.8}$$

Where $i = 1,2, \ldots n$ and n is the number of concentric circles

Figure 24 depicts these features while Table 4 includes these normalized features.

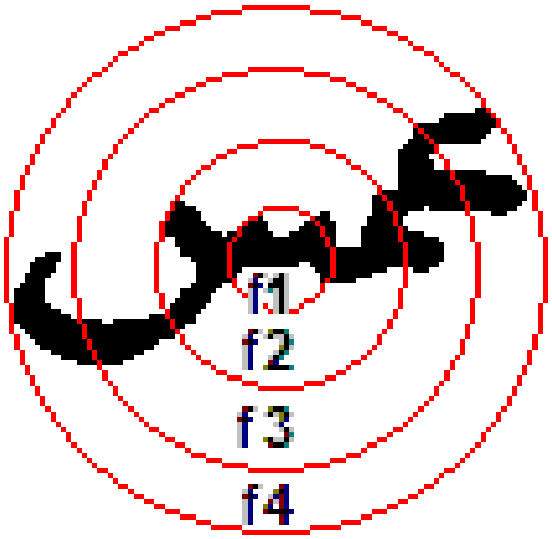

Figure 24: Four concentric circles

Table 4: Concentric feature values

f1	*f2*	*f3*	*f4*
0.0949	0.3609	0.3179	0.2234

4.4.2. Angular Line Features

We compute these features by counting the number of black pixels in 45° degree slices centered at the centroid of the subword image with respect to the horizontal axis. As a result of this slicing, there are eight features. Then, these values are normalized by the total number of black pixels of the image. Figure 25 depicts these angular line features of a given word, while Table 5 includes the eight feature values of that word.

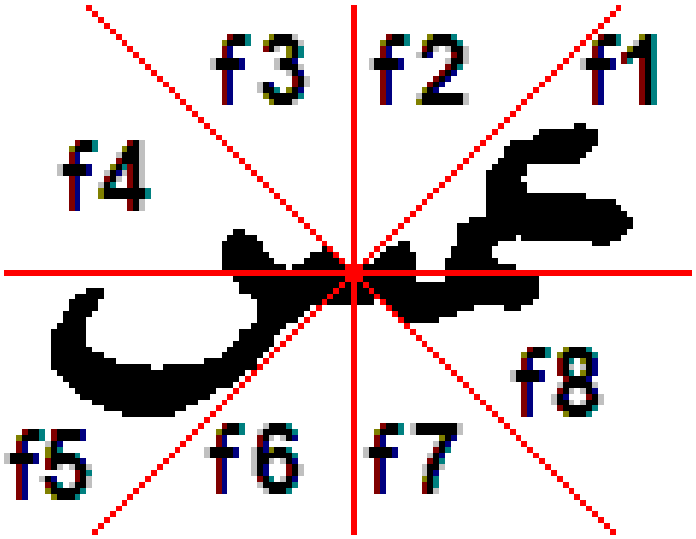

Figure 25: Eight features of angular lines

Table 5: Angular line feature values

f1	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*
0.36688	0.001843	0.00553	0.11225	0.344551	0.006452	0.00443	0.160012

The following steps are used to extract a feature set which belong to the region *f1*:

Firstly, we compute the centroid, $Ctr(xCtr, yCtr)$, of the image I using algorithm 1

Secondly, compute the slope of two lines, m, at $\theta = 0^{\circ}, 45^{\circ}$, respectively, and centered at centroid, $Ctr(x, y)$, using the following equation:

$$m = \tan(\theta) \tag{4.9}$$

Thirdly, the following equation is used for a given line:

$$y = mx + b \tag{4.10}$$

Thus, there are two equations resulted from plugging the two different slopes m_1 and m_2 in the above equation:

$$y_1 = m_1 x + b_1 \tag{4.11}$$

$$y_2 = m_2 x + b_2 \tag{4.12}$$

Fourthly, because of each line is passing through the centroid $(xCtr, yCtr)$, we can compute Y-intercept i.e. b using the following equation:

$$b_i = \frac{xCtr - yCtr}{m_i} \tag{4.13}$$

Where i is the line number.

Finally, we can find y_1 and y_1 by plugging the X-coordinate of each black pixel in the equation of lines defined in (4.11). If $y_1 < y \leq y_2$ then, this pixel belongs to the region 1. Practically, we compute the count of black pixels within the two angular lines with slope 0 and slope 1 by applying the developed Algorithm 3

Algorithm 3: Computing the number of black pixels within the two angular lines.

Input: $Image\ I, Ctr(xCtr, yCtr)$

Output: Count of black pixels

$m_1 \leftarrow \tan(0)$

$m_2 \leftarrow \tan(45)$

$$b_1 = \frac{xCtr - yCtr}{m_1}$$

$$b_2 = \frac{xCtr - yCtr}{m_2}$$

for each black pixel $p(x, y)$ in image I

$y_1 = m_1 x + b_1$

$y_2 = m_2 x + b_2$

if $y_1 < y \leq y_2$

count(1) ← count(1) + 1

end if

end for

4.4.3. Rectangular Region Features

These are nine feature values resulted from dividing the image area into nine rectangular regions. Then, the count of black pixels is computed for each region. These values are normalized by the count of all black pixels of the image. Practically, we applied the following steps in order to calculate these feature values:

Firstly, we compute the dimensions i.e. the height and the width for each rectangular areas:

$$RecRegW = \frac{Img.Width}{3} \tag{4.15}$$

$$RecRegH = \frac{Img.Height}{3} \tag{4.16}$$

Secondly, the sum of black pixels within each area is calculated using the following equation:

$$RecRegCnt_i = \sum_i p(x, y) \tag{4.17}$$

Where $i = 1,2, \dots, n$, $n = 9$ is the number of rectangular regions and $p(x, y)$ is a black pixel.

Thirdly, these feature values are normalized by dividing each rectangular region counter by the subword total black pixel count. Figure 26 depicts these features while Table 6 shows the normalized values. For instance, to find the count of black pixels contained in the second rectangular region, the following algorithm is applied:

Algorithm 4: Computing the number of black pixels within rectangular regions

Input: *image I*

Output: Count of black pixels, count[1:4], within concentric circles

$\text{RecRegW} \leftarrow \frac{\text{Img. Width}}{3}$

$\text{RecRegH} \leftarrow \frac{\text{Img. Height}}{3}$

count[1:4] ← 0

for each black pixel *p(x,y)* in image I

  if $1 \leq x < \text{RecRegH}$ **and** $\text{RecRegW} + 1 \leq y < \text{RecRegW} \times 2$

    count(2)=count(2)+1

  end if

end for

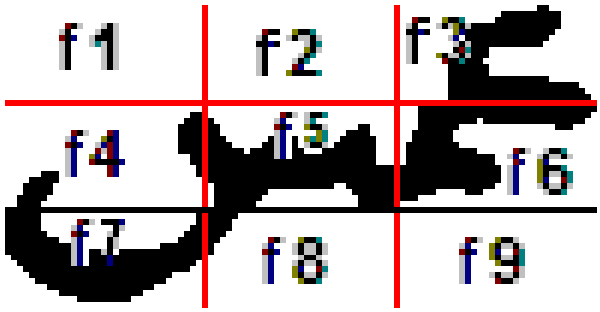

Figure 26: Nine rectangular regions

Table 6: Rectangular region feature values

f1	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*
0	0	0.2005	0.0510	0.2682	0.2119	0.2392	0.0290	0

4.4.4. Circular Polar Grid Features

To the best of our knowledge, we are the first to investigate circular polar grids as features used in image retrieval. These features are thirty two values which are calculated by drawing four angular lines with equally-spaced angles i.e. 0°, 45°, 90° and 135° with respect to the horizontal axis (Sec. 4.4.2) and four concentric circles, with radius $1 \times r$, $2 \times r$, …, $n \times r$, where n is the number of concentric circles (Sec. 4.4.1), centered at the image centroid. Then the count of black pixels within each region resulted from the intersection of the drawn angular lines and the drawn concentric circles is calculated. Finally, the normalized values of these features are computed by dividing each region's value counter by the total number of black pixels in the subword. In practice, the following algorithm is applied to calculate a total sum of black pixels within a region resulted from the intersection between the first and the second concentric circles and the two lines with slopes 0 and 1. Figure 27 depicts these thirty two features while the normalized values of these features are showed in Table 7.

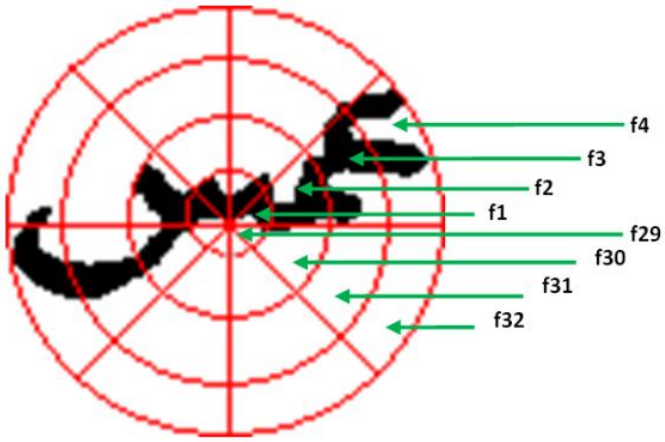

Figure 27: The thirty two features of circular polar grids

Table 7: Some circular polar grid feature values

f1	*f2*	*f3*	*f4*	*f13*	*f14*	*f16*	*f31*	*f32*
0.0754	0.1271	0.1356	0.1524	0.0823	0.1304	0.0687	0	0

Algorithm 5: Computing the number of black pixels within the first sector of circular polar grids

Input: Image I, $Ctr(xCtr, yCtr)$, and $r[1:2]$

Output: Count of black pixels

$m_1 \leftarrow \tan(0)$

$m_2 \leftarrow \tan(45)$

$$b_1 = \frac{xCtr - yCtr}{m_1}$$

$$b_2 = \frac{xCtr - yCtr}{m_2}$$

for each black pixel $p(x, y)$ in image I

$y_1 = m_1 x + b_1$

$y_2 = m_2 x + b_2$

if $y_1 < y \leq y_2$

if $r_2 \leq \sqrt{(x - xCtr)^2 + (y - yCtr)^2} < r_2$

count(1) ← count(1) + 1

end if

end if

end for

4.4.5. Other Information

In order to support several processes such as the query matching process in the developed system, we extract or create pieces of information about each subword/word. This information includes:

- The width and the height of the subword image.
- The centroid
- The location of the subword within the line, the page, the manuscript.

- BMP image

Some of these features, e.g. centroids and dimensions, are used in the feature extraction process. Information such as dimensions, locations and BMP images is used for advanced purpose such as clustering and displaying retrieved images as thumbnails.

CHAPTER 5

EXPERIMENTAL EVALUATION

This chapter presents the testing results of the experimental work of this research work. Furthermore, the results of comparisons with other work are evaluated and presented here.

5.1. System Implementation

Our system is implemented in the MATLAB 2009b environment. It is a GUI based easy to use application. Users can perform several tasks such as preprocessing steps, segmentation and feature extraction steps, and searching steps. Furthermore, Microsoft Excel is used for storage purposes.

5.2. Historical Arabic Manuscripts

Within the experimental work of this thesis, two pre-scanned historical Arabic manuscripts were used. The title of the first one is "Sahih Al-Bukhari" (صحيح البخاري). We used 34 pages from the section of "Mawaqeet Al-Haj wa Al-Umra" (مواقيت الحج والعمرة) of this manuscript. These pages were pre-processed, segmented into lines and words (i.e. connected component), and then the word features were extracted. The

overall number of resulted words is about 5500. The second manuscript is an old Arabic handwritten manuscript which was used in (Shahab, Al-Khatib, & Mahmoud., 2006). This manuscript was used for comparison purposes.

5.3. Query Words

Thoughtfully, we choose 20 words as query words. Some of these words such as (الحج), meaning pilgrimage, may be considered as keywords in order to search particular topics. Names of places and names of people are also selected to be query words. Moreover, phrases such as "peace be upon him" (صلى الله عليه) and "be pleased with him" (رضي الله عنه) are selected to issue queries in order to search prophetic traditions. On the other hand, all of these words consist of one-connected components, two-connected components, three-connected components or four-connected components.

5.4. Performance Evaluation

The standard way to evaluate the performance of a system is to compute recall and precision. Recall is measured as the ratio of the number of relevant documents retrieved to the total number of relevant documents which exist in the collection. In other words, recall is a measure of the accuracy of the retrieval. In contrast, the precision is measured as the ratio of the number of relevant documents retrieved to the total number of documents retrieved. This is a measure of the completeness of the retrieval. Computations of these measures are as follows:

$$Recall = \frac{Relevant\ Images\ Retrieved}{Total\ Relevant\ Images\ Present} \quad (5.1)$$

$$Precision = \frac{Relevant\ Images\ Retrieved}{Total\ Images\ Retrieved} \quad (5.2)$$

5.4.1. Feature Sets

As mentioned before, four different features are used in the experimental work within this thesis. Each one of these features has three different calculated values. The first value is the normalized features, the second one is feature occurrence values, and the third value is term weighting features. Also, we group our query words depending on the number of connected components. Thus we have 12 feature sets that should be tested on each query word. The goal of the tests is to indicate which feature sets and feature value types will be most useful in indexing and retrieval of manuscripts. The main results from these tests are shown in Table 8, while Figure 28, Figure 30, Figure 32 and Figure 34 show some of these tests using circular polar grid features. In these figures, the cells marked with true symbol correspond to relevant words, while the cells marked with false symbol correspond to non-relevant words. The discarded words are shown in Figure 29, Figure 31, Figure 33 and Figure 35. Some of the reasons that the system discards retrieval of these words are: i) Some of these words are concatenating with other words and ii) the prolonging of some letters for decorative reasons. While Figure 36 and Figure 37 visualize the experiment results for some selected query words

using all feature sets that used in this research work. It is clearly that circular polar grid feature sets are the best features over the other feature sets.

Table 8: Recall results of different feature sets

One-Connected Components			
Feature Sets	Normalized Features	Feature Occurrences	Term Weighting
Concentric Circles	47.5%	56.5%	56.2%
Angular Lines	53.7%	60%	60%
Rectangular Regions	53.7%	60%	56.7%
Circular Polar Grids	58.7%	61.2%	63.7%
Two-Connected Components			
Feature Sets	Normalized Features	Feature Occurrences	Term Weighting
Concentric Circles	55%	57.5%	63.7%
Angular Lines	58.7%	62.5%	65%
Rectangular Regions	60%	62.5%	66.2%
Circular Polar Grids	65%	70%	70%
Three-Connected Components			
Feature Sets	Normalized Features	Feature Occurrences	Term Weighting
Concentric Circles	61.2%	60%	62.2%
Angular Lines	62.5%	70%	75%
Rectangular Regions	65%	67.5%	73.7%
Circular Polar Grids	68.7%	73.6%	77.5%
Four-Connected Components			
Feature Sets	Normalized Features	Feature Occurrences	Term Weighting
Concentric Circles	61.2%	61.2%	63.7%
Angular Lines	65%	71.2%	75%
Rectangular Regions	63.7%	72.5%	75%
Circular Polar Grids	70%	77.5%	78.7

The main evident observations from Table 8 are:

- Term weighting of circular polar grid feature set, with recall equal to 78.8%, performs better than other feature sets. While normalized concentric circle feature set is the worst with recall equal to 47.5%.
- There is a steadily increasing relationship between the number of connected components and the recall.

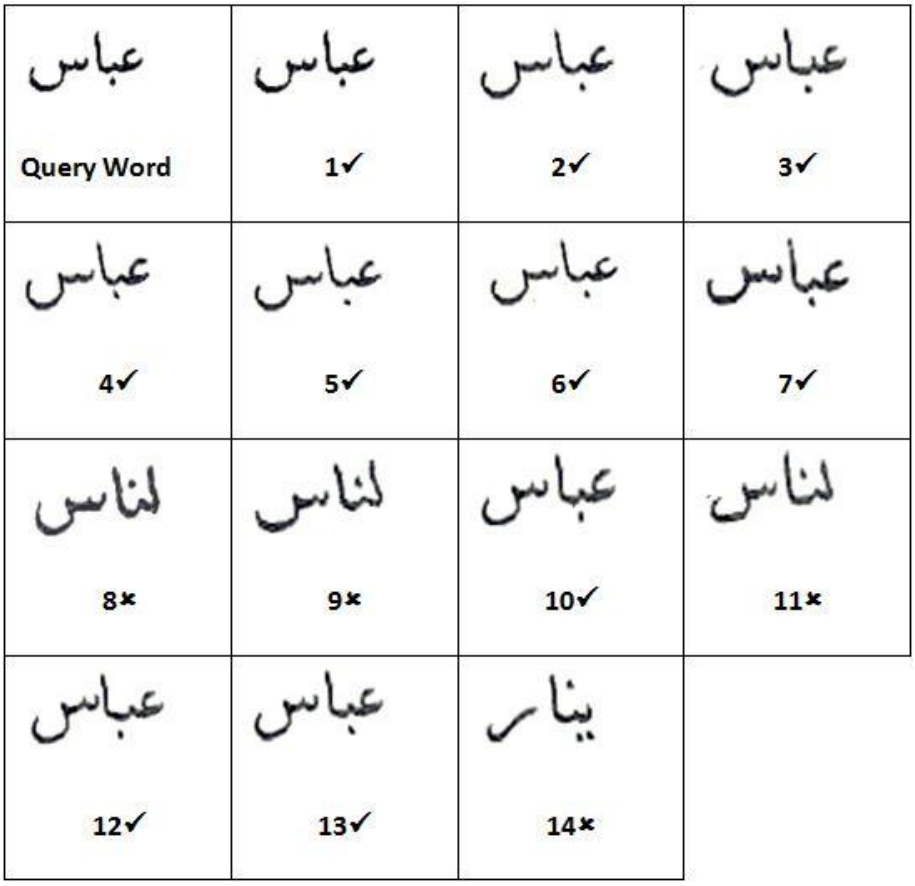

Figure 28: Experiment results of the query word Abbas (عباس) using circular polar grid features

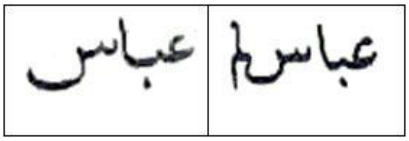

Figure 29: Discarded words of the query word Abbas (عباس)

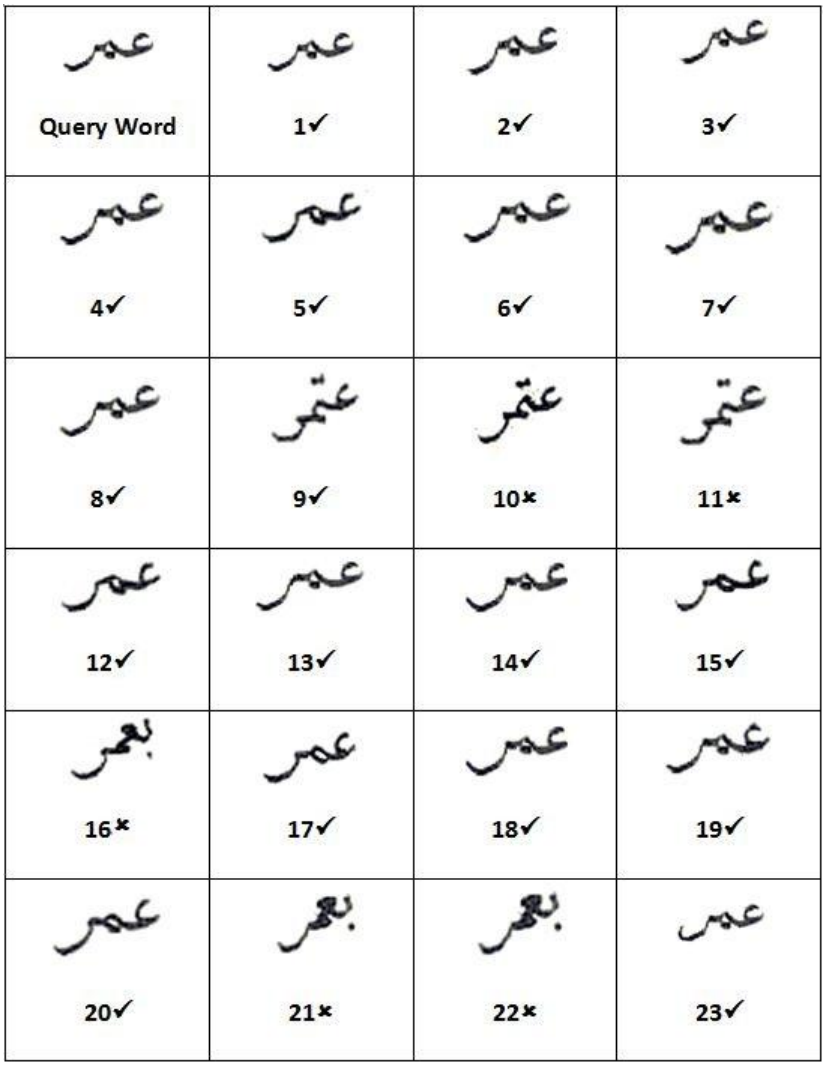

Figure 30: Experiment results of the query word Omar (عمر) using circular polar grid features

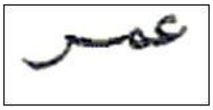

Figure 31: A discarded word of the query word Omar (عمر)

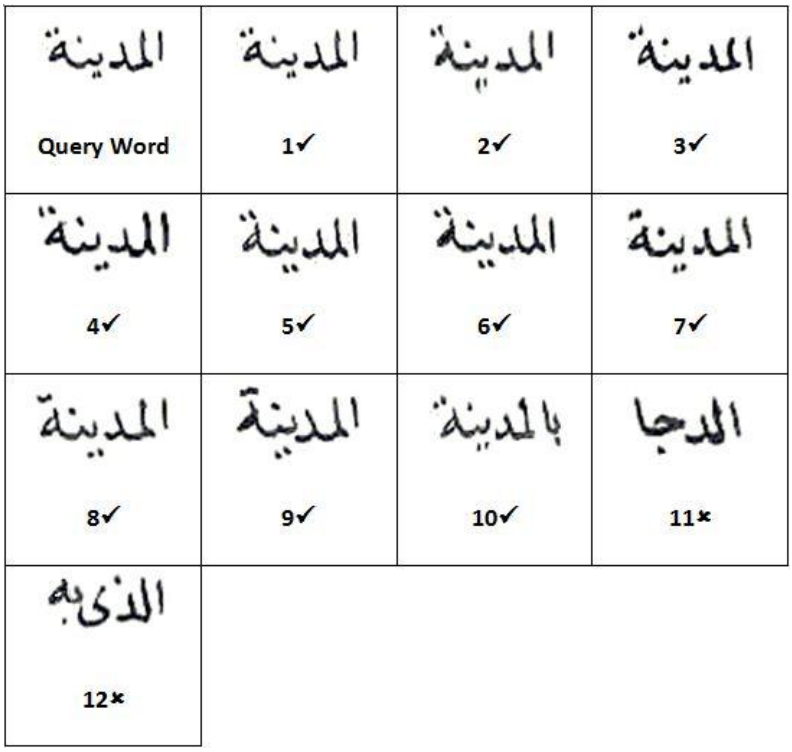

Figure 32: Experiment results of the query word Al-Madinah (المدينة) using circular polar grid features

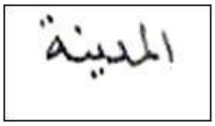

Figure 33: A discarded word of the query word Al-Madinah (المدينة)

Figure 34: Experiment results of the query words "Peace be upon him" (صلى الله عليه) using circular polar grid features

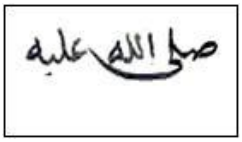

Figure 35: Discarded words of the query words "Peace be upon him" (صلى الله عليه)

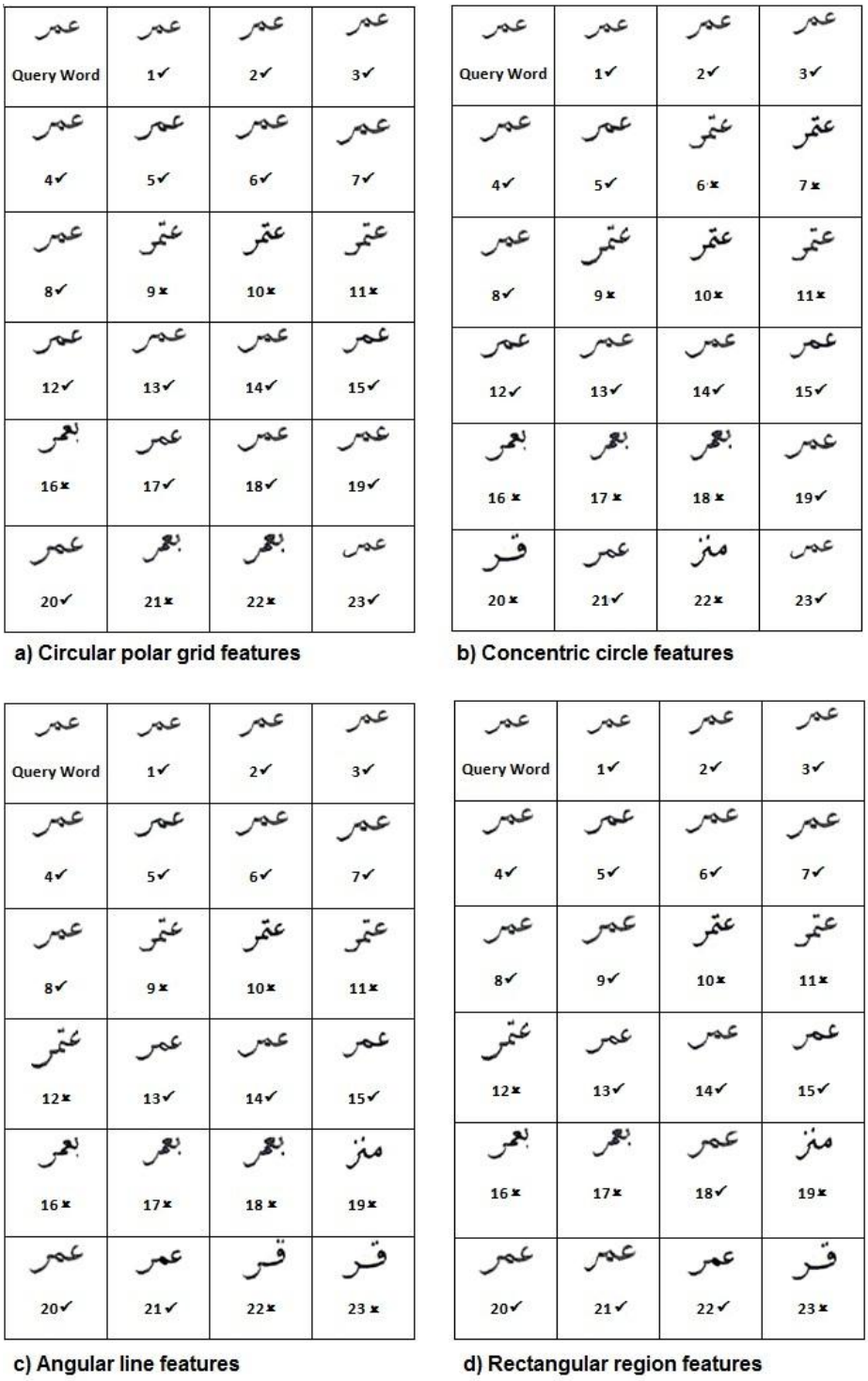

Figure 36: Experiment results for the query word Omar (عمر) using different feature sets

Figure 37: Experiment results for the query words "Peace be upon him" (صلى الله عليه) using different feature sets

5.4.2. Average Precision

We compute average precision over sets of query words in order to represent the performance of the system. Each set contains words that have the same number of connected components. Furthermore, the exact recall points which we compute the average precision are 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0. Figure 38, Figure 39 and Figure 40 represent the performance of the system. While Figure 41 represents the performance of the system using circular polar grid features.

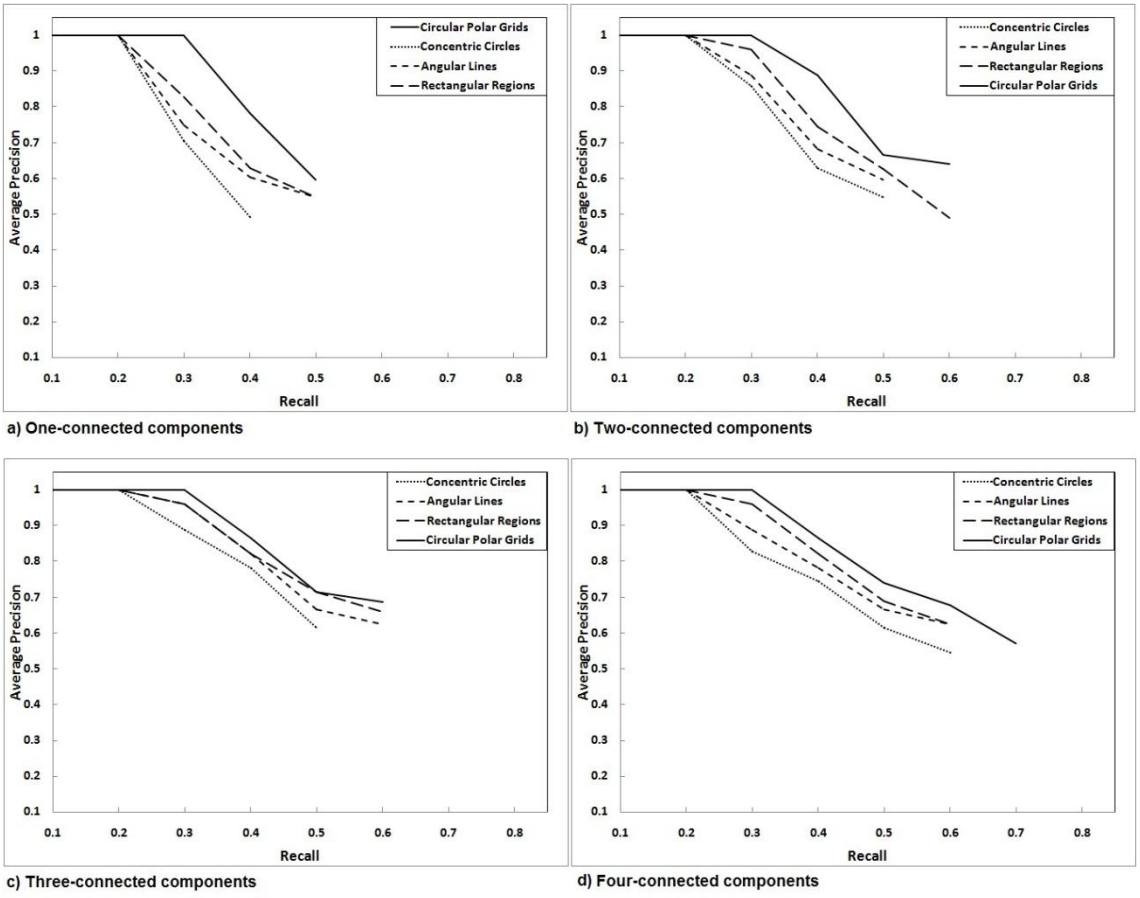

Figure 38: Average precision of normalized feature values

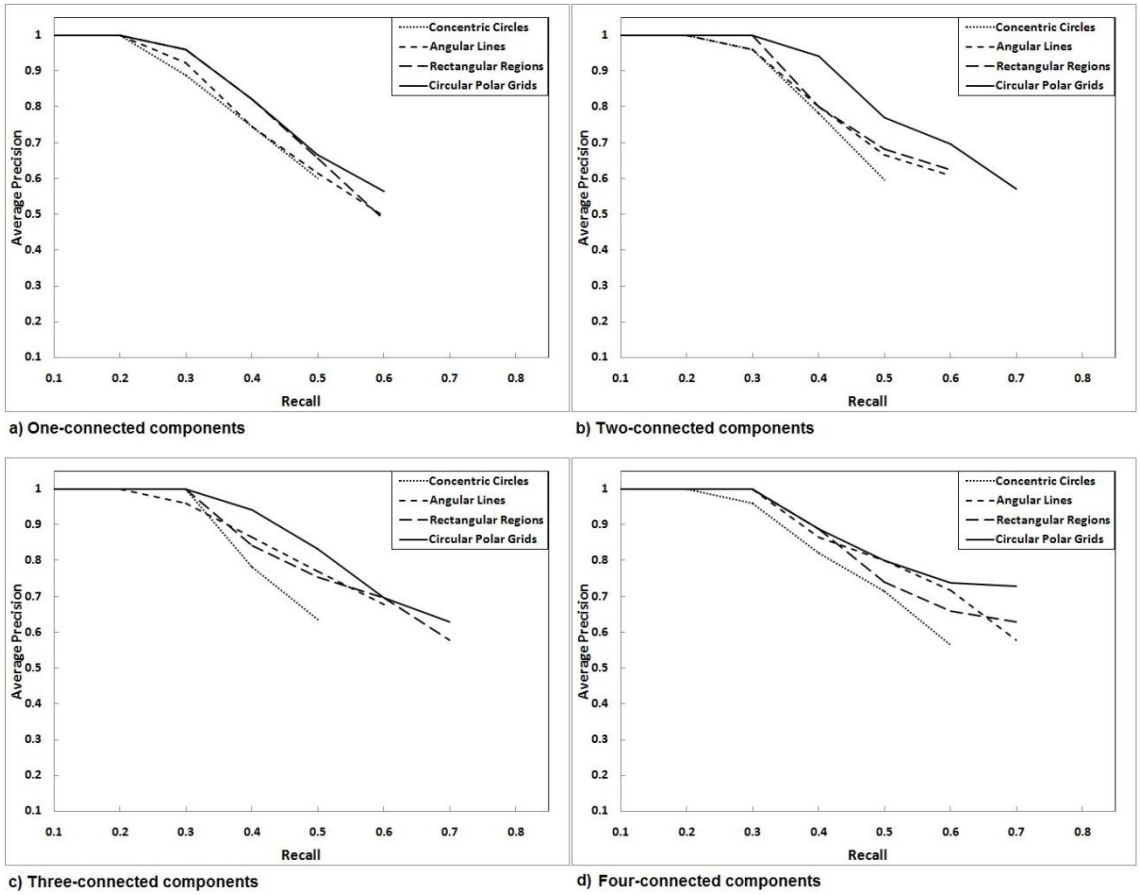

Figure 39: Average precision of feature occurrence values

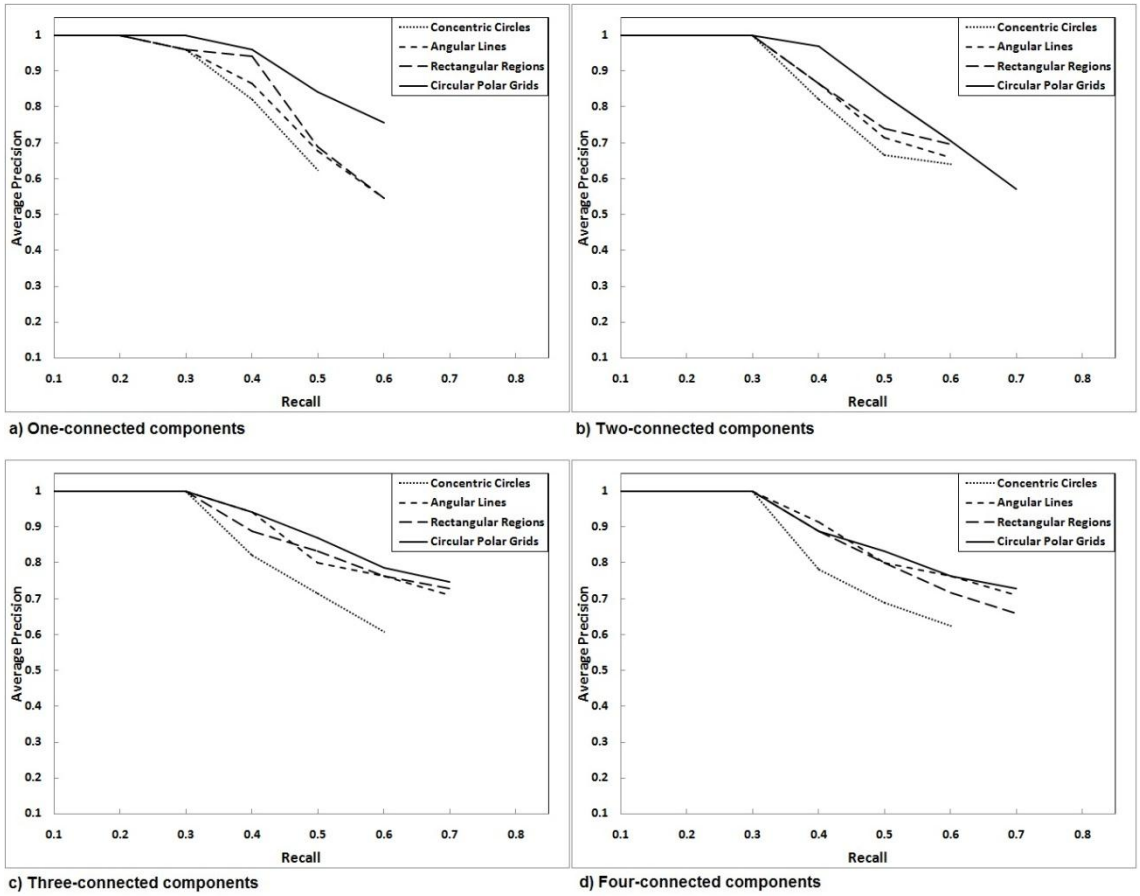

Figure 40: Average precision of term weighting feature values

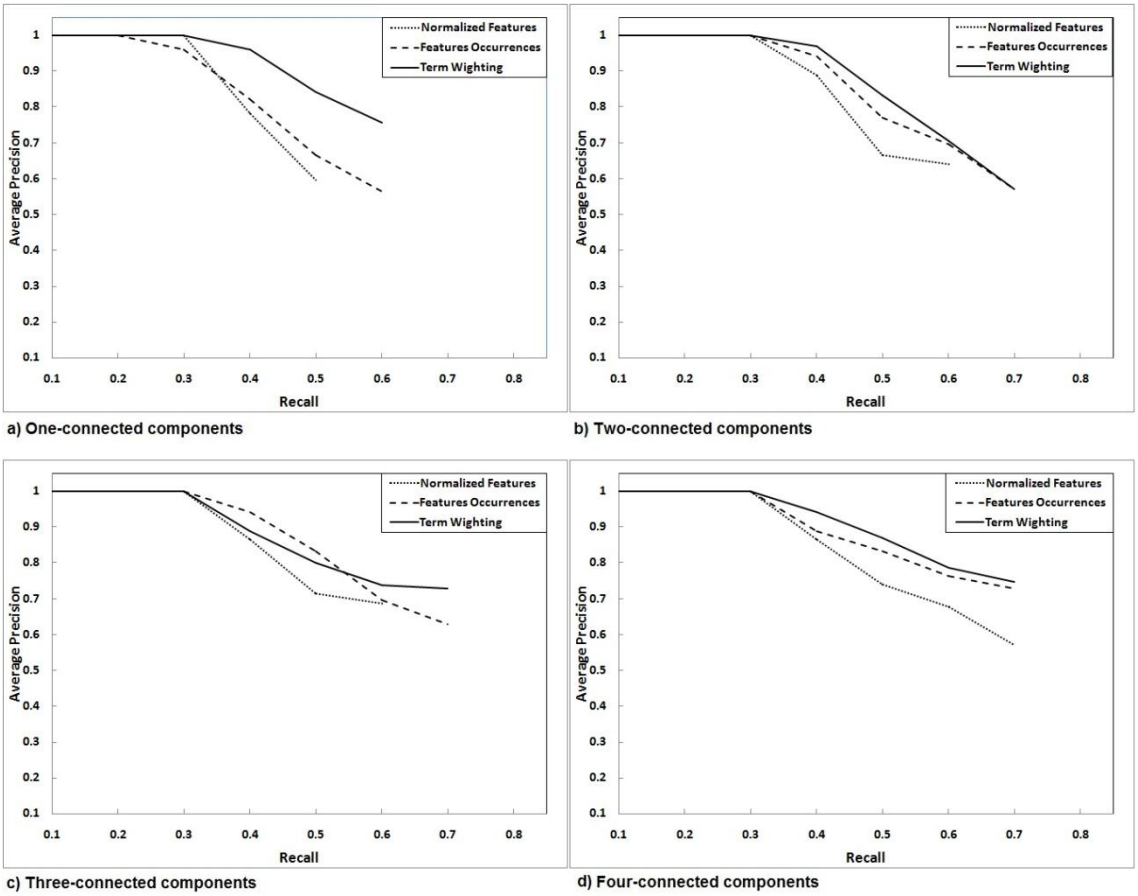

Figure 41: Different feature types of circular polar grid features

From the above figures (i.e. Figure 38, Figure 39 and Figure 40), we can observe the following points:

- The line slope of each figure downwards from left to right, enforcing the notions that as more relevant words are retrieved (recall increase), the more non-relevant words are retrieved (precision decreases).
- The circular polar grid features set performs better than other feature sets. While the concentric circle features set is the worst.

- The other angular line feature set and rectangular feature set perform next best. Additionally, these two feature sets do not differ significantly from each other.
- As evident from Figure 41 which related to the best feature sets, i.e. circular polar grid features, the average precision for all of feature types, e.g. normalized features, feature occurrences, and term-weighting features, increases steadily with the number of connected components.

5.4.3. Performance at Different Thresholds

The similarity in our system depends on the choice of the threshold value specified by how far the query word(s) is from the stored word(s) in the database. Therefore, when there is an exact match, i.e. the angle between the two is equal to zero, the cosine value is equal to 1. Since it is unlikely that an exact match occurs, we need to choose a suitable threshold that will be used to retrieve candidate words. In order to determine the value of that suitable threshold, we test our system with three threshold values, viz., 0.6, 0.7 and 0.8. We used these values such that if the cosine of the angle between a query and a word is greater than or equal to that threshold value, the word is retrieved. We carried out this experiment with nine queries. Table 9 shows the query words along with the precision and recall results associated with each query for the three different threshold values. In order to determine the most suitable threshold, Table 10 shows the number of retrieved results for different precision recall pairs. In this table, with the exception of precision recall pair values equal to (0.8,0.8) and threshold value equal to 0.8, we found that the threshold value equal to 0.7 along with recall precision pairs is

suitable threshold for retrieving relevant results more than the other two threshold values.

Table 9: Recall and precision results using different threshold values

Query Words	Total	Threshold	Total	Relevant	Non-	Precision	Recall
عمر	24	0.6	59	22	37	0.37	0.91
		0.7	44	19	25	0.43	0.79
		0.8	34	15	19	0.44	0.62
صلى	75	0.6	107	68	39	0.63	0.9
		0.7	98	65	33	0.66	0.86
		0.8	72	55	17	0.76	0.73
عباس	12	0.6	51	11	40	0.21	0.91
		0.7	17	10	7	0.58	0.83
		0.8	10	9	1	0.9	0.75
عائشة	10	0.6	54	9	45	0.16	0.9
		0.7	35	9	26	0.25	0.9
		0.8	29	9	20	0.31	0.9
المدينة	12	0.6	29	10	19	0.34	0.83
		0.7	16	10	6	0.62	0.83
		0.8	6	6	0	1	0.5
رسول	29	0.6	33	22	11	0.66	0.75
		0.7	24	19	5	0.79	0.65
		0.8	13	13	0	1	0.44
الشام	8	0.6	15	8	7	0.53	1
		0.7	11	8	3	0.72	1
		0.8	7	7	0	1	0.87
صلى الله	64	0.6	86	55	31	0.63	0.85
		0.7	63	45	18	0.71	0.70
		0.8	35	30	5	0.85	0.46
الحج و العمرة	8	0.6	22	6	16	0.27	0.75
		0.7	19	6	13	0.31	0.75
		0.8	7	5	2	0.71	0.62
رضي الله	28	0.6	35	24	11	0.68	0.85
		0.7	27	22	5	0.81	0.78
		0.8	21	19	2	0.90	0.67

Table 10: The number of results retrieved where the (precision, recall) pair for some queries is at least as specified in each column for different threshold values.

(Precision, Recall)	(0.9,0.9)	(0.8,0.8)	(0.7,0.7)	(0.6,0.6)	(0.5,0.5)
Threshold=0.6	0	0	0	4	5
Threshold=0.7	0	0	3	6	7
Threshold=0.8	0	1	3	5	6

5.4.4. Comparisons with (Shahab, Al-Khatib, & Mahmoud., 2006)

We used the same manuscript and the same query words that were used by (Shahab, Al-Khatib, & Mahmoud., 2006) and compared the recognition rate of our system with the proposed system of the other work. The recognition rate of the other work is about 76% while the recognition rate of our work for the same manuscript is 78.7 %. Table 11 shows the experimental results of our system for the same query words, while the retrieval results for the query word Al-Muhtasib (المحتسب) are shown in Figure 42. In this figure, the cells marked with true symbol correspond to relevant words, while the cells marked with false symbol correspond to non-relevant words. The discarded word is shown in Figure 43. The reason that the system discards retrieval of this word is that this word is concatenating with other word. One of the reasons that our work yields a better performance compared to the results reported by the other is that the precision here is better than the other. For example, the number of all retrieved words for Al-Muhatsib word in the other work is 68 words while in our work is 23. Figure 44 shows

the average precision of our work and the other's work. In this figure, we can observe that the overall of the average precision of our system is better than the other's work.

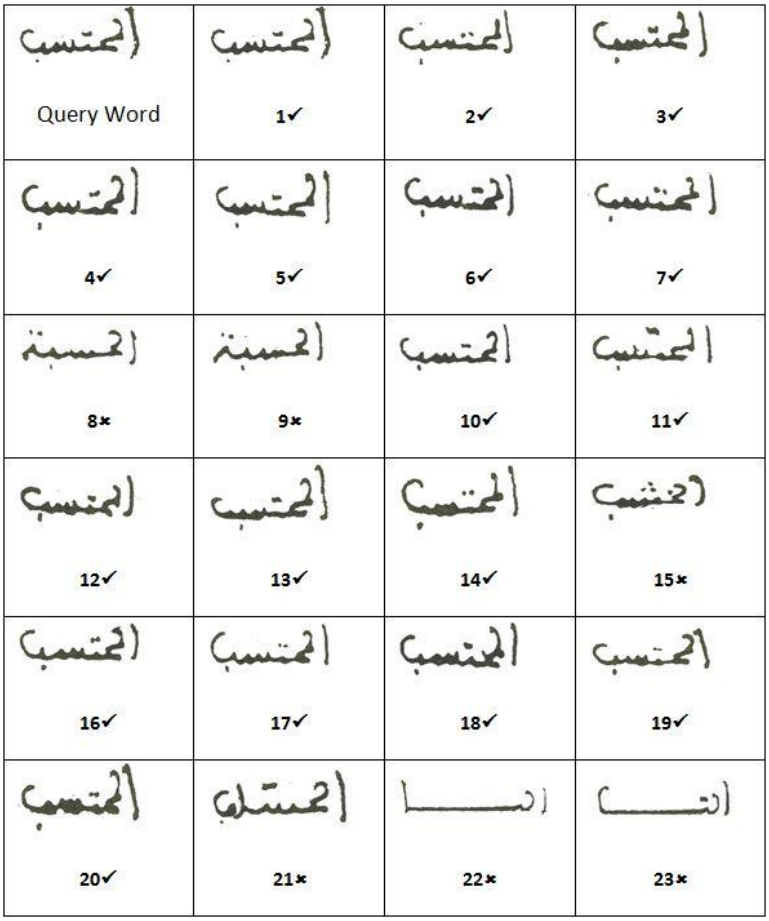

Figure 42: Example query resultsfor the word Al-Muhtasib (المحتسب)

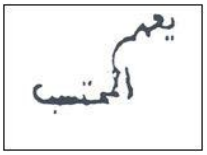

Figure 43: A discarded word of Al-Muhtasib (المحتسب) word

Table 11: Our experimental results for the same query words of (Shahab, Al-Khatib, & Mahmoud., 2006)

Query word	Total words	Total of Relevant
فصل	10	5
الحسبة	17	14
المحتسب	18	17
الباب	16	13
صلى الله	5	3

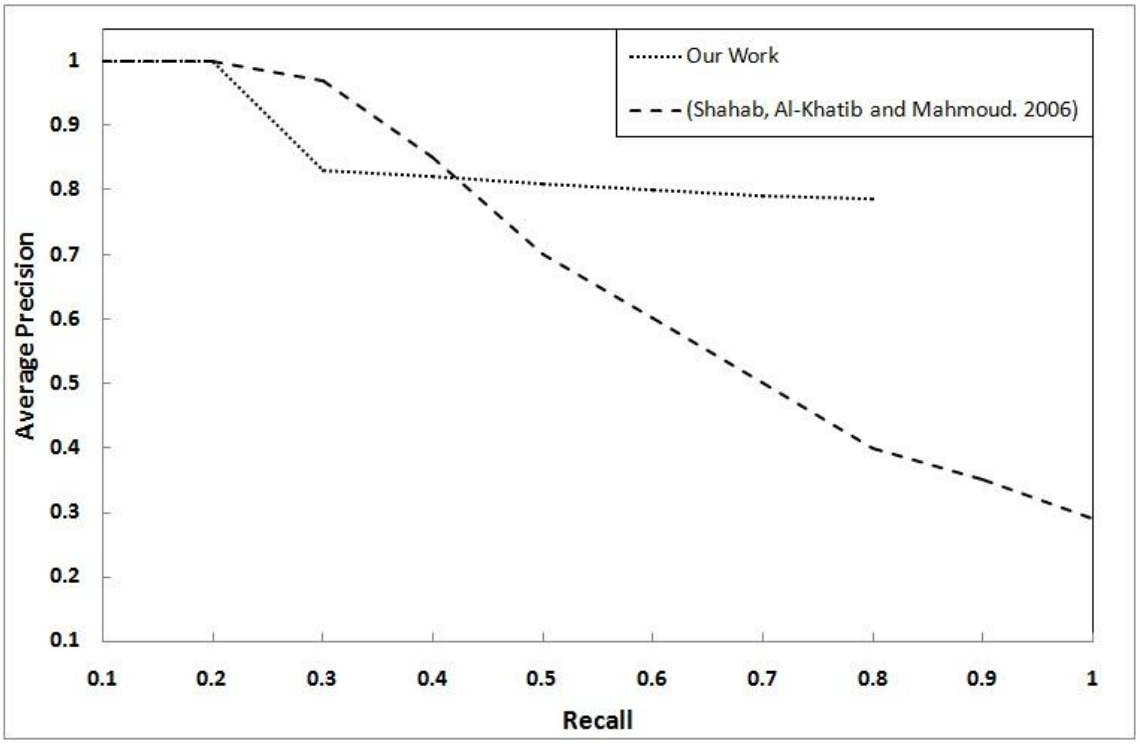

Figure 44: Average Precision of our work and the work of (Shahab, Al-Khatib, & Mahmoud., 2006)

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter presents the conclusions of the aspects studied in this research work along with the directions for the extension of this work are offered as possible future work.

6.1. Conclusions

An indexing and searching system for Arabic historical manuscript is developed and implemented. There are several modules such as preprocessing module, segmentation module, and features extraction module, applied before constructing the term-by-document matrix. Features such as concentric circular, angular lines, rectangular regions, and circular polar grids are extracted from the word images. Latent Semantic Indexing (LSI) is employed to represent subword images and their corresponding features as a term-by-document matrix. Query matching mechanisms are applied in order to retrieve candidate words. Additionally, setting a threshold for similarity matching using LSI is less dependent on the underlying features. The performance evaluation is measured using recall and precision. Our system shows encouraging results and is able to retrieve relevant words. Additionally, the newly proposed feature, circular polar grid, performs better than previously used feature sets for automatic indexing of Arabic handwritten manuscripts.

6.2. Future Work

For the future work, we are planning to:

- Use sample images of the manuscript to train a system that can carry out Optical Character Recognition on the whole manuscript with the premise of automatically digitizing its content.
- Use database management systems in order to increase the efficiency of the retrieval operations.
- Investigate other features that can improve the precision-recall of the system.
- Use the concepts of the relevance feedback in order to enhance the indexing process.

REFERENCES

Aghbari, Z. A., & Brook., S. (2009). Word stretching for effective segmentation and classification of historical Arabic handwritten document. *Third International Conference on Research Challenges in Information Science, RCIS*, (pp. 217-224).

Ahmad, I., & Grosky, W. (2003). Indexing and Retrieval of Images by Spatial Constarints. *Visual Communication and Image Representation, 14*, 291-320.

AlKhateeb, J., Ren, J., Jiang, J., & Al-Muhtaseb., H. (2011, June). Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking. *Pattern Recognition Letters, 32*(8), 1081-1088.

Al-Khatib, W., Shahab, S., & Mahmoud., S. (2007). Digital library framework for arabic manuscripts. In:Shahab, S. (ed.). *International Conference on Computer Systems and Applications, AICCSA '07* (pp. 458–465). Amman, Jordan: IEEE/ACS.

Alshuhri, S. (2008). Arabic Manuscripts in a Digital Library Context Digital Libraries. In *Digital Libraries: Universal and Ubiquitous Access to Information* (pp. 387-393). Berlin, Heidelberg: Springer.

Ataer, E., & Duygulu., P. (2007). Matching Ottoman words: an image retrieval approach to historical document indexing. *Proceedings of the 6th ACM CIVR* , (pp. 341–347).

Berry, M., Dumais, S., & O'Brien, G. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37*(4), 573–595.

Borbinha, J., Gil, J., & G. Pedrosa, a. K. (2006). The case of the digitized works at a national digital library. *Second International Conference on Document Image Analysis for Libraries, DIAL '06* , (pp. 116-125.). Lyon, France.

Bunke, H., Roth, M., & Schukat-Talamazzini., E. (1995). Off-line cursive handwriting recognition using hidden Markov models. *Pattern Recognition, 28*(9), 1399-1413.

Eakins, J., & Graham, M. (1999). *Content-based image retrieval.* Newcastle: University of Northumbria.

Gacek, A. (2009). Arabic Manuscripts: A Vademecum for Readers. Section 1: The Near and Middle East. In *Handbook of Oriental Studies* (Vol. 98). Leiden, Boston: Brill.

Gudivada, V., & Raghavan, V. (1995). Content-based image retrieval systems. *28*(9), 18-22.

Hare, J., & Lewis, P. (2005). On image retrieval using salient regions with vector-spaces and latent semantics. *In Image and Video Retrieval: Third International Conference (CIVR)*, (pp. 540–549).

Kane, S., Lehman, A., & Partridge., E. (2001). *Indexing GeorgeWashington's handwritten manuscripts.* Amherst, US: Center for Intelligent Information Retrieval, University of Massachusetts Amherst.

Knerr, S., Augustin, E., Baret, O., & Price., D. (1998). Hidden Markov Model Based Word Recognition and its Application to Legal Amount Reading on French Checks. *Computer Vision Image Understanding, 70*(3), 404-419.

Krishnan, N., Banu, M., & Christiyana., C. (2007). Content Based Image Retrieval using Dominant Color Identification Based on Foreground Objects. *International Conference on Computational Intelligence and Multimedia Applications.*

Li, Z., & Drew, M. (2003). *Fundamentals of Multimedia.* Englewood Cliffs, NJ: Prentice-Hall.

Mahmoud, S. (1994). Arabic character recognition using fourier descriptors and character contour encoding. *Pattern Recognition, 27*(6), 815–824.

Otsu, N. (1979, Jan). A threshold selection method from gray-level histogram. *IEEE Transactions Syst. Man Cybern., SMC-9*, 62–66.

Pecenovic, Z. (1997). Image Retrieval using Latent Semantic Indexing. *Master's thesis*. Department of Electrical Engineering, EcolePolytechnique Federal de Lausanne.

Rahman, M., Desai, B., & Bhattacharya., P. (2006). Visual keyword-based image retrieval using latent semantic indexing, correlation-enhanced similarity matching and query expansion in inverted index. *10th International Database Engineering and Applications Symposium, IDEAS*, (pp. 201–208.).

Rath, T., & Manmatha., R. (2003). Features for word spotting in historical manuscripts. Proc. of the 7th. *The 7th International Conference on Document Analysis and Recognition*, (pp. 512–527).

S. Zakariya, R. Ali, and N. Ahmad. (2010). Combining Visual Features of an Image at Different Precision Value of Unsupervised content based Iimage Retrieval. *International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1-4). IEEE.

Saykol, E., Sinop, A., Gdkbay, U., Ulusoy, O., & Cetin, A. (2004, March). Content-Based Retrieval of Historical Ottoman Documents Stored as Textual Images. *IEEE Trans Image Process, 13*, 314-325.

Shahab, S. A., Al-Khatib, W. G., & Mahmoud., S. A. (2006). Computer aided indexing of historical manuscripts. *Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation.*

Smeulders, A., Worring, M., S.Santini, Gupta, A., & Jain, R. (2000, Jan). Content-Based Image Retrieval at the End of the Early Years. . Mach. Intell. 22, 12,. *IEEE Trans. Pattern Analysis and Machine intellegence, 22*(12), 1349–1380.

Solomn, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing A Practical Approach with examples in Matlab.* Wiley Blackwell.

T, R., & R, M. (2003). Word image matching using dynamic time warping. *Conference on Computer Vision and Pattern Recognition (CVPR'03). II*, pp. 521-527. IEEE.

Tomai, C. I., Zhang, B., & Govindaraju, V. (2002). Transcript mapping for historic handwritten document images. *Proc. of the 8th Int'l. Workshop on Frontiers in Handwriting Recognition*, (pp. 413-418).

Vailaya, A., Figueiredo, M., Jain, A., & Zhang, H. (2001). Image classification for content-based indexing. *Image Process, 10*(1), 117-130.

Woods, R., Gonzalez, R., & Eddins, S. (2004). *Digital Image Processing Using MATLAB.* Prentice-Hall.

Xiaohong, Y., & Jinhua., X. (2008). The Related Techniques of Content-based Image Retrieval. *International Symposium on Computer Science and Computational Technology ISCSCT '08*, *1*, pp. 154-158.

Zhang, D., & Lu, G. (2003). Evaluation of similarity measurement for image retrieval. *International Conference on neural network and signal processing* (pp. 928–931). Nanjing, China: IEEE.

Zhu, X., & Huang., T. (2003). Relevance feedback in image retrieval: a comprehensive review. *Multimedia System, 8*(6), 536–544.

Zhuang, D., & Wang, S. (2010). Content-Based Image Retrieval Based on Integrating Region Segmentation and Relevance Feedback. *International Conference on Multimedia Technology (ICMT)*, (pp. 1-3).

VITA

Personal Information

Nationality	Jordanian
Place and Date of Birth	Kuwait, Dec 14, 1972
Marital Status	Married
Languages	Arabic (Native language), English (Competent user, IELTS Score : 6.5)
Address	Mathematics and Statistics Dept., King Faisal University, P.O.Box 380, Hofuf 31982, Saudi Arabia
e-mail	myahia1972@hotmail.com, omhnm@hotmail.com

Education

2011	**King Fahd University of Petroleum and Minerals**, Dhahran, , Saudi Arabia *Master of Science in Computer Science* ***(GPA 3.59 out of 4)***
1999	**Al-Isra Jordanian University**, Amman, Jordan *Bachelor in Computer Science* ***(Cumulative Average 90.3%)***

Experience

1999-Present — **King Faisal University,** Dhahran, Saudi Arabia
"Teaching & Research Assistance"

- Teaching Courses of Al-Hasa Teachers College (Educational Computers, Computer Mathematics, Visual Basic Programming, Advanced Visual Basic Programming, Principles of Data Base, Instructions Technology).
- Supervising Computer Labs (PCs and Windows Server 2003)
- Participating in designing and developing several systems (Admission & registration system, Human Resource systems, Financial systems, and web pages).
- Designing and Teaching Short training courses (MS Office)
- Participating in several committees (Admission and Registration Committee, Non-academic Activities Committee).

1996-1999 — **Al-Isra Jordanian University,** Amman, Jordan
"Computer Labs Technician & Operator"

- Supervised Computers Labs (UNIX, Windows NT, and Novell)
- Assisted in teaching labs courses (Programming Languages, Operating Systems)

Computer Skills

- Designing and developing web pages and application systems using Microsoft .Net Technologies.
- Designing database systems using MS SQL Server, Oracle, and MS Access.
- Programming languages: Java, C, and MS Visual Studio (VB and C#)
- Packages and Tools: MS Office, Matlab, Mathematica, and SPSS

Interests

- Arabic Computing
- Image Processing and Pattern Recognition
- E-learning
- Database and Web Development

Training Courses

Training Course Name	Hours	Year	Institute Name
Designing E-Portfolios	10	2011	King Faisal University, Saudi Arabia
Introduction to SPSS	10	2005	King Faisal University, Saudi Arabia
Development of Managerial Skills	10	1997	Al-Isra University, Jordan
Windows NT core Technology	20	1998	Information And Communication Systems
Novell Network V4.1	45	1997	The University of Jordan
TCP/IP Networking	20	1998	Information And Communication Systems, Jordan
UNIX Command & Utilities	20	1998	Information And Communication Systems, Jordan
Introduction to ORACLE	24	1996	Information And Communication Systems, Jordan
Oracle Developer Forms	24	1998	Information And Communication Systems, Jordan
Oracle Developer Reports	15	1998	Information And Communication Systems, Jordan
Oracle Data Base administrator	24	1998	Information And Communication Systems, Jordan
Management Skills Development	10	1997	The University of Jordan