ARABIC MANUSCRIPT LAYOUT ANALYSIS AND CLASSIFICATION

BY

GALAL MUNASSAR ABDULLAH BIN MAKHASHEN

A Dissertation 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

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

APRIL, 2018

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

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **GALAL MUNASSAR ABDULLAH BIN MAKHASHEN** under the direction of his thesis adviser 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 **DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING**.

Dissertation Committee

Mahmoud
Dr.Sabri A. Mahmoud (Adviser)

Marwan
Dr.Marwan Abu-Amara (Member)

Md. Rafiul Hassan
Dr.Md. Rafiul Hassan (Member)

Dr.Mohamed Mohandes (Member)

Dr.Shokri Z. Selim (Member)

For :
Dr. Adel Ahmed
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

18/7/18
Date

©Galal M. Bin Makhashen
2018

To my beloved parents
& family

ACKNOWLEDGMENTS

In the name of Allah, the most Gracious, the most Merciful.

All praises to Allah who gave me strengths, knowledge, ability, and opportunity to undertake this research and His blessing in completing this thesis.

I would like to thank and express my deep gratitude to my adviser ***Dr. Sabri A. Mahmoud****, for the support, invaluable discussions, and suggestions throughout this research that have contributed to the success of this work. Also, I would like to express my appreciation to the committee members for the fruitful discussions and suggestions during this research.*

I would like to express my gratitude to King Fahd University of Petroleum and Minerals for surrounding students with a healthy environment, providing access to highly impactable digital databases, and computing facilities.

My deepest gratitude and sincere thanks go to my beloved parents; ***Mr. Munassar A. bin Makhashen Al-Hamomi*** *and* ***Mrs. Faria'h M. AlJaidi Al-Marri****, and also to my beloved family, my wife and my children* ***Ala'a****,* ***Amer****, and* ***Nora*** *who surrendered me with endless love, prayers, and encouragement.*

TABLE OF CONTENTS

ACKNOWLEDGEMENT **iii**

LIST OF TABLES **viii**

LIST OF FIGURES **ix**

LIST OF ABBREVIATIONS **xiii**

ABSTRACT (ENGLISH) **xiii**

ABSTRACT (ARABIC) **xv**

CHAPTER 1 INTRODUCTION **1**

1.1 Historical Manuscript Analysis and Classification 1
    1.1.1 Document Layout Analysis 3
    1.1.2 Document Classification 5
    1.1.3 Manuscript Layout Analysis and Classification Framework . . 8
    1.1.4 Arabic Historical Handwritten Manuscript(AHHM) database . 10
1.2 Motivation 12
1.3 Problem Statement 13
1.4 Contributions of the Dissertation 13
1.5 Dissertation Organization 15

CHAPTER 2 LITERATURE REVIEW **16**

2.1 Document Preprocessing 16

2.1.1 Binarization methods for document images 17

2.1.2 Skew detection and correction 21

2.1.3 DLA Parameter Estimation 32

2.2 Document Layout Analysis Strategies 34

2.2.1 Bottom-Up Strategy 34

2.2.2 Top Down Strategy 42

2.2.3 Hybrid Strategy 47

2.3 Document classification 51

2.3.1 Query by example 54

2.3.2 Query by string 58

2.3.3 Hybrid query type 59

2.4 Performance Evaluation 61

2.4.1 Datasets 61

2.4.2 Evaluation Metrics 64

2.5 Discussions 72

2.5.1 Document layout analysis 72

2.5.2 Document Classification: Keyword spotting System 75

2.6 Conclusions 78

CHAPTER 3 ARABIC HISTORICAL HANDWRITTEN MANUSCRIPT DATASET **81**

3.1 Introduction 83

3.1.1 AHHM framework 85

3.2 AHHM Data Sources 85

3.3 Data Format and Properties 89

3.3.1 Ground truth methodology 91

3.3.2 Ground Truth Format 92

3.4 Keywords 93

3.5 Conclusions 97

CHAPTER 4 A HYBRID LAYOUT ANALYSIS FOR ARABIC HISTORICAL MANUSCRIPTS 98
4.1 Historical Manuscript Analysis Framework 99
4.2 Preprocessing 101
  4.2.1 Binarization 102
  4.2.2 Noise removal 104
4.3 Manuscript characterization 105
  4.3.1 Adaptive main-content localization 105
  4.3.2 Whitespace localization using anisotropic diffusion 106
  4.3.3 Static whitespace localization 115
  4.3.4 ADF and SWL integration 117
4.4 Feature Extraction 117
  4.4.1 Geometric features 118
4.5 Moving window analysis 121
  4.5.1 Feature based main-content segmentation 122
  4.5.2 Local main-content/side-note analysis 124
4.6 Conclusions 128

CHAPTER 5 DOCUMENT CLASSIFICATION BY KEYWORD SPOTTING 129
5.1 Introduction 130
5.2 Related Work 132
5.3 Proposed word spotting system 137
  5.3.1 Feature Extraction 137
  5.3.2 Skeleton-based interest points 142
  5.3.3 Feature Representation Using Bag of Visual Words 145
  5.3.4 Support Vector Machines (SVM) 146
  5.3.5 Recognition response analysis 147
5.4 Conclusions 150

CHAPTER 6 EXPERIMENTAL RESULTS 152

6.1 Experimental setup 152

6.1.1 Datasets 153

6.1.2 Evaluation metrics 156

6.2 Document Layout Analysis Results 160

6.2.1 Discussions 164

6.3 Document Classification Results 164

6.3.1 Training Results 165

6.3.2 Spotting Results 169

6.4 Error Analysis 173

6.4.1 Document Layout analysis 173

6.4.2 Document Classification 174

6.5 Conclusions 178

CHAPTER 7 CONCLUSIONS AND FUTURE DIRECTIONS 180

7.1 Concluding Remarks 180

7.2 Future Research Directions 183

REFERENCES 186

VITAE 224

LIST OF TABLES

1.1 Strengths and weaknesses of DLA essential strategies 5

2.1 Skew detection and correction: a summary 31
2.2 Document layout analysis algorithms: a summary 49
2.3 Surveyed Keyword spotting methods 60
2.4 Document layout analysis and spotting datasets 64
2.5 Examples of DLA evaluation metrics & results 70

3.1 AHHM dataset resources 89
3.2 AHHM selected keyword samples 95
3.3 AHHM Dataset Summary 96

6.1 Samples of HADARA80P dataset's keywords 154
6.2 Performance evaluation using [1] metric 161
6.3 Segmentation accuracy based on success rate 161
6.4 Recognition results using HADARAH80P dataset 167
6.5 Recognition results using AHHM dataset 168
6.6 Keyword spotting performance on HADARA80P dataset 172
6.7 Keyword spotting performance on AHHM dataset 173

LIST OF FIGURES

1.1 Keyword spotting taxonomy of methods 8
1.2 General overview of Arabic historical analysis and classification (MLAC) system. 9
1.3 Comparison between AHHM and HADARA80P databases 11

2.1 Projection profile (vertical projection) on normal text lines 21
2.2 Static versus dynamic local projection profile analysis 23
2.3 Example of data reduction before applying Hough transform [2] . . . 24
2.4 Skew detection in frequency domain analysis; a) original document image, b) Fourier transform magnitude, c) Radon transform [3] 27
2.5 Confusion of gradient performed on letter ”e” [4] 28
2.6 Comparison between contemporary and historical documents characteristics 33
2.7 Document layout analysis strategies: main techniques 34
2.8 Example of Rose of Directions[5],a) different images, b) autocorrelation response, c) rose of direction response. 37
2.9 Energy map text line segmentation[6] 43
2.10 Keyword spotting systems: main design options 52
2.11 Bottom-up techniques relationship set-view 73
2.12 Document layout analysis techniques statistics 74
2.13 Summary of KWS literature 77

3.1 Arabic word segmentation variations 83
3.2 AHHM framework phases 86

3.3 Example of digital libraries software interface 87
3.4 AHHM manuscript samples 90
3.5 Examples of block and word segmentation; a) Block segmentation, b) Word segmentation 91
3.6 Two examples of keywords that have different writing fonts, and styles, a) KW06, b)KW24 92
3.7 XML ground truth example 94

4.1 Manuscript samples 100
4.2 General overview of the proposed algorithm. 101
4.3 Binarization example 102
4.4 Global and Local Binarization Comparison 103
4.5 ADF illustration on different simulated text situations; a) Touching text lines, b) Normal text lines, c) Regular spaced, not aligned text lines, d) Irregular aligned and spaced text lines, e) Scattered text components 111
4.6 An example of two types of white components, **a)** simulated input-text, **b)** ADF response using estimated S_x, **c)** ADF response with Fixed scale $S_x = 15$. 113
4.7 ADF main-content detection pipeline. a) An input manuscript; b) Vertical and horizontal responses; c) Removing weak response; d) Integrating vertical and horizontal preprocessed responses; e) Adjust left and right responses based on top and bottom clues and vice versa if needed; f) Generate ADF mask; g) Main-content result 114
4.8 Static whitespace mask generator; a) An input manuscript, b) SWL scans; vertical (first row), and horizontal(second row), c) SWL outputs, d) Cleaned outputs, e) Combined vertical and horizontal outputs. . . 116
4.9 An integration mask using ADF and SWL 118
4.10 Distance Transform response. a) A patch taken from Main-content region; b) A patch taken from side-notes left part 121

4.11 Moving Window stop conditions. 123
4.12 Main-content region boundary detection; a) Moving stops by different conditions, b) Stop window correction 124
4.13 Main-content segmentation 127

5.1 Overview of the proposed Keyword Spotting 132
5.2 Comparison between SIFT and SURF responses [7] 135
5.3 Integral image calculation 139
5.4 Gaussian approximation (Box filter) 140
5.5 Overview of feature extraction using SURF 143
5.6 Handwritten key points sampling using different approaches 144
5.7 Bag of Visual Words framework 145
5.8 Identify the best three trained SVMs for spotting task 147
5.9 SVM recognition and classification behavior modeling 148
5.10 An example of SVMs response distributions 149
5.11 Spotting thresholds, t_1 estimated between matching/mismatching, t_2 computed between matching/rejection, t_3 is the matching mean . . . 150

6.1 Two page examples from each dataset 155
6.2 PRImA Framework errors 157
6.3 Sample results on Bukhari dataset. a) Segmentation results of [1], b) The proposed method results 162
6.4 Sample results on AHHM dataset. a) Segmentation results of [1], b) The proposed method results 163
6.5 Average recognition time of each method 168
6.6 Two keyword examples and their corresponding word instances 170
6.7 Confusion due to similar shape structures; a) Comparison between keyword HKW14 (Radhi) and another word ”To advise”, b) Comparison after removing written diacritics from keyword HKW14 (Radhi) . . . 171
6.8 Analysis of the error rates breakdown using Bukhari dataset 175
6.9 Analysis of the error rates breakdown using AHHM dataset 176

6.10 Examples of retrieved word errors 177

THESIS ABSTRACT

NAME:	Galal Munassar Abdullah Bin Makhashen
TITLE OF STUDY:	Arabic Manuscript Layout Analysis and Classification
MAJOR FIELD:	Computer Science and Engineering
DATE OF DEGREE:	April, 2018

Vast number of historical Arabic manuscripts is available in digital form online where automatic classification, indexing, and retrieval may be needed. Such tasks depend heavily on the quality of the manuscript layout analysis and classification (MLAC) system. Usually, Optical Character Recognition (OCR) system fails to address such tasks for historical manuscripts. Historical manuscripts suffer from various degradations such as aging, touching-text, faint-text, ink-bleeding, show-through, broken words, unorganized text spacing, and text skewing. Such manuscripts'characteristics make OCR infeasible. Unlike OCR, the MLAC system does not convert documents to text but uses image patches to classify and retrieve documents that match these patches.

In this thesis, we propose a MLAC system for Arabic historical manuscripts that consists of two main phases; document layout analysis, and document classification. We *propose a hybrid document layout analysis approach using anisotropic diffusion of whitespace analysis (as a top-down strategy) and moving window approach powered by connected component analysis (as a bottom-up strategy). The proposed approach performs segmentation at regional levels where it aims at extracting manuscripts' main-content. We also propose a learning-based keyword spotting system (KWS) using word-skeleton and Speeded-Up Robust Features (SURF). The word-skeleton adapts to the nature of handwriting strokes and indicates important interest regions by intrinsic word structure. Moreover, we also propose a novel spotting thresholding method that is objectively estimated by considering the recognition behavior of Support Vector Machines (SVMs) in the training phase.*

In addition, we present an Arabic Historical Handwritten Manuscript (AHHM) database consisting of 108 manuscript pages collected from two main digital libraries. The database consists of manuscripts dated between 10th to 18th centuries from the Islamic heritage by scholars in Hadith, Islamic Doctrines, and Sufism. Moreover, the manuscripts are segmented into main-content, side-notes and words. There are 2135 segmented words and 25 keyword classes. We evaluated our Manuscript Layout Analysis & Classification system (MLAC) on three datasets; Bukhari, HADARA80P, and AHHM datasets (AHHM is developed in this work). The performance of our layout analysis system and document classification yield promising results with success rate up to 98.83% PRImA metric, and 91.40% in terms of $F_{measure}$ *respectively.*

ملخص الرسالة

الاسم: جلال مُنَصَّر عبدالله بن مُخاشِنْ
عنوان الرسالة: تحليل هياكل المخطوطات العربية والتاريخية وتصنيفها
التخصص: علوم وهندسة الحاسب الآلي
تاريخ الدرجة العلمية: ابريل 2018

تحوي العديد من المكتبات الرقمية حول العالم عدداً كبيراً من المخطوطات العربية والتاريخية النادرة والتي تشكّل مصدراً مهماً للعلماء والباحثين والمهمتين في ابحاثِهم وفي دراسة التراث العربي. عادةً ماتُعرض المخطوطات بصيغة صور رقمية وليست نصوصاً مما يُعيق عمليات البحث عن المعلومات واسترجاعها من قبل الباحثين. وهذا يشكّل ضعفاً بتقنيات إسترجاع المعلومات وخاصةً التي تتعامل مع المخطوطات التاريخية، وبالتالي فإن ماتوفرة المكتبات الرقيمة من مخطوطات لم يُستَغل بالشكل الأمثل. ولمعالجة هذه المشكلة، تحتاج هذه المخطوطات التاريخية إلى معالجة تحليلية لهيكلها (Layout Analysis) لاستخراج محتواها الرئيسي ومن ثم القيام بفهرستها بناءً على ذلك المحتوى.

في هذه الدراسة اقترحنا منظومة متكاملة للتحليل الالي والتصنيف للمخطوطات العربية التاريخية (MLAC). تتكون المنظومة المقترحة من تقنيتين رئيسيتين هما تقنية تحليل هياكل المخطوطات (Document Layout Analysis) لاستخراج النص الرئيسي، وتقنية إسترجاع المخطوطات باستخدام طريقة البحث عن الكلمات ذات العلاقة (Keyword Spotting).

قمنا بتطوير نظام تحليل هجين (Hybrid) يضم أسلوب التحليل التنازلي (Top-down) وأسلوب التحليل التصاعدي(Bottom-Up) لتحليل المخطوطات التاريخية. يعمل النظام في مرحلة التحليل التنازلي وبشكل آلي على تحديد النص الرئيسي للمخطوطة مبدئياً ثم يستخلص مجموعة من المميزات الهندسية لتلك النصوص. وبعدها يعمل نظام التحليل التصاعدي على رصد وتتبع مسار النص الرئيسي للمخطوطة باستخدام تلك المميزات الهندسية لتقسيم المخطوطة بشكل نهائي إلى نص رئيسي وهوامش.

كما اقترحنا طريقة البحث عن الكلمات ذات العلاقة لتصنيف المخطوطات عن طريق نصوصها الرئيسية. الجديد في اقتراحنا لهذه الطريقة هو تحسين استخلاص المميزات (Feature extraction) من صور الكلمات باستخدام هيكل الكلمة (word–skeleton) بالإضافة إلى تحليل أسلوب التعليم الآلي أثناء التدريب لمعرفة عوامل اتخاذ قرار التصنيف بها، ومن ثم تكييف الباحث الآلي باستخدام تلك العوامل لاسترجاع المعلومات بشكل ادقّ.

ولتقييم أداء المنظومة المقترحة، قمنا بتطوير قاعدة البيانات (AHHM) للمخطوطات العربية والتاريخية والتي تتألف من 108 صفحة تم تجميعها من مكتبتين رقميتين هما مكتبة هارفرد ومكتبة برلين. هذا وتحتوي قاعدة البيانات في المجمل على 2135 كلمة تم استخراجها من المخطوطات، كما تم اختيار 25 كلمة رئيسية منها للقيام بعملية التصنيف والاسترجاع للمخطوطات. كما سيتم إتاحة قاعدة البيانات هذه للباحثين مجاناً.

CHAPTER 1

INTRODUCTION

In this chapter, we present an introductory material on document layout analysis and classification. Section 1.1 highlights the importance of document layout analysis and classification for historical manuscripts. Moreover, it gives an introduction to the proposed techniques for manuscript layout analysis and classification. Also, it provides an overview of the Arabic Historical Handwritten (AHHM) database developed in this work. The motivation of this study is presented in Section 1.2. Section 1.3 states the problem statement of the dissertation. In Section 1.4, we highlight the main contributions of this work.

1.1 Historical Manuscript Analysis and Classification

Historical manuscripts are valuable documents that reflect the human heritage and form an important source for historical studies. Often, digital libraries grant access to scanned copies of historical manuscripts and preserve original manuscripts in a controlled environment. Searching for a piece of information within these scanned manuscripts is time-consuming due to the absence of manuscript's digitized transcriptions.

There are two essential approaches that address scanned documents in general; optical character recognition (OCR), and Document Image Retrieval (DIR).

Optical Character Recognition is widely used to analyze and process scanned-documents and produce their digitized versions. Therefore, OCR allows fast text-searching for large-scale archives. However, the performance of OCR is degraded severely on processing cursive handwritten documents [8]. Usually, it fails to analyze handwritten documents due to several reasons such as unrecognized font types, irregular writing styles, touching-text, ink-bleeding, etc. [9].

Document Image Retrieval is considered an alternative approach to OCR, which can perform image-based searching to retrieve documents [10, 11]. DIR can be implemented globally by retrieving matches using overall document structure, or locally by retrieving documents with similar keywords, signatures, or logos.

Unlike regular documents, historical manuscripts are, in most cases, composed of complex layouts. They may contain text with irregular spacing, different font sizes, multi-writer styles. In addition, historical manuscripts suffer from various degradations such as aging, faint text, show-through, touching-text, ink-bleeding, decoration-text, marginal text etc. Such issues have made historical manuscript analysis challenging task.

Several digital libraries are archiving large amount of historical manuscripts that lack some searching facilities. For example, the Islamic Heritage Project (IHP) of Harvard library [12] offers 156,000 of ancient materials (e.g. manuscripts, maps etc.) that lack searching service especially within historical manuscripts. In other words, a visitor can only search for historical manuscripts using titles or subject categories, but within historical manuscript books, a visitor needs to read through all pages to find information.

Manuscript layout analysis and classification (MLAC) tries to fill this gap exploiting the advances in computer technology to perform easy and fast information retrieval. These systems are not equivalent to OCR that converts images to text for information retrieval. In MLAC, historical manuscripts are analyzed to extract main contents, text lines, and/or words. Then, manuscripts can be indexed by their keywords to facilitate information retrieval. The analysis of historical manuscripts is still in its early stages of maturity due to large number of analysis objectives. In other words, there is no single solution that can address all issues of historical manuscripts at once. Therefore, the aim of this dissertation is to contribute to the research in Arabic historical manuscript analysis and classification by providing a benchmark dataset and developing techniques to analyze and retrieve manuscripts.

1.1.1 Document Layout Analysis

Document Layout Analysis (DLA) consists of two general stages; physical and logical analyses. The physical layout analysis determines structural regions of a document image. In other words, it performs document segmentation at various levels such as blocks, regions, and zones. Then, these segmented components may be further analyzed into functional or logical entities (i.e., labeling). For instance, the segmented blocks, or regions can be text columns, paragraphs, figures, tables, etc., but zones usually refer to smaller entities such as words. Furthermore, the logical analysis may determine the reading order and infer the relationships of document regions.

DLA algorithms are divided into three analysis strategies; bottom-up, top-down, and hybrid. Bottom-up strategy starts document analysis at small document-element levels such as pixels, or connected components. Then, it identifies and groups homogeneous elements to form larger zones such as words. It continues grouping zones repeatedly until no more elements can be grouped. The layout analysis objective has to be set beforehand.

On the contrary, top-down strategy performs the analysis at the document level first. It identifies blocks at high level, which may contain several regions, and zones. For example, Krishnamoorthy et. al [13] started by dividing a document into column blocks using whitespace analysis. The analysis in top-down strategy continues dividing large document elements into smaller elements until no further division can be made, or a target analysis level has been reached. For example, whitespace analysis as in [14] targeted column regions, and X-Y cuts algorithm was applied on detected column-regions to extract text-lines as in [13].

Both essential strategies have their strengths and weaknesses. In one hand, a bottom-up strategy is naturally slower to analyze documents in comparison to top-down. Be-

Table 1.1: Strengths and weaknesses of DLA essential strategies

Metrics		**Bottom-Up**	**Top-Down**
Performance	Speed	Slow-moderate	Moderate-high
	Accuracy	Moderate-high	Low-moderate
Parameterized	–	Often	Rare
Layouts	Complex	Yes	NA
	Regular	Yes	Yes

cause it processes large number of document elements at lower levels. On the other hand, top-down strategy is less precise to define regions' boundary cuts [15]. Table 1.1 lists the most crucial factors of top-down and bottom-up strategies.

Hybrid strategy is the third type of document analysis strategies that combines both bottom-up and top-down methods [16, 17]. It may start the analysis using top-down, or bottom-up based on the analysis objectives. Usually, it is used in analyzing more complex document layouts and aiming to produce better segmentation results by integrating the strengths of both strategies.

1.1.2 Document Classification

Document analysis can be considered as a preprocessing phase to document classification. The document classification can be divided into two main types; global, and local. In global classification, documents' content is not important, only document structure is used to classify documents [18]. In this case, the preprocessing does not include segmentation task, rather it consists of image enhancement and noise filtering. On the other hand, document classification can be approached locally. In this scenario, documents are characterized by their content such as logos, signatures, and words. In this work, we consider local document classification using keyword spotting techniques.

Keyword spotting (KWS) is a searching method that locates all instances of a query word. This technique has emerged as an effective technique for large-scale document retrieval and classification. In particular, it is used to retrieve degraded and handwritten documents [19].

KWS is usually confused with word recognition approach. To distinguish between both tasks, KWS approach retrieves all matched word-instances to a given keyword-query image. On the other hand, word recognition task is meant to recognize the semantic information of a given word image. In other words, it produces transcripts of the recognized words using lexicon and language models [20]. In KWS, a user formulates a word-query and the KWS computes similarity-scores to all possible keyword templates. Then, it returns a ranked list of matched template keywords that are most similar to the queried word. This process includes preprocessing, feature extraction/representation, and matching tasks [11].

Generally, keyword spotting methods can be divided into three categories according to algorithms common factors such as type of preprocessing, query, and matching[11]. First, KWS can be either segmented or segmentation-free system. In segmented-based KWS, the system segments and indexes all words or keywords that characterize different documents' content in off-line mode. Then, in on-line mode, the system accepts queries and performs matching to retrieve ranked list of matches. On the other hand, the segmentation-free KWS may accept keyword query and perform spotting at a document image-level or at the level of text-lines[21].

Secondly, KWS can be categorized based on the type of query into Query By Example (QBE), Query By String (QBS), or hybrid KWS. In QBE, a user can initiate an image-based word query, and the KWS spots similar words in document archives[22]. Secondly, a user can type-in string query. Then, the KWS may use recognizers such as Hidden Markov Models(HMM) to search directly in documents and match recognized words with the string query[23]. Alternatively, It may use string-to-image synthesizing technique to generate image-based word query and perform spotting as QBE[21]. Lastly, QBE and QBS can be supported simultaneously by the KWS system. In this case, it is called hybrid query[20].

Finally, based on how matching is performed, the KWS can be categorized to either learning-based or template-based spotting system. In learning-based KWS, the system uses machine learning models to perform matching. This type of KWS is characterized by low memory requirements, since it needs no further data at the operation mode. In operation mode, the system uses a trained classifier to guess the keyword class of the input query. On the other hand, template-based spotting has no off-line mode, it carries out instant matching of input query with all template words. Therefore, the template-based KWS requires both query and templates at the matching stage [24]. Figure 1.1 illustrates general taxonomy of keyword spotting methods.

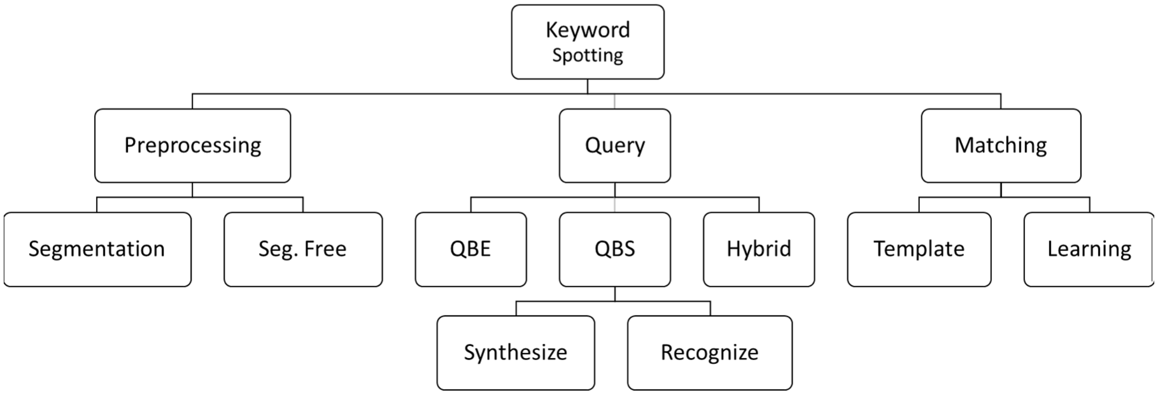


Figure 1.1: Keyword spotting taxonomy of methods

1.1.3 Manuscript Layout Analysis and Classification Framework

In this section, we propose our Manuscript layout analysis and classification (MLAC) framework. It consists of two main phases; document layout analysis and keyword spotting. The DLA is responsible for preprocessing manuscripts. It performs binarization and noise removal. Then, DLA carries out hybrid analysis, which starts with top-down techniques to extract features from manuscripts' main regions. Then, it performs bottom-up analysis to segment manuscripts main content. The main content is indexed using keywords and stored in Arabic Historical Handwritten Manuscript(AHHM) database.

Secondly, the keyword spotting method has two main modes; configuration, and operation. In configuration, the KWS extracts word features and use Bag of Visual Words(BoVW) to compute their fixed representation. Then, a set of Support Vector Machines (SVMs) are trained in a multi-class strategy to learn all predefined keyword classes. During SVMs validation, MLAC models the SVMs matching behavior to esti-

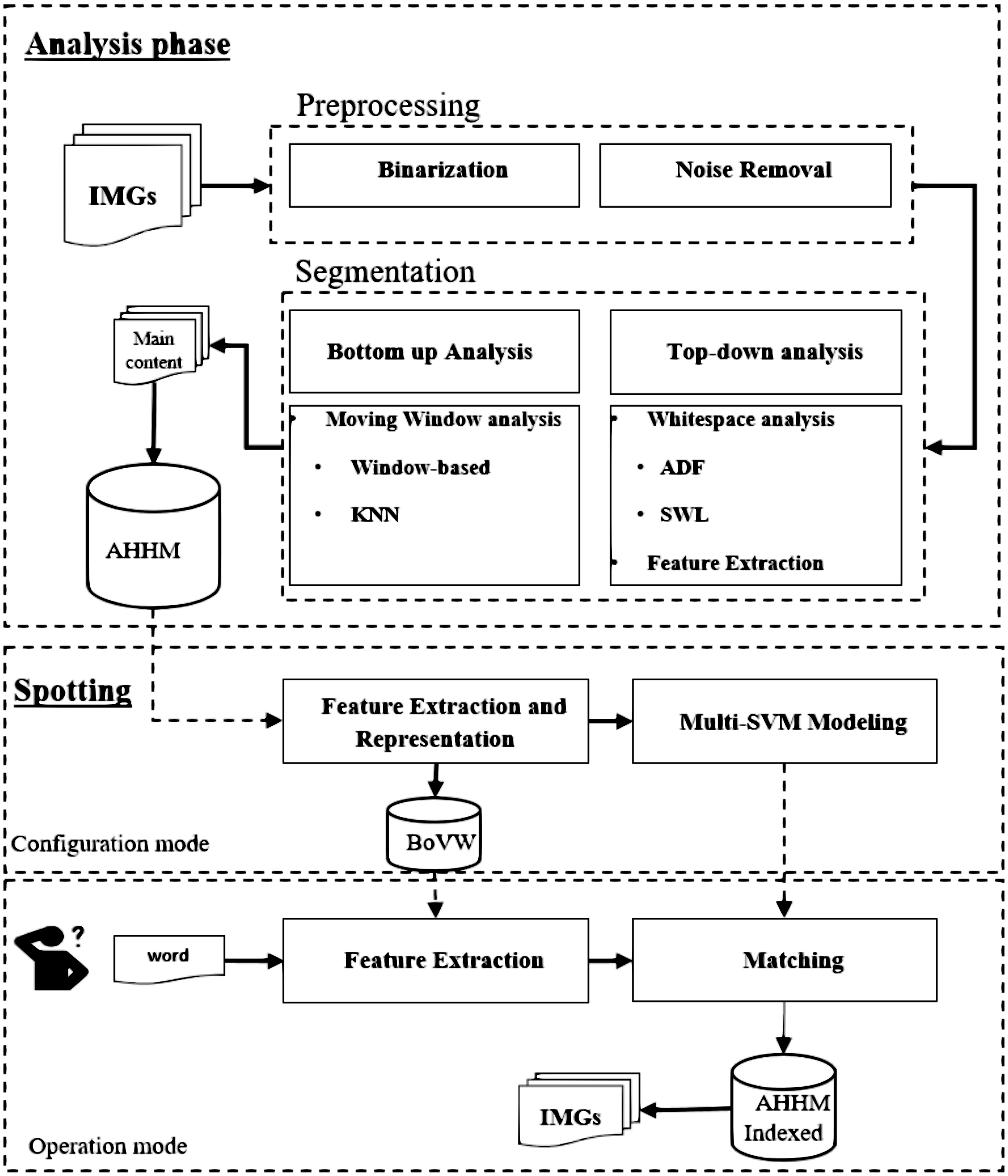


Figure 1.2: General overview of Arabic historical analysis and classification (MLAC) system.

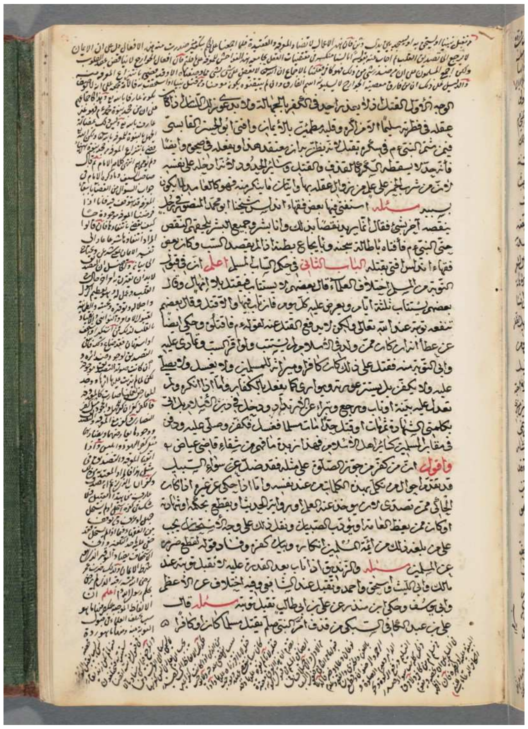
mate spotting thresholds. Finally, in operation mode, MLAC uses trained SVMs and estimated thresholds to perform keyword spotting. Figure 1.1 illustrates an abstract framework of the proposed MLAC system.

1.1.4 Arabic Historical Handwritten Manuscript(AHHM) database

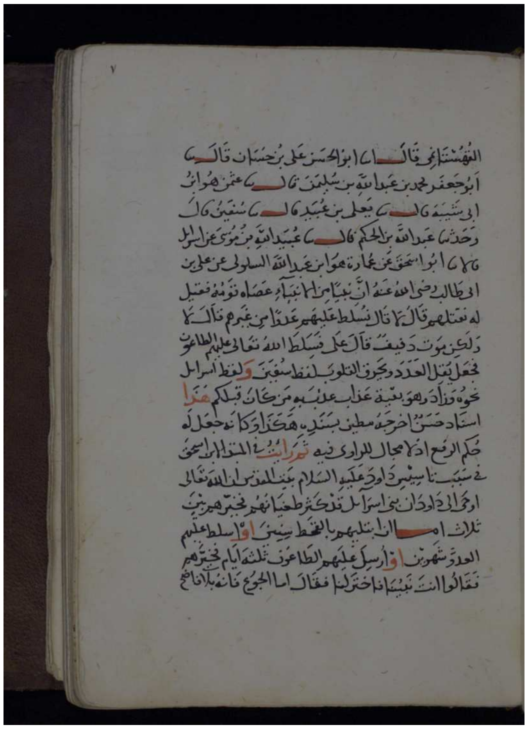
Recently, an Arabic historical database has been published to benchmark keyword spotting methods; HADARA80P [188]. Pages of HADARA80P database contain the main-content text without side-notes. They are extracted from a single manuscript written by single writer. This makes HADARA80P database suits keyword and classification algorithms while it does not suit regional analysis because each page contains only uni-content text.

Unlike HADARA80P database, our Arabic Historical Handwritten Manuscript (AHHM) database is designed to address document layout analysis and classification. Therefore, it consists of various pages extracted from four manuscripts. These pages contain two types of text; main-content and side-notes. The AHHM allows the evaluation of analysis algorithms that perform regional analysis such as extraction of main-content or side-notes. Moreover, it can be used to evaluate text-line and complex word analysis.

The AHHM pages are characterized by sets of keywords that are extracted manually. Therefore, keyword spotting and classification techniques can be evaluated using AHHM database. An example of HADARA80P and AHHM is shown in Figure 1.3. Further details are given in Chapter 3.



(a) AHHM Page Example



(b) HADARA80P Page Example

Figure 1.3: Comparison between AHHM and HADARA80P databases

1.2 Motivation

Arabic heritage contains many valuable historical manuscripts that are informative and content-rich in Islamic and historical information. Motivated by advances in communication and information technology, digital libraries offer vast number of precious Arabic historical manuscripts on-line. However, these historical manuscripts are only available as images, hence textual search is not possible.

The well-established Optical Character Recognition (OCR) can be used to convert scanned documents into textual form. However, it requires clean and regular documents to produce acceptable results. On the other hand, document classification can form a promising solution to retrieve documents with minimum image-to-text conversion. Most of the document classification algorithms have been tested using non-Arabic documents. Moreover, the majority of previous researches were conducted on contemporary documents due to the lack of Arabic historical manuscript benchmarking data set.

Giving the above facts, there are several gaps that need to be addressed. The aim of this dissertation is to fill some gaps, improve the state-of-the-art in Arabic historical manuscript analysis and classification, and build a benchmarking Arabic historical dataset.

1.3 Problem Statement

Searching and retrieving Arabic historical manuscripts have been a limitation facing digital libraries. By considering readers' time, a search for a piece of information in scanned historical manuscripts, in general, takes comparatively long time.

Several methods have been proposed to address this issue mostly for non-Arabic languages. However, layout analysis and document classification techniques are in most cases language specific. Therefore, we can state the problem of this work as analyzing Arabic historical manuscripts to extract main content and constructing suitable representation for them. Then, characterize manuscripts using their words. Finally, given a keyword, locate all its instances in the archive of manuscripts using that representation. In addition, to facilitate this work an Arabic manuscript benchmarking database is constructed.

1.4 Contributions of the Dissertation

The following summarizes the outcomes of this dissertation.

A. Extends the literature survey of document layout analysis, and updates the research community with state-of-the-art analysis methodologies. we presented a comprehensive survey of DLA algorithms guided by a proposed general framework. It brings two major contributions to the research community:

   - A comprehensive DLA survey that presents a critical study of DLA algorithms on various analysis levels (i.e., regional, text line, etc.).

  - It discusses the complete pipeline of DLA framework that has been included in typical DLA algorithms such as preprocessing, analysis strategies, post-processing, and performance evaluation.

B. Hybrid document layout analysis approach for Arabic historical manuscripts.

  - A fast whitespace analysis for handwritten documents using an anisotropic diffusion filtering (ADF). Up to-our-knowledge whitespace analysis has not been employed in analyzing handwritten document layouts [25].
  - A hybrid technique that integrates global and local analysis to extract manuscript main content.

C. A learning-based keyword spotting algorithm with the following novelties:

  - A word-skeleton based keypoint sampling.

    Analyzing keyword's interior structure and locating keypoints using skeletonization. Unlike automatic detectors such as SIFT or SURF that may detect keypoints off-writing zones, word-skeleton based keypoint sampling selects keypoints on-writing zones.
  - Proposing a novel approach to estimate spotting thresholds based on modeling the SVMs matching, mismatching and rejection behavior.
  - Investigating the integration of skeleton, SURF, and dense keypoint sampling for keyword spotting and word recognition tasks.

D. Arabic Historical Handwritten Manuscript (AHHM) dataset. The AHHM extends existing datasets and add several required features as follows:

- AHHM dataset is collected from four manuscripts dated between 10^{th} to 18^{th} centuries.
- AHHM is multi-writer dataset: Manuscripts in AHHM dataset are written by four writers, which makes the dataset more realistic for information retrieval.
- Document layout analysis: Each page, in AHHM, has a main-content and several side-notes. Moreover, it offers manuscripts with different text-densities per page. Therefore, various document analysis algorithms can be evaluated using the proposed AHHM.
- Information retrieval: AHHM pages are characterized by a list of 25 keywords.

1.5 Dissertation Organization

The dissertation is organized as follows. In Chapter 2, a comprehensive literature review on document layout analysis and classification techniques is presented. In Chapter 3, we describe the AHHM database that includes, data sources, properties, and formats. In Chapter 4, a learning-free hybrid document layout analysis algorithm for Arabic historical manuscripts is outlined. Then, a novel document classification using keyword spotting approach is discussed in Chapter 5. Experimental results and discussions are given in Chapter 6. Finally, Chapter 7 concludes this dissertation.

CHAPTER 2

LITERATURE REVIEW

In this chapter, a comprehensive review of previous studies on document layout analysis and classification is presented. The review includes document preprocessing in Section 2.1. In Section 2.2, document layout analysis strategies are discussed. Document classification and retrieval methods are presented in Section 2.3. Standard performance evaluation metrics for document layout analysis and classification are given in Section 2.4. In Section 2.5, discussions and literature summaries are presented. Finally, Conclusions are presented in 2.6.

2.1 Document Preprocessing

Digitized documents may suffer from some degradations that negatively affects document layout analysis (DLA). These degradations have two main sources; native, and auxiliary [26]. The native degradations are permanent artifacts that are generated naturally due to aging, ink usage, writing style, etc. It leads to several issues such as text fading, ink bleeding, text show-through, touching components, irregular text-spacing or baseline fluctuation. Auxiliary degradations are due to external factors such as scanning device malfunction, lighting conditions, and document alignment. Some issues due to external noise are global document-skew, unbalanced document illumination, text blurring, etc. Therefore, preprocessing is an essential step of DLA algorithms. Although some studies did not discuss their preprocessing procedures in details, they assumed preprocessed document-images.

In this section, we discuss the essential pre-processing tasks of DLA; namely binarization and skew detection and correction.

2.1.1 Binarization methods for document images

In general, binarization converts grayscale images into binary images using pixel intensity thresholds. The produced binary image may contain a value of one for the background, and zero for the foreground (or vice versa). Most of the reviewed DLA algorithms require binarized document image. It reduces the image processing dimensionality to a single layered image and emphasizes document-elements' geometric structures.

Binarization can be categorized based on threshold estimation into four types; namely variance, entropy, contrast, and error-minimization thresholding. Ideal binarization computes a binary image of a clean document image with clear foreground/background content. In other words, it estimates the optimum grayscale thresholds for separating the foreground and background where their intra-class variance is minimal. Otsu [27], Niblack [28], and Sauvola [29] are examples of variance based binarization. Similar assumption of the existence of two main classes (i.e. foreground/background) is employed to compute the entropy-based threshold of the two classes that maximizes its sum in [30]. Due to outliers (i.e., noise) in pixel intensities, the performance of the variance or entropy based binarization methods may be degraded. Therefore, contrast-based binarization that finds midrange value between the lowest and the highest pixel intensities is used to estimate a binarization threshold [31]. Finally, error-minimization considers the binarization process as a classification problem of two classes foreground/background, where the target is to find a threshold that minimizes the classification error rate [32].

The application of binarization can be either global or local. The global binarization estimates the grayscale threshold from the whole image such as [27]. On the other hand, local binarization estimates a dynamic threshold that is changing according to the characteristics of the current processing position. A naive application of local binarization can be described as a process of dividing the image into blocks. Then, it computes the grayscale threshold and performs the conversion on each block individually. However, such application could fail especially if the blocks have different contrast values that vary notably from one block to another. This issue has been addressed using adaptive binarization [28, 29].

Adaptive binarization algorithms tend to estimate their thresholds for every pixel. Examples of adaptive binarization are [29, 33, 34, 35, 36]. Adaptive binarization may introduce some artifacts if the original image contains severe degradations. For instance, Gatos et. al in [36] observed that irregular illumination may lead to noisy artifacts in the binarized image. So, Gatos's method equalizes and smooths a document-image using image filtering to reduce illumination effects. Then, their algorithm estimates foreground pixels using Niblack method [28]. Finally, it computes the binary image using the following equation:

$$T(x,y) = \begin{cases} 1, & if\ B\left(x,y\right) - I\left(x,y\right) > d\left(B\left(x,y\right)\right) \\ \\ 0, & otherwise \end{cases}$$

where $T(x,y)$ is the binary image, $B(x,y)$ is the background image, $I(x,y)$ is the original image and $d(.)$ is the computed threshold, defined by:

$$d\left(B\left(x,y\right)\right) = q \times\ \delta \left(\frac{(1-P_2)}{\left(1+exp\left(\frac{(-4\times B(x,y))}{b(1-P_1)} + \frac{2\times(1+P_1)}{1-P_1}\right)\right)} + P_2 \right)$$

where δ is the average distance between the foreground and the background, q, p_1, and p_2 are free parameters that are set empirically to 0.6, 0.5 and 0.8 respectively; $b(.)$ is the average background value that is computed to boost foreground pixels separation. The free parameters are critical thresholds of the Gatos's method and require empirical estimation. Moreover, the foreground estimation using Niblack may produce mixed foreground/background mask due to dark foreground and background pixels. Similarly, Singh et. al in [37] reduced illumination variations of each image-block by image equalization before applying Otsu's method [27].

Dark foreground pixels that cause issues to binarize transition-pixels are addressed by Su et. al in [38]. Su's method computes binarization thresholds locally. The algorithm normalizes each image block $E(x, y)$ using Equation 2.1. Then, it computes statistical threshold based on blocks' transition-pixels, which is a combination of mean and standard deviation of image-blocks as in Equation 2.2.

$$E(x,y) = \frac{f_{max}(x,y) - f_{min}(x,y)}{f_{max}(x,y) + f_{min}(x,y) + \varepsilon} \tag{2.1}$$

where $f_{max}(x, y)$ and $f_{min}(x, y)$ are the maximum and minimum intensities of a block, and ε is a positive small value to avoid division by zero.

$$R(x,y) = \begin{cases} 1 & Ne \geq N_{min} \ and \ f(x,y) \leq (IE_{mean} + IE_{std})/2 \\ 0 & Otherwise \end{cases} \tag{2.2}$$

where Ne refers to the number of transition pixels found in the working block, N_{min} is a predefined threshold of a minimum number of transition pixels, IE_{mean} and IE_{std} are the mean and standard deviation computed by considering all blocks.

Although Su's method is robust in preserving text edges, it critically depends on two empirical parameters; the neighborhood-window size, and the N_{min} threshold.

In summary, binarization of a document image that suffers varying illumination and noise is a challenging task [39]. Even a method that integrates multiple techniques such as [36] or that performs image enhancement such as [38] may introduce noise artifacts in the resulting binary image. Moreover, deformations in text shapes such as fractures and merges can severely affect the threshold estimation. These challenging issues have called for dedicated binarization research [40].

2.1.2 Skew detection and correction

There are two main types of document skew, global and local. The global skew is formed because of a scanning process, where a scanned document might be tilted and thus the image will have a global skew angle. On the other hand, writing style may cause local skew. Previous studies have proposed various skew detection/correction techniques to address these two main types[41]. In general, Skew detection/correction methods can be categorized into seven classes based on the core procedure; projection-profile, Hough transform, nearest neighbor, cross-correlation, line fitting, frequency domain, and gradient.

1. **Projection-profile:** It enjoys ease of implementation and speed in the detection of text orientations [42]. Generally, it computes the sum of all pixel values along the horizontal direction(i.e., assuming text is written horizontally) to form a vertical profile histogram. Then, the histogram is analyzed to find peaks and valleys (see Figure 2.1). In a typical situation, the peaks represent text-lines, and valleys represent line-gaps. However, in real situations, dense text, touching text, and/or fluctuating text can confound the projection-profile

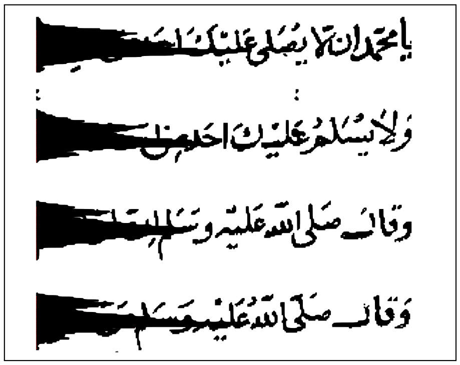


Figure 2.1: Projection profile (vertical projection) on normal text lines

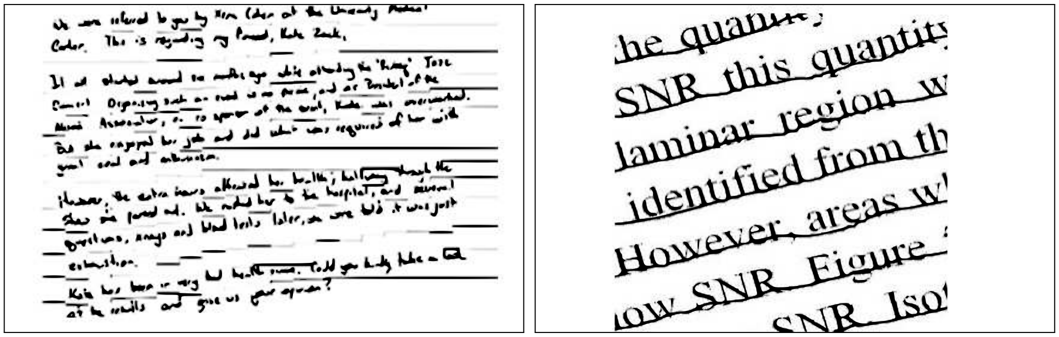
histogram by some false peaks and valleys.

For global skew detection, the projection profile technique computes several projections along multiple orientations in a range of $[\theta_s, \theta_e]$ where $\theta_s = 0$ and $\theta_e = \pi$. Then, a histogram of each projection is computed along horizontal direction. Finally, the projection with orientation θ_i that constitutes maximum variation indicates the skew angle of the document-image.

This method has been applied at the characters' level as in[45], and the connected components' level as [46, 47, 48] . The Application at fine levels requires high computational costs because it requires repetitive calculations of the projection. Although [49] has proposed a method to address this issue via reducing image size before performing the projection profile, their method loses accuracy in the trade-off computation cost.

In general, the projection profile skew detection/corrections methods may fail to find the true skew angle when large portions of a document image are not text [50]. This issue has been partially addressed by removing the non-text regions using image filtering before skew detection/correction. For instance, a wavelet transform has been used to remove non-text regions in [51].

Finally, to boost the projection profile approach against arbitrary layouts, some researchers proposed two types of local projection-profile(LPP); static and dynamic analysis [43, 44, 52]. In local static LPP, the algorithm divides a document image into fixed vertical stripes and applies the projection profile approach on each of them [43, 52]. However, It results in a staircase effect (see Figure



(a) Fixed stripes yields staircase effect [43]

(b) Dynamic strips [44])

Figure 2.2: Static versus dynamic local projection profile analysis

2.2.(a)), which confuses the final detection/correction of the skew angle. This issue has been addressed using dynamic striping LPP approach [44]. The stripes are defined with overlaps. Figure 2.2.(b) shows dynamic LPP results that allow tilted text-line analysis.

2. **Hough transform:** It was first introduced in [53]. The basic idea of Hough transform is to perform angular scanning of image pixels and accumulate votes of each scan in Hough space [54]. Then, a candidate line is detected by finding the highest response in the Hough space. Hough transform is used in detecting document skew for several years [55, 56, 57]

   Hough-based methods require preprocessed images that have enhanced line segment structures of text. Examples of preprocessing, smearing text-lines to improve their line structures [58], or reducing character components to single points [2] (see Figure 2.3).

   After image preprocessing, Hough space is generated by scanning a document image. Then, the Hough space is analyzed to find angles (θ) of the highest

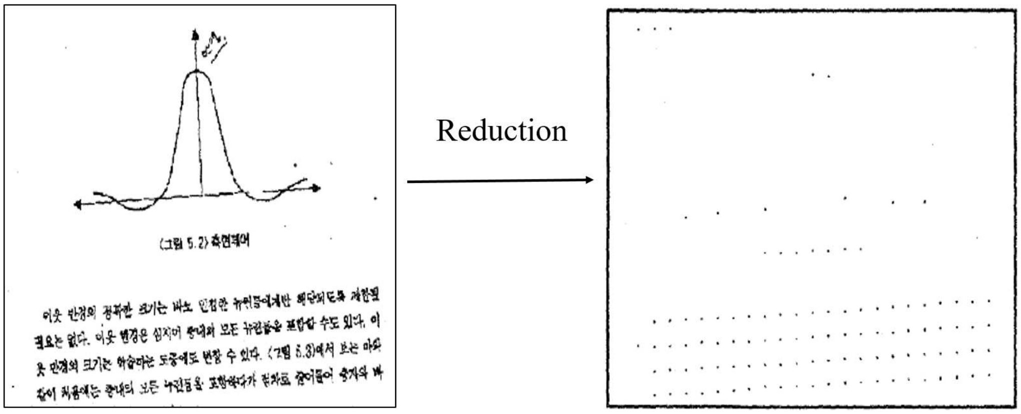


Figure 2.3: Example of data reduction before applying Hough transform [2]

Hough responses. The average of these angles describes the global document skew angle as in [56].

Other preprocessing examples to enhance line-structure of written text is applying Prewitt edge detection [54], reducing connected-components into their centroids as in [2, 59, 60, 61]. Figure 2.3 illustrates the reduction of connected components by their centroids. The text area shows continuous and regular dots along the horionztal/vertical directions, while figure components have larger vertical gaps. Although Hough-based techniques can detect skew angle accurately with small error (see Table 2.1), they require high computations to preprocess an image and to build the Hough space. Moreover, the line structure of handwritten text may be very challenging to be enhanced.

3. **Nearest neighbor approach:** The distance relationships of connected components can be utilized to detect and correct document skew angle. This technique divides a document image into small components, then finds all relative neighbors along specific directions. Then, it accumulates the angles of these

components in an angular histogram where the peak value is the skew angle of the document [62, 63]. Although the nearest neighbor approach is applicable to various document layouts, the resultant skew angle estimation may lack precision [63]. Like Hough-based approaches, handwritten documents could be very challenging. Therefore, most of the studies on nearest neighbor skew detection/correction are performed on printed documents such as [65, 66, 67, 68, 69].

4. **Cross-correlation:** cross-correlation method analyzes text lines to detect/correct a skew angle of document images [70]. Yan et. al in[70], suggested an algorithm that accumulates foreground pixels across pairs of inter-text line spaces. The algorithm moves horizontally and then vertically with two different fixed shifts d_h, and d_v respectively, to compute the cross-correlation as in the following equation:

$$R\left(d_v\right) = \sum_{x=0}^{(X-d_h-1)} \sum_{y=0}^{(Y-d_v-1)} I\left(x+d_h, y+d_v\right) \times I(x,y)$$

where $X-1$ and $Y-1$ are the horizontal and vertical lengths of the image respectively, and $I(x,y)$ is the image at location x, y. The maximum of $R(d_v)$ indicates an optimal d_v value of an average vertical gaps. Thus, the skew angle is computed by dividing the optimal vertical gap by the difference of two horizontal shifts Δd_h:

$$Skew = \tan^{-1}\left(\frac{d_v}{\Delta d_h}\right)$$

Yan's algorithm suits printed documents because it requires fixed inter-text line spacing. An improvement of Yan's algorithm that reduces the computational cost and addresses fixed inter-text spacing was introduced in [71]. The main idea is to apply the algorithm on selected equidistance vertical lines on the document-image. Therefore, the algorithm considers less image pixels to process. Generally, the cross-correlation methods are limited to document images of $\pm 15^o$ skew angles (see Table 2.1).

5. **Line fitting:** Like Hough-based approaches, line-fitting methods by nature do not need large data to describe a text line. For a line segment, only two points can describe a line segment. An example of a line-fitting algorithm that divides text into connected components and represents each connected component with a single point called Eigen-point is described in ([72]). It estimates document skew angle by a linear regression based on the coordinates of the Eigen points. It uses equations 2.3 and 2.4 to compute a document skew angle.

$$b=\frac{\left(n\sum_{i=1}^{n} x_i y_i - \left(\sum_{j=1}^{n} x_j\right)\cdot\left(\sum_{k=1}^{n} y_k\right)\right)}{\left(n\sum_{j=1}^{n} x_j^2 - \left(\sum_{j=1}^{n} x_j\right)^2\right)} \tag{2.3}$$

where b is the line slope, x, y are the Eigen-Points coordinates. The skew angle is computed using the following equation:

$$\theta=\tan^{-1} b \tag{2.4}$$

Although Eigen points are better than normal centroids to reduce non-convex shapes such as connected components, a single point representation of a connected component is not enough due to text-line fluctuation, and components descenders/ascenders variations. Therefore, Shivakumara et. al in [73] proposed a line-fitting approach using two point representation; uppermost and lowermost Eigen-points for better representation of the connected-components.

6. **Frequency domain:** Postl et. al [46] method is among the earliest algorithms that applied Fourier transform (FT) to detect/correct document image skew. A modified version of Postle's algorithm was proposed by Peake et. al in [74]. Peake's algorithm divides the document image into blocks and accumulates the block angle results in an angular histogram to detect a document image skew angle. Although Peake's method enhances the computation cost, it loses accuracy due to considering inconsistent FT responses from various blocks. This issue is addressed by normalizing FT local responses [3]. Lowther et al. in [3] analyzed the FT response using Radon transform. Figure 2.4 illustrates an example of a document image FT and Radon transform skew angle detection. In Figure 2.4.(c), the Radon transform shows several peaks due to minor responses

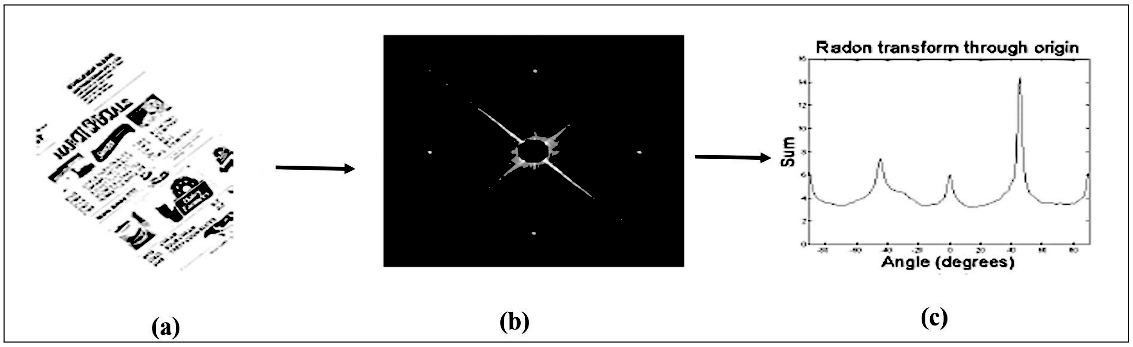


Figure 2.4: Skew detection in frequency domain analysis; a) original document image, b) Fourier transform magnitude, c) Radon transform [3]

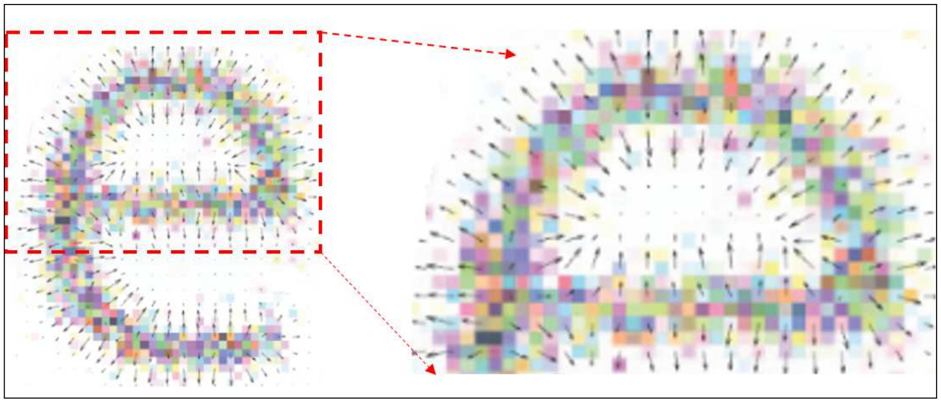


Figure 2.5: Confusion of gradient performed on letter ”e” [4]

on Fourier transform. Th minor Fourier responses were caused by irregularities in text, and drawings. This issue has been addressed by grouping similar contents using K-nearest neighbor algorithm and computing the convex hull of the grouped elements in [69]. So, Fourier transform was applied on the resultant convex shapes that represent document's blocks instead of applying it on the image text to detect document skewness.

Radon transform has been employed to find a skew angle of document image directly based on energy function in [75]. Radon transform is similar to projection profile approach which scans the document in multiple orientations. Consequently, it inherits projection-profile's limitations. For this reason, the method in [75] divides the document image into blocks and then uses a bootstrap aggregating (Bagging) to combine local skew angles found by Radon transform. Another study [76] suggested to remove non-text components using wavelet transform before applying Radon transform.

7. **Gradient approach:** Gradient methods require conversion of characters into edges or lines before carrying out document skew detection as in [77, 78]. Usu-

ally, text components have many edges and corners that represents writing strokes. Therefore, gradient may produce responses with many directions that could confuse document-images skew detection(see Figure 2.5). In Figure, 2.5, the gradient magnitude and angles are shown as black arrows. Therefore, direct application of gradient methods on full text may fail to determine the correct skew angle. Consequently, it requires character-based preprocessing to avoid false detection of skew angle.

Sauvola et. al [77] suggested to collapse character structures by reducing image resolution by a factor of 1/5. Then, the algorithm determines the document skew angle by finding the maximum peak of a histogram using the following equation.

$$A_{\theta}^{w} = \sum_{(i,j)\in W} G\left(i,j\right)\cos^{2}\left(\theta - \phi_{ij}\right)$$

where A is the accumulator histogram that indicates the maximum skew angle, θ is a rotational angle from 0 to 179, $G(i,j)$ and ϕ_{ij} are the magnitude and the phase of the gradient at position (i,j) respectively. A similar algorithm that fits a cubic polynomial over the angle histogram of the gradients is reported in [78]. Another different gradient method that addresses angle confusion due to direct application of gradient is described in [4]. Their method combined gradient algorithm with focused nearest neighbor clustering of interest points to estimate document skew.

Summary

Table 2.1 summarizes the discussed skew detection/correction techniques in this section. It shows the type, the language, and the number of documents used in the experiments. A comparison of these approaches is difficult and biased because the test documents are mostly different. However, the summary table gives an indication of the ability and the type of documents that can match future study setting.

To sum up, the projection profile, and frequency domain techniques are capable of detecting/correcting document skew at large angle correction range. However, they may be severely affected by non-text contents. Hough transform, line-fitting, nearest neighbor, and the gradient-based techniques have middle ranged angle correction abilities up to 180^o. However, they require clean document images. They can be easily miss-leaded by noisy artifacts. The cross-correlation techniques have the lowest angle correction range up to $\pm 15^o$.

Table 2.1: Skew detection and correction: a summary

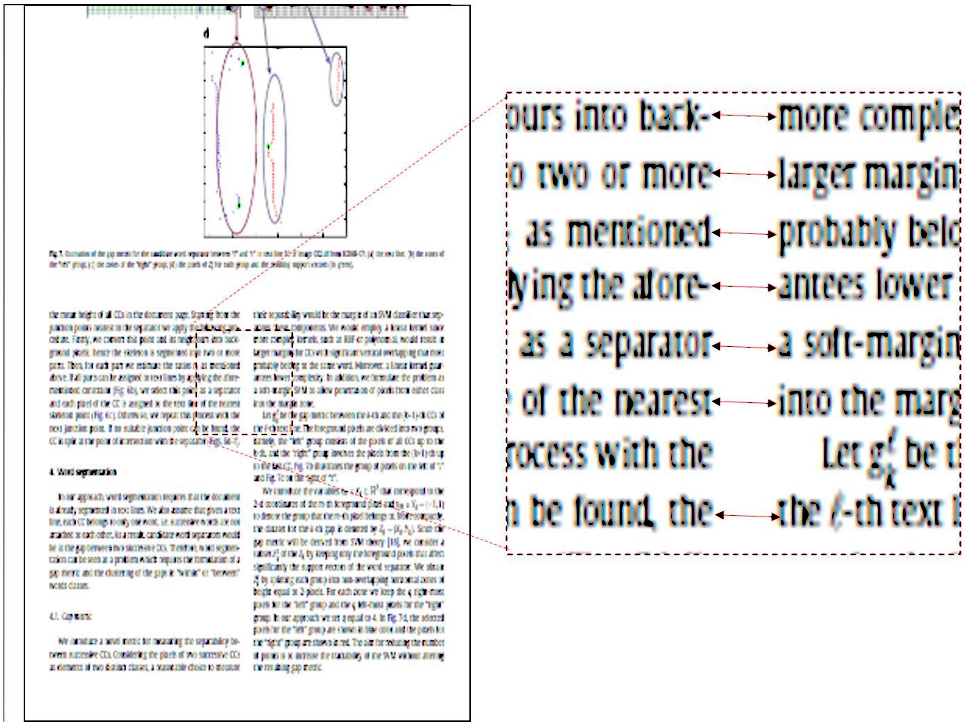
Method	Ref.	Angle		Document			
		Range	Error	Type	Test	Language.	Layout
PP	[44],2009	$\pm45^o$	0.2	H	30	Multi	MB
	[50],2007	$\leq25^o$	0.1	P	3	Multi	MB
	[51],2007	$\pm15^o$	0.2	P	500	Eng.	MC
	[43],2007	NA	0.12	H	720	Multi	MB
	[42],2005	$\leq360^o$	0.1	P	270	Eng.	MC
	[52],2001	NA	0.3	H	100	Arb.	MB
	[45],1997	NA	NA	P	460	Eng.	MB
	[48],1990	$\leq40^o$	0.1	MX	8	Multi	MB
	[49],1989	$\pm5^o$	NA	P	NA	Eng.	MC
	[47],1987	$\pm15^o$	0.05	P	NA	Eng.	MB
	[46],1986	$\pm45^o$	0.6	P	NA	Eng.	MB
HT	[79],2011	$\leq180^o$	0.3	P	500	Eng.	MC
	[54],2008	2^o-20^o	0.07	P	20	Eng.	MB
	[57],2008	2^o-150^o	0.05	P	300	Eng.	MB
	[61],2007	NA	NA	H	50	Eng.	MB
	[55],1996	$\leq180^o$	0.1	P	NA	Eng.	MB
	[56],1996	$\leq45^o$	0.84	P	100	Eng.	MB
	[60],1995	NA	NA	H	NA	Eng.	MB
	[59],1994	$\pm15^o$	0.167	P	250	Eng.	MC
	[58],1990	$\pm45^o$	NA	P	13	Eng.	MC
NN	[69],2014	NA	0.05	P	175	Eng.	MC
	[68],2008	$\leq40^o$	0.33	P	979	Eng.	MC
	[67],2005	$\leq180^o$	NA	P	30	Multi	MB
	[66],2003	$\pm45^o$	0.2	P	280	Multi	MC
	[65],2001	$\pm45^o$	-6	P	78	Eng.	MB
	[63],1993	$\leq180^o$	NA	P	NA	Eng.	MC
	[62],1986	$\pm90^o$	1.66	P	NA	Eng.	CX
CrsC	[71],1997	$\pm4^o$	0.068	P	NA	Eng.	MC
	[70],1993	$\leq12.5^o$	NA	P	2	Eng.	MC
LF	[73],2005	$\leq30^o$	0.5	P	100	Eng.	MB
	[72],2003	4.2^o,3.8^o	0.029	P	200	Eng.	MB
FD	[76],2013	1^o -25^o	NA	P	150	NA	MB
	[80],2007	$\leq120^o$	2	P	100	Arb.	MC
	[3],2002	$\pm45^o$	0.25	P	94	Eng.	MC
	[74],1997	NA	NA	P	32	Multi	MC
Grdnt	[4],2012	$\leq180^o$	1.75	H	658	Eng.	MB
	[78],1997	$\pm90^o$	NA	P	NA	Eng.	MC
	[77],1995	$\leq20^o$	0.1	P	11	Eng.	MC

- PP: Projection Profile, HT: Hough Transform, NN: Nearest Neighbor, CrsC: Cross-Correlation, LF: Line-Fitting, FD:Frequency Domain, Grdnt: Gradient-based.
- P: Printed, H:Handwritten, MX: mixed type documents, Eng: English, Arb: Arabic, MB: Manhattan based, MC: Multi-column document, NA: Not Applicable.

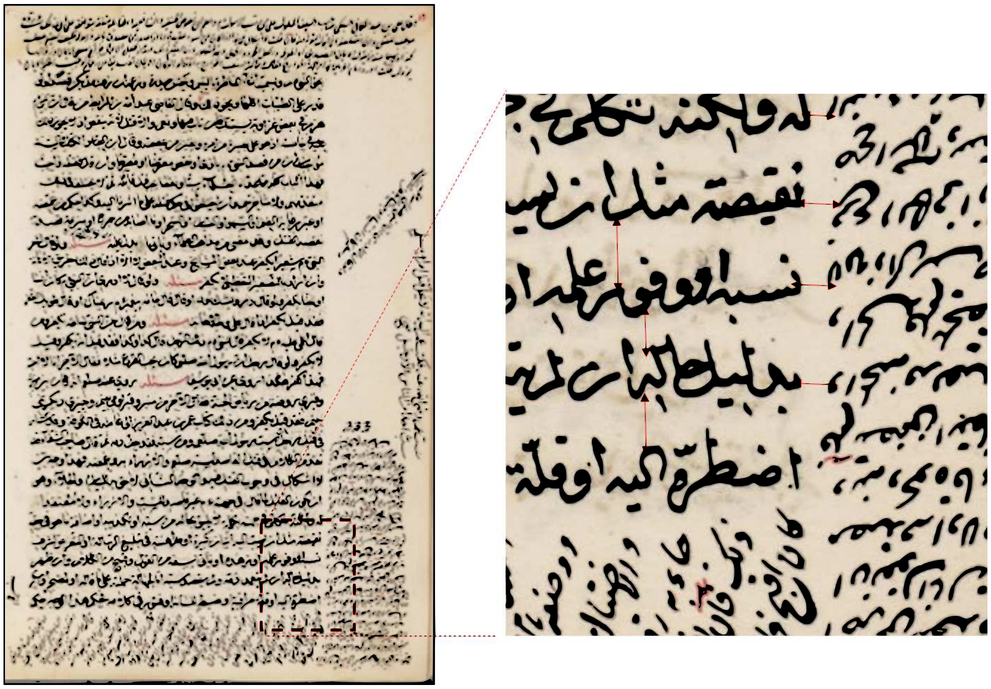
2.1.3 DLA Parameter Estimation

Most of the DLA algorithms, either bottom-up or top-down, need to set some parameters to guide the document analysis. There are two categories of a DLA parameter estimation: dynamic, and static.

In general, these parameters are critical thresholds that should be set carefully to perform robust DLA. The dynamic parameter estimation (i.e., data-driven) is computed from a document image directly. This type of estimation is used whenever the documents under analysis are heterogeneous. The parameters are dynamic because they change from one document to another. Examples of such methodology can be found in [81], [82], and [83]. On the other hand, static parameters can be determined at the beginning of document analysis. They remain fixed throughout the processed documents. Static parameters suit DLA algorithms that analyze constrained (i.e., structured) document layouts. Examples of static parameter estimation based on structured documents for known text-block locations are given in [84], regular region gaps [85], regular number of lines per region [86], and size of text components [6]. Figure 2.6 depicts two examples of regular, and irregular document layouts. Figure 2.6.(a) illustrates an example of possible static parameter estimation such as regular font size. Figure 2.6.(b) shows a situation where dynamic parameter estimation should be used because of variable writing styles.


ours into back- more comple
o two or more larger margin
as mentioned probably belo
lying the afore- antees lower
as a separator a soft-margin
of the nearest into the marg
rocess with the Let g_k^l be t
n be found, the the l-th text l

(a) Regular spacing, and writing styles



(b) Variable spacing, and writing styles

Figure 2.6: Comparison between contemporary and historical documents characteristics

2.2 Document Layout Analysis Strategies

In this section, three main DLA strategies are discussed, namely; bottom-up, top-down, and hybrid strategies. The bottom-up and top-down strategies are divided into five, and four classes respectively. Figure 2.7 shows a general taxonomy of the DLA strategies.

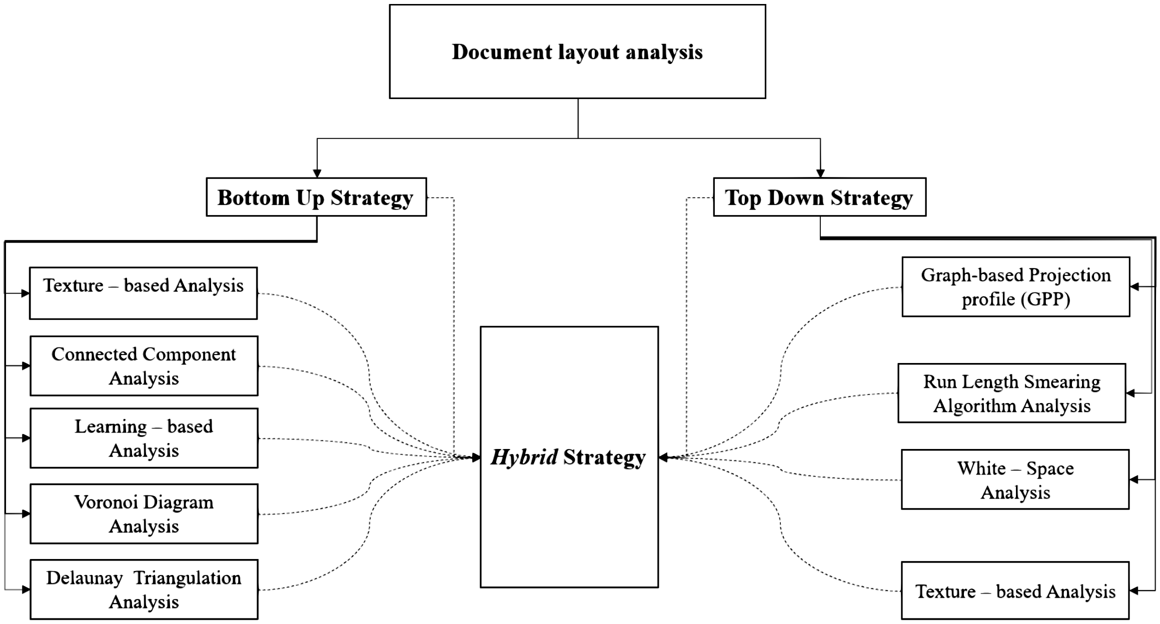


Figure 2.7: Document layout analysis strategies: main techniques

2.2.1 Bottom-Up Strategy

Bottom-up strategy is a data-driven methodology that may derive its parameters dynamically from the data. It estimates the parameters using statistics of pixel distributions, properties of connected components, words, text lines or regions. In general, bottom-up analysis starts at low levels of the image such as pixel, component, or word. Then, the analysis grows up to form larger document regions to match the intended analysis objectives. In this subsection, we discuss the bottom-up strategy based on five core categories; namely, connected component analysis, texture analysis, learning-based analysis, Voronoi diagrams, and Delaunay triangulation.

1. **Connected Component Analysis:**

   Connected component analysis allows flexible and robust layout analysis because it offers a wide range of shape properties. Docstrum algorithm is among the earliest successful bottom-up algorithms that adopted connected component analysis[63]. It groups connected components (CC) on a polar structure (distance and angle) to derive final segmentation. Even though Docstrum can cover a wide range of layouts, it was tested on printed documents. Similarly, connected component analysis supports n-ary tree analysis [15]. This algorithm assumes CC as vertices and the distances between them as weighted edges. Then, it uses minimum-cost spanning tree algorithm of [87] to analyze the document components.

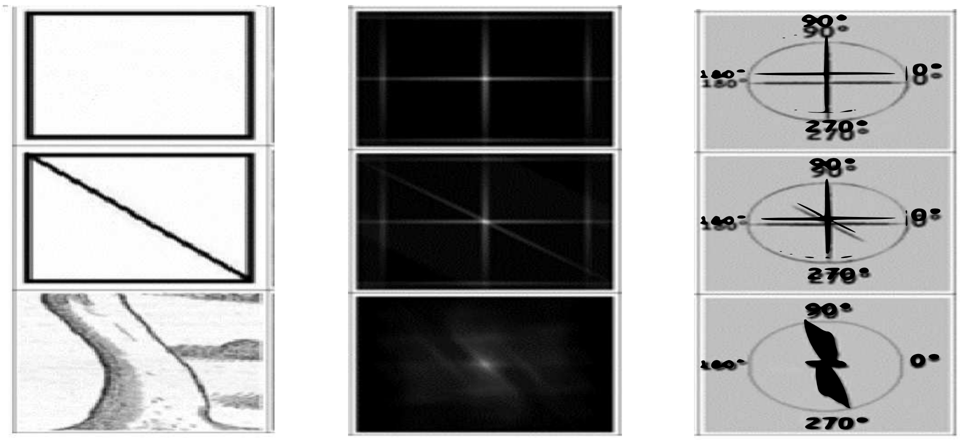
   Connected component analysis suites the extraction of degraded text lines as in [88]. Their algorithm boosts the analysis by evolution maps of connected components on grayscale and binary versions of a document image. Likewise, historical manuscripts suffer several issues such as degraded text, dense text, free writing style, aging, etc. The local features of connected components have helped to address some of these issues as in [89, 90]. In these two studies, both algorithms start by extracting sets of features from each CC. Then, part of these sets is fed to train an automatic multilayer perceptron (autoMLP) classifier for document analysis [91].

Another iterative classification algorithm that uses CCs to differentiate four classes of document's components as figures, separators, text, and noise is reported in [17]. In each iteration, the algorithm expels any found CC from a block that is considered heterogeneous to its neighbors. The CC is heterogeneous if it is non-text. This process is repeated until all detected regions are homogeneous and contain only text. For other non-text regions, the algorithm uses the connected component geometric properties to detect figures, separators, and noise.

2. **Texture Analysis:**

   Texture analysis enjoys speedy detection of image components. Texture analysis techniques can be categorized as bottom-up or top-down based on how it carries out the document layout analysis? Consequently, if the answer is merging smaller document elements to large regions, then it is bottom-up; the opposite is top-down.

   Bottom-up texture analysis starts by extracting texture features directly from image pixels. Then, these features are used to cluster pixels to form homogeneous regions or blocks. Spatial autocorrelation approach is one example of the bottom-up texture-based DLA [92]. This algorithm moves a window over the document image and represents each pixel with 20 feature points in a multi-resolution scale. It repetitively resizes the original document image to produce different image scales. Finally, the autocorrelation is analyzed in



(a) (b) (c)

Figure 2.8: Example of Rose of Directions[5],a) different images, b) autocorrelation response, c) rose of direction response.

rose-of-directions diagram (see Figure 2.8). This algorithm is computationally expensive due to the repetitive resizing of the complete document-image. This issue has been addressed by resizing the analysis window instead of using the complete image [93].

Text and graphic elements can be detected by analyzing rose-of-directions responses. In rose-of-directions diagram, line segments are highlighted by thin line response as shown in Figure 2.8.(c) (first and second rows). On the other hand, the response will be thicker in case of graphic element detection as shown in Figure 2.8.(c) (last row). This behavior of rose-of-directions have been utilized in document segmentation [5].

Due to it is successful application, the autocorrelation approach is compared to other texture analysis techniques such as Gray level Co-occurrences Matrix (GLCM), and Gabor filter bank in [94]. The study highlighted that Gabor filter is suitable for textual regions if a distinct font was used in all documents, while the autocorrelation is better if the document has complex layout or written in different fonts.

Texture analysis, which works on pixel level such as autocorrelation approach, is computationally expensive. Mehr et. al in [95] suggested a DLA based on superpixels. The superpixels is a group of pixels that shares similar spatial and intensity information. Although the superpixeling step is leveraging the separation between foreground and background and boosts DLA, it increases overall analysis time.

3. **Learning-based Analysis:**

   Learning-based approach can be used directly at any analysis level. However, the learning-based approach that labels small elements of a document-image such as characters or words to form larger regions can be considered as bottom-up.

   The, learning-based DLA approaches can be divided into two levels; pixel, and feature levels. Direct pixel values may not be a good choice to build a learning model in comparison to feature-based. This is due to the imbalance distribution of input data per class, and losing spatial information, etc. Often, textual pixels are dominant in document images in comparison to decoration pixels. Therefore, the trained model may be biased towards text-pixels more than decoration pixels. Lastly, pixels of periphery or decoration classes could have similar intensity values which confuse the learning process [96].

   Several studies proposed feature-based learning techniques to improve machine-model training. For example, Baechler et. al in [97] used dynamic MLP to train a learning model for DLA based on pixel values and context information. Similarly, Fischer et. al in [98] proposed pixel intensity and context information to generate a machine learning model for DLA. Both algorithms are computationally expensive as the algorithms perform model training at two different resolutions of each document image. Moreover, the context features may not be robust to identify the decoration-text class that appears within a text zone. Fortunately, more descriptive features was developed to improve learning-based analysis. For example, Gradient Shape Feature (GSF) which extracts geometric information from document components is proposed in [99]. The GSF features can describe different document components regardless of their context information. Another orientation-based shape descriptor using Scale Invariant Feature Transform (SIFT) is proposed in [5, 100, 101, 102]. These algorithms have reported the robustness of SIFT features to identify text verses decorative-text. Graz et. al in [102] found that SIFT interest points are scattered around text regions. Therefore, Garz's algorithm groups and validates interest points using Density-Based Spatial Clustering Application with Noise (DBSCAN) [103] to construct text lines based on their spatial information.

In summary, there are two important factors to build good trained model namely the availability of enough training data, and feature extraction. By nature document-images contain imbalance examples for each data-class. For example, text class have large number of examples than figure class. Hence, model training may be ended with bias convergence. The other factor depends on designing robust feature extraction methods. Many researchers compete to develop and design robust feature extraction methods for DLA. However, every feature extraction method has strengths and weaknesses. Therefore, an integration of existing feature extraction methods may lead to better performance such as [104, 105, 106].

4. **Voronoi-based Analysis:**

   Segmentation of arbitrary document-layout is very challenging. Arbitrary layout has no specific shape in general but can be surrounded with polygon shape. Fortunately, the Voronoi diagram is one solution that can define boundary points around arbitrary regions. It makes no assumptions about document layout shape and can describe the border points of various layouts as Kise's algorithm [107]. In Kise's algorithm, the Voronoi diagram is constructed based on CCs properties. Then, the analysis is conducted based on the selection of Voronoi edges that are characterized by two features; distance, and area ratio. However, the drawback of Kise's algorithm is defining Voronoi points based on connected components' centroids. This is because connected components are non-convex in general, which makes a single point representation inappropriate. This observation was addressed in [108, 109] by defining two points of each connected component. Both algorithms derive a neighborhood graph from the area Voronoi diagram, where each node indicates a document element. Finally, similar elements are merged together to form a document region.

In Voronoi analysis, every node may have large number of neighbors which can be counted by Voronoi edges. In other words, the large number of neighbors may lead to inaccurate region extraction. It appears vividly when a document image contains multi-size text. Consequently, the DLA performance will be degraded because Voronoi analysis will not be able to accurately find the proper segmentation cuts [109]. This issue has been addressed by integrating Docstrum algorithm and Voronoi diagram in [110]. Among the concerns of Voronoi analysis is the construction time of its diagram. It takes considerable amount of time to generate a Voronoi diagram. First, it reduces connected components into dots. Then, it has two passes to generate Voronoi points of document dots; 1) Voronoi points definition, 2) deleting all self Voronoi edges of each Voronoi point. These two passes of the Voronoi algorithm are computationally expensive especially for high-resolution document images [111].

5. **Delaunay Triangulation Analysis:**

   Unlike Voronoi diagram that defines large number of neighbors for each Voronoi node, the Delaunay triangulation reduces the number of neighbors to three. The Delaunay triangulation has been employed successfully to address text line segmentation in DLA [112]. Delaunay edge points ease the document segmentation rules as follows; **1)** the smallest edge-points represent text components on the same text line; **2)** the largest edge-points represent text components between contiguous text lines; **3)** triangles that have sides larger than some pre-computed thresholds represent text column regions or margin borders.

6. There are some miscellaneous methods that addressed bottom-up DLA. As an example, morphological analysis techniques are also called painting approaches. They manipulate document image pixels using a set of basic morphological operations such as dilation and erosion to group document components [113, 114]. Both studies targeted the extraction of text versus non-text regions. They assume that the shape structure is totally different from texture measurements. Therefore, their techniques compute statistical distribution of shapes as features for DLA analysis.

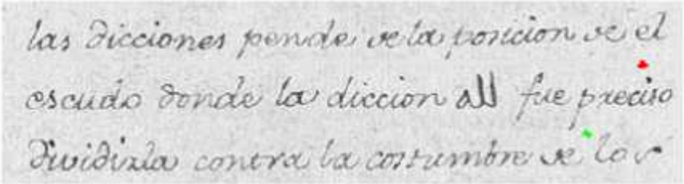
2.2.2 Top Down Strategy

In this section, the discussion is covering four top-down category techniques; Texture-based Analysis, Run Length Smearing Algorithm (RLSA), graph-based projection-profile (GPP), and White space analysis.

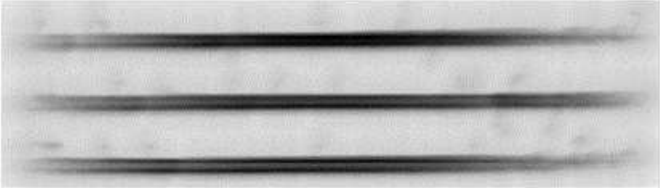
1. **Texture-based Analysis:**

   A direct example of a top-down texture analysis is highlighted in [81]. In this method, texture masks are generated by using neural networks to differentiate multiple document regions.

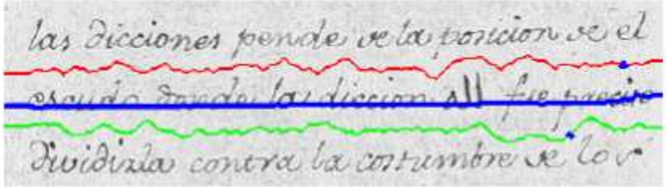
   Textual regions can be analyzed by an energy map to extract text lines via assuming that text regions are darker than background regions. In other words, the energy procedure produces low values for pure foreground or background areas while it is high at the border of both types (i.e., edges). Saabni et. al in [115] proposed energy map algorithm that produces strong seam lines between



(a) Original text



(b) Energy map response



(c) Seems detection(Red, Blue, and Green)

Figure 2.9: Energy map text line segmentation[6]

contiguous text-lines (see Figure 2.9). The drawback of Saabni's algorithm is the repetitive calculation of the energy map on the whole document-image to estimate seam-lines. An improved version of Saabni's algorithm (by the same research group) that requires no global re-computation of the energy map is proposed in [116]. This algorithm updates the energy map locally during text line detection. A detailed description and comparison of both algorithms are reported in [6]. Similar algorithm has analyzed the projection profiles of the energy maps that enhances the overall text line segmentation of historical manuscripts is reported in[117].

Image filtering can reveal strong document-image characteristics that can be utilized in DLA. A work that suggested six anisotropic Laplacian of Gaussian

(LoG) filters with one isotropic Gaussian, was suggested for document layout analysis in [118]. The response space is analyzed to detect its maximas along some orientations to find text regions. Moreover, K-nearest clustering algorithm was used to extract text lines based on the orientation feature. Similarly, another texture-based algorithm that coarsely locates main text regions of documents using Gabor filter is described in [1].

A multi-scale texture analysis is another dimension of top-down approaches. It reveals useful information about the document layout at different scales. This characteristic of multi-scale analysis can help in DLA of degraded documents. For instance, multi-scale analysis may allow tracing of high responses at each scale and map them back to original level to extract document regions as in [119]. The regions with back-traces end up as degraded text lines. Similar studies that analyze the behavior of multi-scale texture analysis are reported in [1, 120]. In [1], document images were filtered using Gabor filter to located different regions. Then, a minimization energy function was used to extract these regions. This technique is extended in [120] where Gabor filter was applied at different angles spanning the interval [0, 180] to detect curvy regions that was detected in [1].

2. **Run Length Smearing Algorithm (RLSA):**

   It is an operation that converts background pixels to foreground pixels if the number of background pixels between two consecutive foreground pixels is less than a predefined threshold (i.e., smears foreground pixels). RLSA was introduced first by Wahl et. al in [84] to conduct text-line analysis from structure of simple document layouts. A modification of the Wahl's algorithm that carried out smearing in two directions is described in [121]. The modified RLSA carried out horizontal scans from left-to-right and vice versa, and for vertical direction, it scans from top-to-bottom and vice versa. The RLSA performance may be degraded severely if a document-image contains multi-sized or curvy text [121, 25].

An adaptive RLSA that dynamically updates its thresholds based on local characteristics is suggested to address the multi-sized text analysis [25]. This algorithm performs two runs of RLSA on connected components; first, run is conducted to remove noisy obstacles and the second run is performed to detect text lines. Another RLSA algorithm that uses generalized adaptive local connectivity map (ALCM) to extract text lines is described in [122]. ALCM is highly sensitive to the height threshold which may lead to false text line extraction if it is improperly set.

The RLSA performance can be enhanced further by performing morphological operation to boost text structures before the application of RLSA [123]. In general, RLSA is a robust and simple to apply technique, but it requires careful estimation of its thresholds. Direct application of RLSA on handwritten documents may result in low performance due to heterogeneous writing styles[124].

3. **Graph-based Projection Profile (GPP):**

   It is a tree-structure based analysis that groups similar homogeneous content to form regions using splitting/merging operations. The X-Y cut algorithm is a well-known recursive DLA algorithm that analyzes horizontal and vertical projection profiles of a document image[85]. It converts every projection-profile response into a binary string and applies grammar rules for splitting/merging operations to find possible cuts. The X-Y cut algorithm suits structured document layouts that have fixed text regions and line spacing. An extension to X-Y cut algorithm that analyzed projection-profiles of connected components is suggested in [125]. The algorithm considers the layout analysis as a clustering problem where its members share the same spatial relationships. Another modification to the X-Y cut algorithm analyzes text regions using edit-cost evaluation metrics to enhance the segmentation results [126]. In general, the GPP algorithms are heavily depending on clean document images to find possible text region or line cuts [127].

4. **Whitespace Analysis:**

   It is used to detect regions that can be isolated by spaces (i.e., background) from all directions. It assumes that all foreground regions are separated from each other by some whitespace. It can be formulated as a maximization optimization problem to detect maximal white rectangles in each direction [14]. The whitespace analysis suites the segmentation of regular document layouts [128]. This algorithm detects globally column text regions and conducts coarse analysis to extract column-text lines. Whitespace analysis is integrated with the X-Y cut algorithm in [13]. It finds proper cuts of the whitespaces to form homogeneous regions. In summary, the whitespace analysis methods suits structured documents with clear whitespaces separation among their regions.

2.2.3 Hybrid Strategy

The hybrid strategy is the integration of both bottom-up and top-down strategies. Even though the research in bottom-up and top-down algorithms are well-established in DLA, there are still many challenging issues that neither bottom-up nor top-down algorithms separately can address.

Usually, the design of DLA technique predefines analysis objectives such as regional, text-lines, or words. These analysis objectives require some analysis parameters such as font size, average text-line gaps, or average word gaps. These parameters can be estimated using combination of both strategies. Moreover, each strategy has its own strengths and weakness. For instance, top-down algorithm using whitespace analysis can be considered fast technique in detecting document regions because the analysis focuses on background data (i.e., spaces). However, its analysis may lack segmentation precision especially if document-regions have irregular layout structures. On the other hand, connected component analysis is better at extracting region elements of various layouts. However, connected component analysis may extract elements of different regions due to the text-touching issue. So, based on these highlighted positive and negative aspects of whitespace and connected component analysis, a combination of both algorithms may boost their behaviors and solve complex document layouts. An example of this integration is suggested in [16]. The algorithm carries out slightly different whitespace analysis to approximate document regions. It computes weighted whitespaces vertically and horizontally to determine candidate regions. After that, each candidate region is analyzed using connected component analysis to determine document elements. Finally, a post-processing step is introduced to perfectly extract each region. Their post-processing involve human interaction for final segmentation. Another hybrid technique that integrates learning-based analysis, RLSA, and whitespace analysis is described in [129]. First, the algorithm detects text and non-text objects using neural networks. Then, it performs RLSA followed by whitespace analysis to determine regions' boundaries and extract text elements.

Unlike other hybrid approaches, rule-based heuristics of connected components can be considered a hybrid technique. The technique starts with a top-down view that uses knowledge-based rules to determine text by detecting document elements within minimum lengths [130]. Then, it analyzes connected components based on context and geometric characteristics to group them into text lines. Similarly, [131] proposed heuristic hybrid approach that detects text and non-text regions using minimum homogeneity algorithm. Then, it used connected component analysis to extract text-lines.

In general, hybrid techniques may provide robust analysis that can deal with arbitrary or complex document layouts. Even though studies such as [81, 129] claimed generalization of their algorithms to cover any document layouts, their experiments were conducted on a small population. Moreover, there are some algorithms that combines techniques of the same strategy such as the integration of two bottom-up algorithms as in [110], or top-down algorithms as in [13]. To sum up, hybrid strategy is rarely investigated in comparison to bottom-up or top-down strategies. The integration of methods may reveal robust algorithms for complex document layout analysis. Therefore, more efforts are needed to study hybrid techniques in the future.

Table 2.2: Document layout analysis algorithms: a summary

STGY	**Ref**	**Techniques**	**Documents**		
			Type	**Lang.**	**Layout**
BU	[95],2015	Texture analysis	P	French	MB
	[17],2015	CC Analysis	P	English	CPX
	[98],2014	Texture Analysis	P	English	MC
	[106],2014	Learning	MX	Lang+	MC
	[105],2014	Learning	MX	Lang+	MC
	[132],2013	Learning	MX	Lang+	MC
	[97],2013	Learning	P	Lang+	MC
	[94],2013	Texture analysis	P	French	MB
	[88],2013	CC Analysis	H	Hebrew	CPX
	[90],2012	CC Analysis	H	Arabic	CPX
	[102],2012	CC Analysis	P	German	MC
	[99],2011	Learning	P	English	MB
	[89],2010	CC Analysis	P	English	MC
	[100],2010	Learning	P	English	MB
	[93],2010	Texture Analysis	P	English	MB
	[133],2010	Voronoi, DOCSTRUM	MX	Lang+	MB
	[110],2009	Voronoi++, DOCSTRUM	MX	Lang+	MB
	[92],2008	Texture Analysis	P	English	MC
	[5],2005	Texture Analysis	P	French	MC
	[109],2004	Voronoi	P	English	MC
	[107],1998	Voronoi	P	Lang+	MC
	[134],1997	Voronoi	P	Lang+	MC
	[108],1995	Voronoi + CC	P	English	MB

STGY	Ref	Techniques	Documents		
			Type	Lang.	Layout
TD	[122],2015	RLSA	H	Lang+	MB
	[119],2014	Texture Analysis	MX	Lang+	MC
	[6],2014	Texture Analysis	H	Lang+	MC
	[117],2014	Texture Analysis	H	Lang+	MB
	[1],2014	Texture Analysis	H	Arabic	CPX
	[118],2013	Texture Analysis	H	Arabic	CPX
	[127],2011	Projection Profile	P	English	MC
	[115],2011	Texture analysis	H	Lang+	MB
	[116],2011	Texture analysis	H	Lang+	MB
	[123],2011	Texture analysis	H	Lang+	MB
	[25],2010	RLSA	P	Lang+	MC
	[135],2009	Texture analysis	H	Arabic	MB
	[124],2008	Whitespace analysis	P	English	MC
	[121],2004	RSLA	H	English	CPX
	[128],2003	Whitespace analysis	P	English	MC
	[81],1996	Texture Analysis	P	Lang+	MC
	[125],1995	Projection Profile	P	English	MC
	[126],1995	Projection Profile	P	English	MC
	[85],1984	Projection Profile	P	English	MB
	[84],1982	RLSA	P	English	MC
HY	[131],2016	Heuristics	P	English	MC
	[129],2014	CC, Learning, RLSA, White-space	MX	Lang+	MB
	[130],2008	Heuristics	H	English	MB
	[130],2008	Heuristics	H	English	MB
	[16],2007	CC, White-space, and user	P	NA	MB
	[96],2004	CC, RLSA, Learning	P	Arabic	MC
Other	[136],2009	Active contours	H	Lang+	MB
	[113],1991	Morphological Analysis	P	English	MB

- BU: Bottom-UP, TD: Top-Down, HY: Hybrid, P: Printed, H: Handwritten, MX: Mixed Documents, Lang+: Different languages used, NA: Not Applicable, CPX: Complex, MC: Multi-Column, MB: Manhattan-Based

2.3 Document classification

Keyword spotting approach have several applications such as document classification, categorization, and indexing. Keyword spotting (KWS) is a searching method that locates all instances of a query word. This technique has emerged in the last decades as an effective technique for large-scale document retrieval and classification. In particular, it is used to retrieve degraded and handwritten documents [19]. Therefore, these methods are considered as alternative approach to OCR methods [137].

In general, the keyword spotting approach retrieves document images by matching a user keyword query to some indexed template keywords in case of segmentation based spotting, and to document images in case of segmentation-free spotting. Therefore, in both cases, the keyword spotting approach does not convert document images into an editable standard text, but it searches directly using images.

KWS has several interesting research point-of-views which are contributing in KWS design as illustrated in Figure 2.10. Often, it is designed by considering four options; **1)** Target documents, such as structured/unstructured printed documents [138], or structured/unstructured handwritten documents [137]; **2)**Type of query, such as query-by-example (QBE) [139, 140], query-by-string (QBS) [141, 142], or hybrid query [20]; **3)** Analyzed documents, such as segmentation-based approach [143], or segmentation-free approach [144]; **4)** Matching strategy, such as static matching (i.e, fixed matching) [24] or dynamic matching [145]. Each of these main design options characterize the behavior of the spotting system.

KWS is usually confused with word recognition approach. To distinguish between

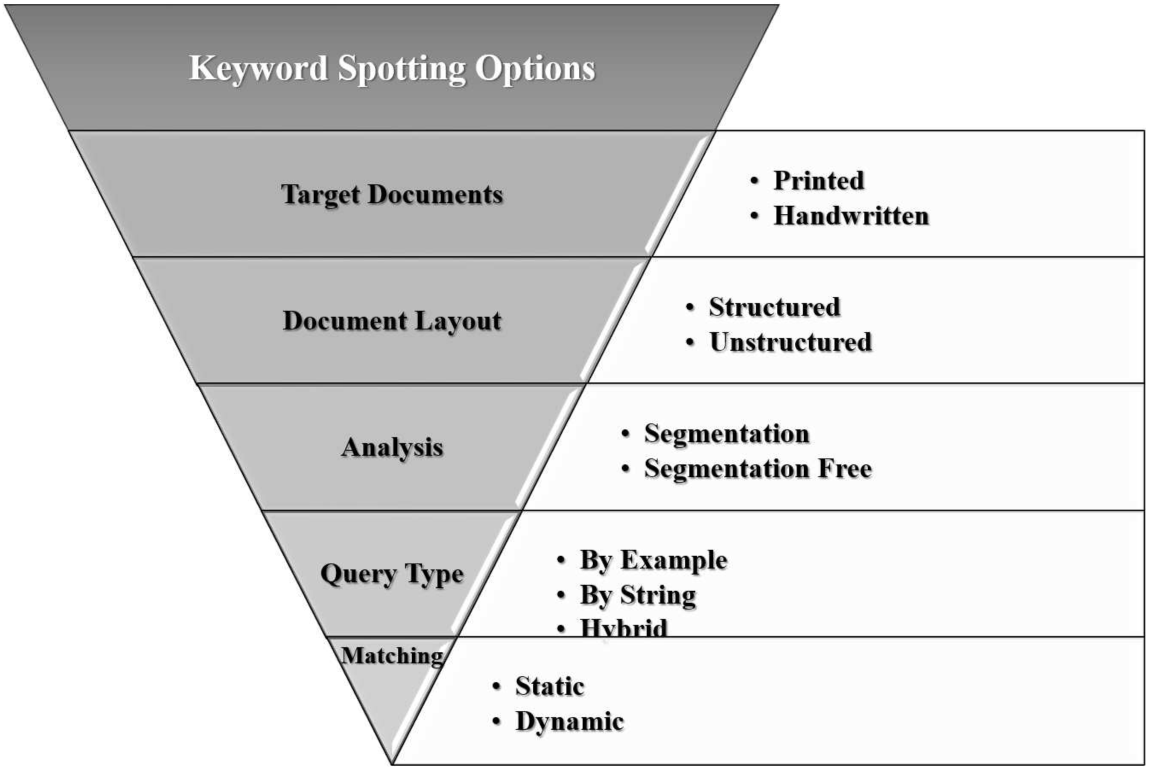


Figure 2.10: Keyword spotting systems: main design options

both tasks, KWS approach retrieves all matched word-instances to a given keyword-query image. On the other hand, word recognition task is meant to recognize the semantic information of a given word image. In other words, it produces transcription of recognized words using lexicon and language models [20].

Recently, two surveys have been published that cover the state-of-the-art in KWS [146, 11]. The interest in research of keyword spotting has increased over the past decade; from less than 15 proposed methods in 2007 to greater than 25 proposed methods in 2015 [11]. This indicates the importance of keyword spotting to address several applications such as document retrieval, on-line searching, automatic sorting of handwritten mails, figures identification etc. [11]. In general, it is common concern in all applications to develop robust feature extraction methods and improving keyword spotting performance. The state-of-art of feature extraction techniques can be categorized into three main categories as in [146]:

- **Geometric features:** It captures global geometric information of word structures. Word-image profile is an example of geometric features that can be matched using fixed metric such as Euclidean distance, or dynamic matching using Dynamic Time Warping (DTW) approach[147].
- **Shape features:** This category can be divided into three groups; **1)** Gradient, structural and concavity (GSC), **2)** Contour features, and **3)** Shape coding. GSC and Contour approaches compute word structural features as in [148, 149]. Then, word matching is carried out using the correlation between the keyword and the template structures. On the other hand, shape-coding approach encodes word strokes such as ascenders, descenders, diacritics etc. of both keyword query and templates [138, 150]. Then, similarity scores are computed using DTW to retrieve indexed documents [151].
- **Bag-of-Features (BoF) :** It is an active research area of keyword spotting because of it is successful application [22]. This method borrows bag-of-words model that is used in text document indexing to index images. In keyword spotting, the method builds bag-of-visual-words(BoVW) representation. It represents each keyword or template-word by histogram of visual-word features. Then, matching procedures can be used to compute similarity scores between keyword-queries and templates.

In the following subsections, we present a brief review of KWS and guide the discussion based on the types of queries. This is because query types are among the state-of-art KWS design concerns [20, 152]. A summary of reviewed methods is given in Table (2.3).

2.3.1 Query by example

In QBE, the KWS approach accepts a query image, then it finds matching word-instances in the document-image archive to that given query-image. In general, this approach requires no prior knowledge of a document's language, or type. So, it can be used to retrieve several document types. However, it is limited to in-vocabulary search. A user has to find at least one occurrence of his desired keyword query manually to initiate a keyword search and retrieve documents.

There are several examples of QBE in the literature, for instance, Quang et al. [139] proposed QBE approach using word-shape invariants. The invariants are some prototypes of clustered strokes that are determined by salient shape features. Even though Quang's method employed fast static matching, the method is language and writer oriented because the construction of shape-invariants is relying on the writing style and written language.

SIFT is used successfully in document retrieval applications [153]. In KWS, SIFT has been employed with an optimal detection and matching scenarios. For instance, Sudholt et al. [140] proposed to construct BoF framework by performing pre-matching of SIFT keypoints. First, their algorithm finds keypoints on a keyword-query image. Then, it matches query keypoints to template keypoints. After that, the locations of fully matched keypoints are chosen to form visual vocabulary of the BoF.

The method of Konidaris et. al in [22] avoided document segmentation by spotting possible keyword matches directly on the whole document. Their algorithm performs matching of keyword-query interest points to detected interest points on the whole document and determines possible candidate zones. Then, these document local zones are extracted and compared to the given keyword image.

The direct application of SIFT for document retrieval is computationally expensive due to processing large number of key points [154]. According to observations drawn in [155], methods that are depending on gradient features have shown good performance in trade off speed. The main reason behind this conclusion is the sensitivity of SIFT approach to noise, which misleads SIFT to detect false interest points [7]. Historical documents usually incorporate large background noise that may result in large number of false keypoints. Consequently, several studies have considered this issue by introducing other feature detectors to find interest regions of text such as blobs, lines, edges, and corners as in [155].

Various feature extraction methods have been introduced in the literature and compared to the performance of SIFT. For instance, shape based descriptors such as Fourier transform that was employed to describe keypoints detected by SIFT [155]. Examples of developing different feature extraction methods to describe detected interest regions are reported in [141, 156]. In [141], a bio-inspired method is described to detect interest zones. It uses Haar-like-features at different levels for word spotting. Histogram of Gradients (HOG), Exemplar Support Vector Machines(SVM), and

Fisher vector methods were integrated to produce better representation of the query in an unsupervised way [156].

Shape and SIFT methods may have an issue of dynamic matching because of detecting varying number of interest regions per keyword image. This issue impacts the performance of a KWS system in terms of matching accuracy and speed. Usually, DWT and dynamic matching algorithms are used to compute similarity scores between keywords and templates in varying-feature representation. Varying-representation of features is not desired in KWS because it requires additional KWS matching parameters. For instance, consider a KWS system that extracts 200 to 500 keypoints from template-images, and between 50 to 400 keypoints from keyword queries. Hence, the KWS system may need to set thresholds for rejecting those keyword queries that have few keypoints or may estimate a threshold for acceptance matching score.

Bag-of-features (BoF) has several advantages to the KWS systems, among the most important ones; **1)** it supports fixed-feature representation, **2)** it allows fast matching of a query image and template images, and **3)** it avoids direct local keypoint matching which is computationally expensive when dealing with large number of keypoints and word instances[144]. In general, most of BoF techniques share four steps, sampling, description, encoding and pooling. The sampling step is related to the size and number of keyword snippets that are required to extract visual words. Then, a description step is a feature extraction method to find better representations of the current keyword-image regions. Once the method extracted features from each keypoint, the BoF encodes them into visual words. Finally, the pooling step is a signature of the bag which is generated by accumulating the weight vectors of encoded visual words. These steps yield fixed length features for both keyword queries and templates. Therefore, static and fast matching can be carried out to spot keywords. Several studies have incorporated BoF framework using different feature extraction methods. For instance, SIFT features are the default choice for BoF [144, 157]. Another study that models word-images by using combination of BoF and Hidden Markov Model (HMM) in [23]. Their algorithm computes a histogram of quantized local descriptors to feed the BoF-HMM and generate keywords' models. However, the algorithm requires to construct a model for each keyword query, which can be considered as a drawback.

Recently, graph-based algorithms have emerged to address keyword spotting [11]. Wang et. al in [145] proposed a skeleton-based graphs with shape context as vertices. In this method, graph construction from skeleton and shape context is time-consuming step. Another method that suggested keypoint representation to construct tree graphs and speedup spotting graph-matching is reported in [158]. For an individual keyword, the algorithm constructs graph representation using key points as vertices and strokes as edges. Then, it employs Graph Edit Distance (GED) for graph matching. Moreover, the algorithm has improved by changing GED with bipartite matching in [159]. Furthermore, as word images can be represented by a combination of local features, a tree graph can be used to represent documents using word-images [160]. This algorithm treats a document image as a container of words. It constructs a tree graph for each document using local shape features that are extracted from the document words. Finally, matching is carried out using suffix trees to find matches.

2.3.2 Query by string

In QBS scenario, there are two types of query representation, pure-text query, or synthesized query. In pure-text query, a user initiates string query, and the KWS uses this string to spot matching zones on text-line images as in [21, 152, 161, 162]. On the other hand, in synthesized query, a user initiates keyword queries by type-in text, or choose from a list of predefined words. Then, a KWS system uses trained models for font, writing-style, background, foreground, and noise to build possible word-image instances of the typed-keyword [142, 163]. In fact, QBS in synthesized scenario is similar to QBE. Although QBS with synthesized queries allows flexible searching, it is challenging to generate keyword-query image instances in case of old documents [139]. Moreover, both scenarios have issues with searching out-of-vocabulary [11].

It is noted that QBS may borrow several features from QBE methods, as it converts the string query into an image prior to spotting process. In this sense, the keyword spotting allows text queries as in [141]. Lee et. al [162] proposed QBS based KWS for printed documents. They did not need to synthesize the keyword to convert text-queries to images because font types and sizes are known.

Finally, graph-based methods are also employed in QBS-based spotting systems. In graph-based QBS, text lines, paragraphs, and documents can be represented in cyclic weighted directed Word-Graphs (WG). The WG encodes words along with their corresponding probabilistic and segmentation information. Often, the WG is built using handwritten text recognition (HTR) system and employing standard Viterbi decoding as in [164]. An example of WG approach is described in [143].

In general, QBS has two scenarios; pure-text QBS and synthesized-based QBS. In pure-text scenario, QBS approaches employ recognizers such as HMM. These algorithms train a set of character HMMs using transcribed text-line images in off-line mode. Then, in operation mode, these algorithms accept text-line image and string query to produce matching scores [21]. Based on this score and pre-computed threshold, KWS labels the text-line image as positive or negative match. In synthesized-based scenario, the QBS algorithms add text-to-image synthesizing procedure. Then, the spotting is carried out similar to QBE.

2.3.3 Hybrid query type

Recently, QBE and QBS are combined to produce new feature extraction method called embedded features. The embedded features combine text attributes and image features. Therefore, such KWS allows string and image based queries simultaneously. Examples of such integration can be found in [20, 161]. Aldvert et al. [161] integrated textual and visual features for keyword spotting. The visual features are represented by using bag-of-features framework powered by gradient features, and textual features are formulated in terms of n-grams. Furthermore, Almazan et al. [20] algorithm accepts both string and image keyword queries. Then, it extracts attributes from the string-query and features from the image-query. After that, the algorithm embeds them into d-dimensional binary space called Pyramidal Histogram

Of Character (PHOC). PHOC is associated with a probabilistic model to indicate how likely a word-image that contains specific character sequence may appear. Shape encoding can be categorized as hybrid approach. For instance, Lu et. al [138] proposed a shape feature extraction that encodes a given image or string query using shape characteristics. This method suites clean printed documents as it is sensitive to shape deformations. Table 2.3 summarizes the reviewed keyword spotting techniques.

Table 2.3: Surveyed Keyword spotting methods

QTyp	**Ref.**	**Document**			**Query**			**Lang**
		Typ	Layout	SEG.	Qry	Mtch	Rslt	
QBE	[155],2016	H	UST	Yes	8	Dyn	mAP:77.18	Eng
	[158],2016	H	UST	No	10	Dyn	mAP:80.94	Eng
	[22],2016	P	ST	No	100	Dyn	mAP:83.6	Lang+
	[165],2016	P	ST	No	NA	STx	mAP:21.2	Eng
	[24],2015	H	ST	Yes	9	STx	mAP:70.41	Eng
	[166],2015	H	UST	Yes	32	Dyn	mAP:51.62	Eng
	[139],2015	H	ST/UST	Yes	38	STx	NA	Lang+
	[140],2015	H	UST	Yes	NA	STx	mAP:56.93	Eng
	[144],2015	both	UST	No	NA	STx	mAP:90.38	Eng
	[154],2015	H	UST	Yes	NA	STx	NA	Lang+
	[167],2014	H	UST	Yes	30	Dyn	mAP:62.02	Lang+
	[23],2014	H	UST	No	100	STx	mAP:30.1	Eng
	[168],2014	H	ST	Yes	NA	STx	mAP:63.34	Arb
	[145],2014	H	ST	Yes	NA	Dyn	mAP:17.5	Eng
	[169],2013	H	UST	Yes	14	Dyn	FS:96.04	Arb
	[156],2012	H	UST	Yes	NA	STX	mAP:58.5	Eng
	[170],2012	H	UST	Yes	NA	STx	mAP:84.00	Eng
	[162],2012	H	UST	Yes	NA	Dyn	Pr:71.5,$\Re$:46.9	Lang+
	[171],2009	P	UST	Yes	10	Dyn	mAP:81.5	Frn
	[172],2009	H	UST	Yes	NA	Dyn	mAP:79.14	Lang+
	[160],2009	P	ST	No	NA	Dyn	Pr:99.54	Eng
	[173],2008	P	ST	Yes	10	Dyn	mAP:71.7	Frn
	[174],2008	both	ST	Yes	NA	STX	Acc:77	Ltn
	[9],2007	H	UST	Yes	298	Dyn	Pr:74.2	Lang+
	[175],2003	H	UST	Yes	15	Dyn	mAP:71.56	Eng

⁻ QTyp: Query type, SEG: Segmentation, Typ: Type, H: Handwritten, P: Printed, ST: Structured Layout, UST: Unstructured Layout, Dyn: Dynamic, STX: Static, Pr: Precision, $\Re$: Recall, FS: $F_{measure}$ Rslt: Results, Qry.: Number of Queries, Mtch: Matching, Lang: Language, Eng: English, Arb: Arabic, Bng: Bangla, Lang+: different languages

QTyp	Ref.	Document			Query			Lang
		Typ	Layout	SEG	Qry	Mtch	Rslt	
QBE	[147],2003	H	UST	Yes	NA	Dyn	mAP:67.92	Eng
	[137],1996	H	UST	Yes	1	Dyn	NA	Eng
QBS	[152],2017	H	UST	Yes	NA	Dyn	mAP:73.12	Bng
	[143],2015	H	UST	Yes	NA	STx	mAP:71.15	Eng
	[176],2015	H	UST	No	NA	Dyn	mAP:76.5	Eng
	[141],2015	H	ST	No	NA	Dyn	Pr:77.5	Eng
	[21],2012	H	ST	No	NA	STx	mAP:88.15	Eng
	[142],2007	H	UST	Yes	25	STX	NA	Ltn
	[163],2006	P	ST	Yes	NA	Dyn	Pr:100	Lang+
HYB	[150],2016	H	UST	Yes	NA	STx	mAP:86.49	Eng
	[20],2014	both	UST	Yes	NA	STx	mAP:93.93	lang+
	[161],2013	H	ST	Yes	1090	STx	mAP:76.2	Eng
	[138],2008	P	ST	Yes	137	STX	FS:92.51	Eng

$^-$ QTyp: Query type, SEG: Segmentation, HYB: Hybrid, Typ: Type, H: Handwritten, P:Printed, ST:Structured Layout, UST:Unstructured Layout, Dyn: Dynamic, STX:Static, Pr: Precision, FS: $F_{measure}$, Rslt: Results, Qry.: Number of Queries, Mtch: Matching ,Lang: Language, Eng:English, Bng:Bangla, Ltn:Latin, lang+: different languages

2.4 Performance Evaluation

In this section, we discuss the experimental settings and the evaluation metrics that have been used to measure the performance of document layout analysis and spotting algorithms. In general, the evaluation process of document layout analysis and spotting algorithms consists of two aspects; used datasets and evaluation metrics.

2.4.1 Datasets

There are several datasets that can be used to evaluate document layout analysis and spotting techniques. Table 2.4 lists benchmark datasets with their statistics. In general, there are three types of document datasets based on document type; printed, handwritten, and mixed (printed and handwritten) documents.

Printed datasets are generally developed to test contemporary document analysis and classification methods. An example of such dataset is developed by the University of Washington (UW-3) [177]. It consists of 1600 skew corrected technical English articles. The ground-truth was manually defined by bounding box-coordinates for text, and non-text zones. Each region is labeled by either text, math, table, or figure. The dataset is suitable for page layout analysis of technical articles that targets text and non-text objects. Another common dataset is published by Pattern Recognition & Image Analysis (PRImA) Research Lab [178]. PRImA is considered a realistic general document type dataset. The dataset was created essentially for the evaluation of layout analysis methods that are targeting modern document styles of every day use such as scanned memos, letters etc. It consists of 305 pages from various sources with emphasis on technical publications and magazines. Recently, collaboration between Boston University, Cairo University, and Electronics Research Institute developed BCE-Arabic dataset [179]. The BCE-Arabic consists of 1833 printed pages collected from 180 books. The ground-truth is generated manually using several tools such as Pixlabeler [180], Groundtruthing Environment for Document Images (GEDI) [181], and "Document, Image, and Video Analysis, Document Image Analysis" (DIVADIA) [182]. The dataset suites various analysis objectives such as text only analysis (1235 pages), text vs. images (383 pages), text vs. graphic elements (179 pages), text vs. tables (24 pages), single or double column text vs. images (29 pages). Similarly, A large dataset is developed under the project of IMProving ACcess to Text(IMPACT) [183]. It contains 70,000 historical printed pages of 17^{th} - 20^{th} centuries. The data were collected from various books, newspapers, journals, and legal documents in 17 languages.

There are few handwritten-document datasets which are published for layout analysis. For instance, The famous George Washington papers (GW20) dataset for keyword spotting techniques consists of 20 pages segmented into text lines, words, and word classes[184, 185]. Saint Gall [186] dataset consists of 60 pages from a medieval manuscript written in Latin by single writer. Saint Gall dataset is divided into 20 pages for training, 30 pages for testing, and 10 are selected for validation. The Saint Gall dataset is mainly used for keyword spotting and text line analysis. Parzival dataset [186] consists of 47 pages written by three writers is developed for the same purpose. The Parzival's pages are written in German in the 13^{th} century and contains the epic poem Parzival by Wolfram Von Eschenbach. The dataset is divided into 24 pages for training, 14 pages for testing, and two pages for validation. The ground truth of both datasets (Saint Gall and Parzival) are extracted using DIVA-DIA software. Another handwritten dataset called CENIP-UCCP was collected by the Center of Image Processing Urdu Corpus Construction Project (CENIP-UCCP) [187]. It contains 400 text pages written by 200 writers in Urdu. The ground-truth is provided at the text-lines level to evaluate text-line analysis and segmentation-free keyword spotting. Finally, a recent Arabic dataset (HADARA80P) is published for keyword spotting[188]. It contains 80 pages extracted from a single book. The dataset provides main content, text line, and word level ground truth.

Even though analyzing mixed documents of handwritten and printed content is seldom in the literature, a mixed dataset called LMP is designed for this purpose. The LMP is developed by the laboratory for Language and Media Processing (LMP) [189]. It contains 203 pages; 109 in Arabic and 94 in English. It could be used for analyzing complex document layouts to extract text, images, or signatures.

Several other subsets of datasets can be found within the ICDAR Page Segmentation Competitions 2003-2011 that suites different analysis objectives [190].

Table 2.4: Document layout analysis and spotting datasets

DB	**Age**	**Data Statistics**				**Ground Truth**			**Type**	**Pub**
		Wrts	**Typ**	**Doc.**	**Lang.**	**TL**	**CC**	**BB**		
BCE-Arabic-v1	CN	NA	P	1833	Arb.	No	No	Yes	LA	[179],2016
HADARA80P	HS	1	H	80	Arb	Yes	Yes	Yes	LA/SP	[188],2014
UW-3	CN	NA	P	1600	Eng.	Yes	Yes	Yes	LA	[177],2013
CENIP-UCCP	CN	200	H	400	Urdu	Yes	No	No	LA	[187],2012
LMP	CN	NA	MX	203	Lang+	No	No	Yes	LA	[189],2010
PRImA	CN	NA	P	305	Eng.	Yes	Yes	Yes	LA/SP	[178],2009
Parzival	HS	3	P	47	Gmn	Yes	No	No	LA/SP	[186],2009
GW20	HS	1	H	20	Eng.	Yes	Yes	Yes	LA/SP	[185],2007
Saint Gall	HS	1	P	60	Latin	Yes	No	No	LA/SP	[186],2006
IMPACT	HS	NA	P	7K	Lang+	No	No	Yes	LA	[183],2000

- CN: Contemporary, HS: Historical, P:Printed, H: Handwritten, LA: Layout Analysis, Sp: Spotting NA: Not Applicable, Lang+: different languages, Pub: Publishing Year

2.4.2 Evaluation Metrics

Keyword spotting systems

In some cases, a formal evaluation metric called $F_{measure}$ which is commonly used in information retrieval (i.e., keyword spotting) is used in DLA to evaluate its performance (see Table 2.5). The $F_{measure}$ combines the precision and recall performance values. It shows how precise is the retrieved results to recall correct elements out of all components. The $F_{measure}$ is computed according to the following Equation:

$$F_{measure} = 2 \times \frac{\Pr \times \Re}{\Pr + \Re} \tag{2.5}$$

where Pr is the precision of results, and it is computed as in Equation (2.6), and $\Re$ is the recall of results that is computed as in Equation (2.7).

$$\Pr = \frac{TP}{TP + FP} \tag{2.6}$$

where TP is the true positives that represent the correctly retrieved information, FP is the false positives, which represents the incorrect retrieved information.

$$R = \frac{TP}{TP + FN} \tag{2.7}$$

where FN is the false negatives that represent the incorrectly rejected information.

Document layout analysis

It is observed from the reviewed algorithms in Section 2.2 (especially the studies published before 2000) that most of them are evaluated subjectively. This could be attributed to the lack of benchmarking datasets [130] and unestablished segmentation evaluation metrics [81, 128]. A simple way to evaluate the performance of DLA algorithm is via counting the ratio of correct segmented elements to all elements (i.e., accuracy rate). Element counting could be based on connected components as in [135], words [109], or text-lines as in [102, 116, 124]. Similarly, the evaluation via counting correct segmented elements is named recognition rate as in [6, 89, 114]. There are efforts to standardize the DLA performance evaluation metrics by introducing DLA evaluation frameworks [191, 192, 195] . In this subsection, we describe three DLA performance evaluation frameworks.

A. **Framework 1:**

In the competition of document layout segmentation [191], two-level evaluation scenarios were considered; pixel or region levels. In the pixel level evaluation, segmented pixels and ground-truth pixels are matched to compute the performance score using the following equation:

$$MS\left(j\right) = \frac{\left(T\left(G_j \cap R_j \cap I\right)\right)}{\left(T\left(\left(G_j \cup R_j\right) \cap I\right)\right)} \tag{2.8}$$

where MS is the matching score of a single element, T is a function to count the matches of ground-truth region G_j to segmented region R_j and I is image pixels.

On the second hand, region-based metric evaluates the segmentation results based on the number of region matches. The performance is translated to an $F_{measure}$ using detection rate $(DR) \approx \Pr$ and recognition rate $(RR) \approx R$ which are computed using equations (2.9), and (2.10):

$$DR = \frac{o2o}{N} \tag{2.9}$$

$$RR = \frac{o2o}{M} \tag{2.10}$$

where $o2o$ is the number of one-to-one region matches, N is the count of ground truth regions, and M is the count of the segmented regions. Then, the system computes $F_{measure}$ as in Equation (2.5).

B. **Framework 2:**

The performance of document layout analysis technique can be evaluated using weighted bipartite graph (i.e., pixel-correspondence graph) [192]. In this method, each pixel either a segmented or a ground-truth element is considered as a node. The edges are established only if there is an overlap between the segmented and the ground-truth components over some pixels. Hence, a perfect matching is declared if the segmented pixels overlap the ground-truth, so there is exactly one edge incident to each pixel. Using edge counts with their significance, four types of matching errors can be identified:

- over-segmentation: more than one significant edge from ground-truth joins segmented pixels.
- under-segmentation: more than one significant edge from segmented pixels joins ground-truth.
- missed components: a number of ground-truth foreground pixels that matches the background of segmented pixels.
- false alarms: a number of segmented foreground pixels that matches the background or noise of ground-truth pixels.

These measurements are the metrics used to compare different DLA systems.

C. **Framework 3:**

Combination between pixel and region based evaluation is commonly used in the DLA evaluation [193, 194, 178]. Usually, there are two main issues in the evaluation; correspondence, and type of comparison. In this framework, the correspondence is based on segmented regions against ground-truth. Then, a regional DLA score can be computed based on region foreground-pixels.

A segmentation result is considered successful if it completely overlaps only one ground truth region. Therefore, errors are characterized based on PRImA framework [195] to five types:

- Merge: A segmented region that overlaps more than one ground truth region
- Split: A ground truth region is overlapped by more than one segmentation region.
- Miss: A ground truth region that does not overlap any segmented region.
- Partial Miss (PMiss): A ground truth region that partially overlaps a segmented region.
- False detection: A segmented region that overlaps no ground truth region.

In order to calculate the success rate of the segmentation SR, each error type ER is first multiplied by the affected foreground pixels of the merged, missed, partially missed, split, or falsely detected regions. Then, these error rates ER

are used in calculating the success rate (Eq. 2.11) as described in [193]:

$$SR = \frac{\sum_{i=1}^{N} \omega_i}{\sum_{i=1}^{N} \frac{\omega_i}{1-ER_i}} \tag{2.11}$$

where N is the number of error types, and ω_i is the final weights that are calculated of each error type as:

$$\omega_i = \frac{(N-1)ER_i + 1}{N} \tag{2.12}$$

Pixel-based evaluation metric is an aggressive and rigid method to evaluate DLA performance. In addition, it may not be suitable to evaluate large regions of historical documents that requires error-tolerance of minor segmentation misses. In other words, the segmentation of document images with severe degradation due to aging, writer-style, lighting conditions, or ink-usage, should have some level of flexibility [196]. Table 2.5 lists examples of DLA quantitative results.

Table 2.5: Examples of DLA evaluation metrics & results

Metric	**Ref**	**Results (%)**
$F_{measure}$	[131],2016	94.58
	[95],2015	75.00
	[17],2016	96.80
	[6],2014	98.00
	[117],2014	99.97
	[1],2014	98.84
	[119],2014	99.69
	[118],2013	92.95
	[94],2013	74.00
	[197],2012	95.20
	[93],2010	96.80
	[25],2010	84.80
	[136],2009	96.30
Recognition Rate	[95],2015	96.00
	[98],2014	96.80
	[105],2014	97.90
	[88],2013	88.23
	[97],2013	96.3
	[132],2013	97.47
	[102],2012	98.65
	[116],2011	98.00
	[115],2011	98.90
	[123],2011	76.00
	[99],2011	63.00
	[114],2011	99.35
	[89],2010	95.72
	[100],2010	91.35
	[135],2009	99.50
	[124],2008	98.40
	[92],2008	87.50
	[96],2004	98.43
	[109],2004	99.05
	[121],2004	93.00

Metric	Ref	Results(%)
Subjective	[130],2008	NA
	[128],2003	NA
	[107],1998	NA
	[134],1997	NA
	[81],1996	NA
	[108],1995	NA
	[126],1995	NA
	[113],1991	NA
	[85],1984	NA
	[84],1982	NA
Other	[122],2015	98.66
	[106],2014	5.49
	[129],2014	57.80
	[127],2011	7.50
	[133],2010	70.78
	[110],2009	78.30
	[123],2011	98.00
	[16],2007	86.30
	[5],2005	95.00
	[125],1995	0.37

- Other: such as Recognition Error Rates, Detection Rate, Precision, Recall, Jaccard index.

2.5 Discussions

2.5.1 Document layout analysis

Bottom-up techniques are dominant in the literature according to the reviewed methods (see Figure 2.12.(a)). Regardless of the required space and time complexity of bottom-up strategy, its positive characteristics have attracted researchers to develop more bottom-up methods. Its most important characteristic is the dynamic behavior. Around 53% of the techniques according to our literature survey are bottom-up.

In general, the basic building block of bottom-up techniques is the connected components. Many studies have proposed their techniques using connected components instead of pixels because they are a group of related pixels with defined connectivity. They are presented in almost all bottom-up techniques such as learning-based [90], texture-based [88], SIFT [102], Voronoi [133] and Delaunay [199]. Figure 2.11 illustrates the relationship between these techniques.

The top-down strategy may work on less amount of data in comparison to bottom-up strategy. For instance, the white space analysis locates spaces surrounding regions while bottom-up technique may require components of text, non-text etc. Usually, the top-down strategy works perfectly on Manhattan layouts, however requires clean and skew corrected document images as in [14, 13, 124]. Yet, top-down strategy techniques are important for DLA because most of the contemporary documents layouts are Manhattan-based. The hybrid strategy has not been studied enough, based on our reviewed literature (see Figure 2.12) around 8% of the population are hybrid techniques. This reflects a need for studying hybrid analysis to address complex document layouts.

In general, there are three main observations on DLA literature; universal DLA, preprocessing phase and document languages. Some studies claim that their techniques are applicable to all types of document layouts, a claim that was not proved. Since most of these studies were conducted on printed/typed document images which did not include all ranges of document layouts. Therefore, a one-fits-all DLA solution is not developed yet.

Secondly, the importance of the preprocessing phase including binarization, noise removal, and de-skewing is not negligible. Several studies did not detail their preprocessing phase and assumed input document images are preprocessed. Around 45% of reviewed algorithms require binarization and de-skewing of a document image before DLA.

Thirdly, the analysis might be different from one language to another. For example,

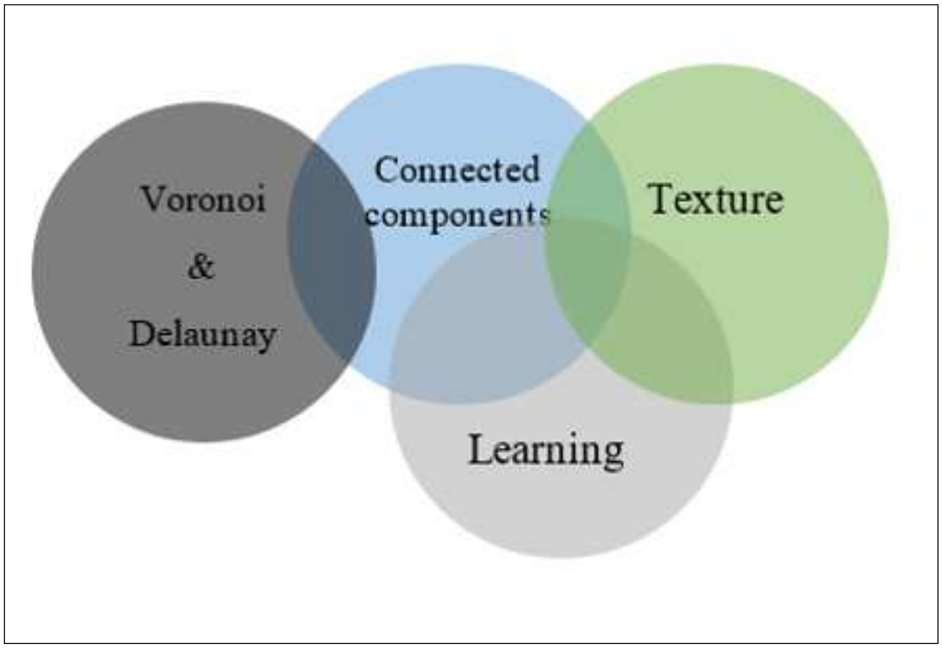
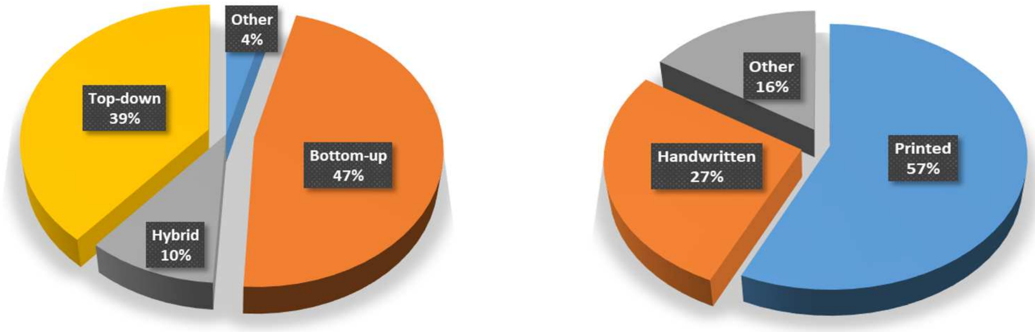
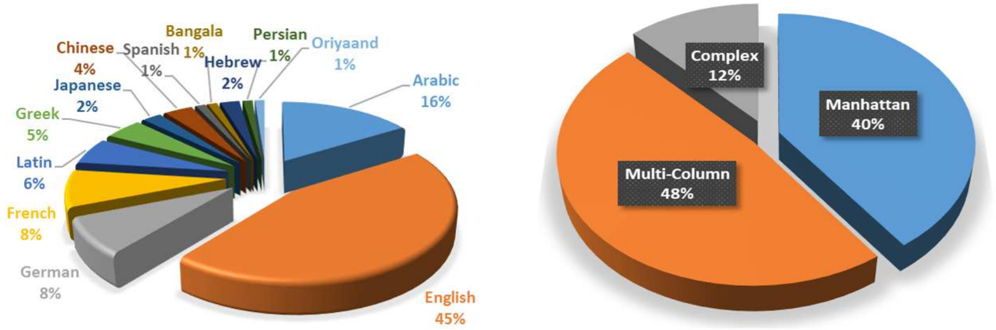


Figure 2.11: Bottom-up techniques relationship set-view



(a) Strategy based distribution (b) Document Type Distribution



(c) Language based Distribution (d) Document Layout Distribution

Figure 2.12: Document layout analysis techniques statistics

a printed/typed English document can be broken down into columns, paragraphs, text lines, words, connected components, or characters, while Arabic can be broken down to the level of part of an Arabic words (PAW) which is, in general, more than one character. In Arabic language, it would be very challenging to analyze at the level of characters due to several reasons; Arabic alphabet has up to four shapes for a letter based on its position in the word (beginning, middle, last, isolated), the writing is cursive in typed and handwritten, it could have irregular spacing among PAWs, and may have letter elongation. Therefore, some languages may need to be treated differently in DLA.

Table 2.2 summarizes the literature review on DLA. Most of the studies addressed contemporary documents. This is reasonable because modern life requires computerized offices, libraries, etc. So, documents are often digitized and because of their huge amount, retrieving them back would need software facilities such as keyword spotting. In recent years, the studies of DLA on handwritten documents are limited due to the degradations in writing style. Currently, most of the documents are not handwritten. Examples of the remaining handwritten sources are personal letters, notebooks, and bank cheques. It is good to note that the handwritten documents listed in Table 2.2 are either historical documents or mixed documents. Second observation, most of the work were conducted on English language documents which may be attributed to the availability of benchmark datasets and/or the easy access to repositories of scientific journals which are mostly written in English. This explains further why most of the research have conducted experiments to analyze either multi-column or Manhattan layouts 57% and 35% respectively of the reviewed studies. Figure 2.12 summarizes the statistics of the previous studies.

2.5.2 Document Classification: Keyword spotting System

Keyword spotting is grown research field because of its importance as an alternative approach to OCR for retrieving documents from digital archives. We focused on query types and feature extraction methods of KWS because they are forming the characteristics of these KWS systems. KWS has three main phases, namely, input phase, feature extraction phase, and matching phase. Researchers tend to design KWS techniques that provide easy to form queries such as strings [143]. However, it affects the overall performance of the KWS system due to introducing new issues such as out-of-vocabulary spotting.

According to our literature review sample, hybrid-based KWS approach that exploits the text and image of the queries to generate keyword queries is forming around 9.5% of the population of the reviewed KWS work. This type of KWS systems might witness an increase in interest of studying its effects on the performance.

In general, 71.42% of the previous research have focused on QBE because it is intuitive choice for researchers and part of QBS queries are converted to images before carrying out spotting tasks. This means several issues of the current QBE system should be studied and addressed before adding an auxiliary task such as query synthesizing step.

KWS speed is another factor that researchers took care of. It is the product of feature design and matching strategies. It has been reported in several previous studies that word spotting algorithms used fixed-length features have carried out word spotting faster than variable-length features [24, 11].

Other KWS design factors such as document types, layouts, and analysis also contributed to the overall KWS system performance. Basically, KWS was introduced as an alternative to OCR to deal with handwritten, and degraded documents because OCR does not preform well in these situations. It is observed in our literature review that 76.2% of the population focused on handwritten documents. Moreover, 61.9% of the studies considered unstructured documents. These two statistical figures indicate that keyword spotting is used for complex document-image retrieval. Finally, the KWS algorithms addressed English language more than other languages, around

68.42% of the reviewed studies used English documents. This could be attributed to availability benchmark datasets in English. Figure 2.13 summarize the reviewed KWS algorithms based on the discussed factors.

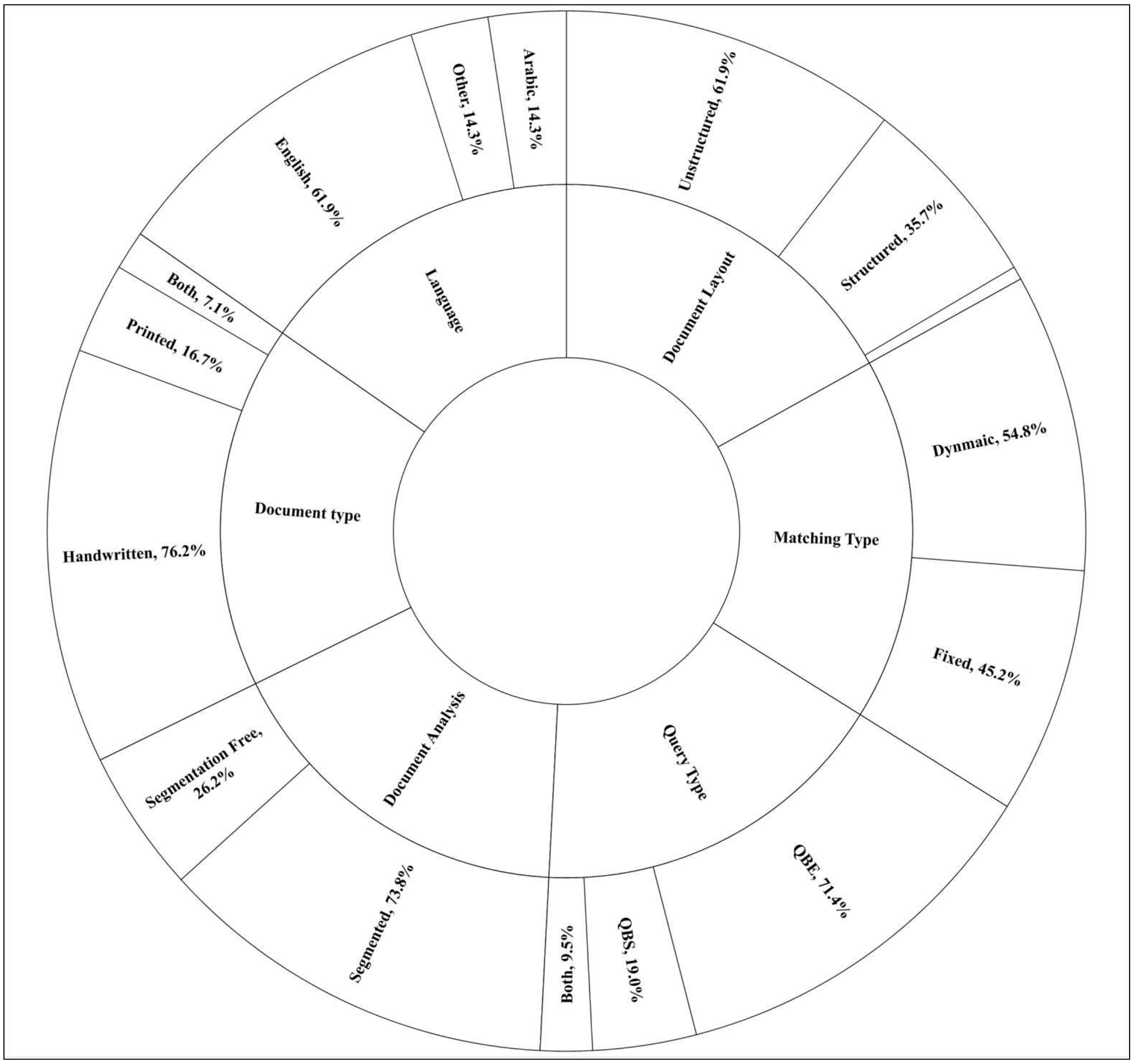


Figure 2.13: Summary of KWS literature

2.6 Conclusions

In this Chapter, we presented a comprehensive review of document layout analysis and classification methods. This review consists of two main parts Document layout analysis (DLA), and keyword spotting (i.e., document classification).

First, the general DLA framework includes preprocessing, analysis, and evaluation phases. The preprocessing phase consists of three main tasks; binarization, skew detection/correction, and algorithmic parameter estimation. There are several binarization techniques that have been developed in the past years. They are divided into global and local (i.e. adaptive). The Global binarization is suitable for document images that have been scanned in a controlled environment. The adaptive binarization is appropriate for degraded document images. Binarization is avoided in some techniques due to three reasons; **1)** utilize color information in the analysis, **2)** avoid new binarization leftover noise, and/or **3)** reduce the preprocessing time. Still, the color-based document analysis approach is not studied as much as binary document images.

Skew angle detection/correction is a broad research area. Like binarization, the skew angle can be detected at global or local levels. The global level skew angle is found over the document image, while the local skew angle is detected and corrected at the region levels. It can be observed from the literature that skew detection/correction techniques have different range of angle-correction that starts by small range $\pm 15^o$ as in the cross-correlation approach to large angle-correction range as in the Radon transform approach. Finally, most DLA algorithms set analysis parameters at the preprocessing phase. These parameters are of two types; static or dynamic. The static parameters are fixed thresholds, which are set by the algorithm's designers based on pre-knowledge of processed documents. DLA algorithms with static parameters are usually fast and used in top-down strategy. On the other hand, the dynamic parameters are concluded from document images automatically. Dynamic parameters are often found in bottom-up techniques.

Document analysis can be carried out based on three strategies namely bottom-up, top-down and hybrid. Most of the reviewed algorithms use either bottom-up or top-down strategy. A single strategy could be insufficient to carry out robust document analysis for complex layouts. Therefore, hybrid strategy may be used to address such situation. According to our review, hybrid strategy has been studied less than other strategies(around 10% of reviewed population).

There are three types of evaluation frameworks; pixel-based, graph-based, and region-pixel-based. The pixel based evaluation is aggressive, because it assumes that all ground-truth pixels are matched to the segmented pixels to be counted as correct segmentation. On the other hand, both region and graph frameworks provide degree of tolerance in the evaluation. Therefore, they may suitable to evaluate the performance of DLA algorithms that analyze complex document layouts.

Document classification can be addressed using keyword spotting techniques by retrieving documents that have instances of a query-keyword. In the literature, keyword spotting approach has various design options related to processed documents, and query representation. Essentially, keyword spotting is designed to avoid handwritten document transcription by spotting and retrieving documents using keyword images (i.e., retrieve by example). Then, word spotting techniques have evolved by allowing QBS scenario, where a user can search digital archives using text. Both techniques have strengths and weaknesses. For example, QBE techniques are considered faster than QBS, but the search is limited to the existing keyword examples. On the other hand, QBS allows arbitrary word searching and suffers from out-of-vocabulary spotting issue. Therefore, current state-of-the-art research direction could be formulated towards studying combination of both type of queries such as PHOC technique.

For the future, we believe that DLA algorithms should relax assumptions on page layouts and develop generic algorithms. To approach this objective, adaptive and learnable DLAs may be used. A Texture-based analysis could be strong choice to carry out feature extraction from document images to support learnable algorithms. Moreover, layout features can be learned automatically using deep learning algorithms. Therefore, it is expected that deep learning will find its way in DLA.

CHAPTER 3

ARABIC HISTORICAL HANDWRITTEN MANUSCRIPT DATASET

Benchmark datasets are important for the research community. It enables researchers to compare their algorithms, it saves researchers from building their own data set and hence reduce the time of development. Most of the available benchmarking historical document datasets are in non-Arabic language. Although analysis algorithms should be generic, they are language specific in most cases. For example, in historical English manuscripts such as GW20 dataset [185], the writing may be divided into single words. However, in Arabic manuscripts as HADARA80P dataset [188], the writing, in most cases cannot be divided into single words. Arabic handwriting can be separated into Part of Arabic Words (PAW). Moreover, it could be hard to separate Arabic writing into exact PAWs due to handwritten ligatures, and touching-words (see Figure 3.1). Such handwritten issues also existed in other languages, however they have been addressed in language-specific manner. Therefore, document analysis and classification can be carried out differently from one language to another.

There is a lack in available historical Arabic datasets which is noticed by several research groups. A new historical Arabic dataset has been published (e.g. HADARA80P dataset [188]). However, the characteristics and objectives of the proposed dataset are different from others. It includes pages from different manuscripts(i.e., multi-writer dataset), and each page consists of main-content and side-notes(i.e., various levels of text density).

In this chapter, we present our proposed Arabic Historical Handwritten Manuscript (AHHM) dataset. In Section 3.1, a brief introduction and AHHM framework are presented. AHHM dataset sources are outlined in Section 3.2. In Section 3.3, AHHM data format and properties are described. Manuscript pages of AHHM dataset are characterized by a list of keywords that are presented in Section 3.4. Final remarks are given in Section 3.5.

3.1 Introduction

Arabic historical manuscript analysis and classification has not been addressed equally in research as other languages; one important reason is the lack of benchmark historical Arabic datasets. Most of the available Arabic benchmark datasets are either contemporary or non-Arabic. Examples of these datasets are discussed in Chapter 2. Although digital libraries allow access to large amount of valuable ancient Arabic manuscripts online, they miss automatic searching tools to speedup information retrieval. Therefore, developing such tools requires representative datasets.

Recently, there are two Arabic Handwritten document datasets that have been published BCE-Arabic-v1, HADARA80P (discussed in Chapter 2). The BCE-Arabi-v1 is a large dataset of contemporary Arabic documents. Therefore, it targets office-based document analysis and classification such as memos, magazines, articles etc. [179]. HADARA80P dataset has similar objectives to AHHM dataset. It was developed with a focus on keyword spotting and information retrieval [188]. Therefore, all pages in

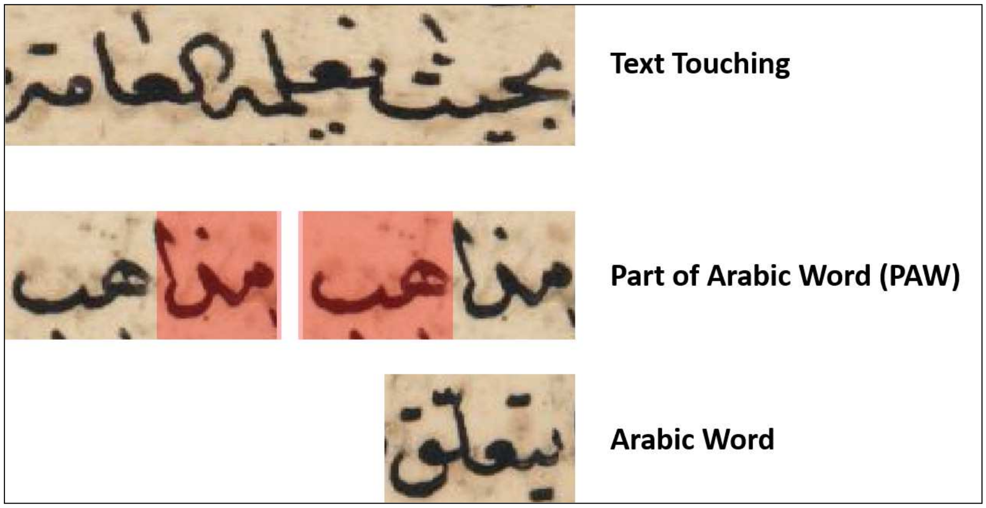


Figure 3.1: Arabic word segmentation variations

HADARA80P dataset are almost clean with main-content and rare side-note texts. The pages are written by a single writer. All pages extracted from one source book that preserves the handwriting style of all keywords. Thus, HADARA80P dataset has missed addressing multi-writer text, which is needed to allow researchers test various handwriting styles, font types, and different levels of text-density per page. In addition, the proposed AHHM dataset is created to address the following concerns:

- Historical manuscript layout analysis: Each page, in AHHM, has a main-content and several side-note blocks. Moreover, it offers different text-density per page. Therefore, different document analysis algorithms can be evaluated using the proposed AHHM.

- Information retrieval: AHHM pages are characterized by a list of 25 keywords. The dataset's words are extracted manually from the main content of each page. Then, the selection of keywords is carried out based on the number of examples per word.

- Multiple writers dataset: Manuscripts in AHHM dataset are written by at least four writers, which makes the dataset more realistic for information retrieval. Page samples in AHHM are extracted from four source books. Moreover, side notes are written by different writers. Therefore, at least four writing-styles can be tested by analysis and retrieval algorithms.

Table 3.3 presents a summary of AHHM dataset word breakdown. There are 20 words on average extracted from each manuscript page.

3.1.1 AHHM framework

The framework of AHHM dataset includes four phases; identify data sources, page selection, binaraization, and segmentation. In the data-sources phase, we visited several digital libraries and archives to gather manuscript samples. There are several digital libraries that provide historical manuscripts on-line, but few of them are utilized with document surfing software. In data source phase, we were able to identify two main digital libraries to collect manuscript pages as described in Section 3.2. Then, to match our research aims, we set two main rules to select a manuscript page; 1) A manuscript page should contain mostly text. 2) It should have some marginal comments. The rules are set to create different dataset and to extend the available ones.

There are several DLA techniques that require binarized manuscripts such as [1, 193]. Therefore, manuscript pages of AHHM dataset are provided in two versions, colored and binary. Finally, the main content and keywords are manually segmented to form AHHM ground truth. Figure 3.2 illustrates the AHHM framework phases.

3.2 AHHM Data Sources

The AHHM data is collected from two main digital libraries; Harvard Library[12], and the State Library of Berlin (SLB)[200].

First, within the Islamic Heritage Project (IHP) of Harvard library, 280 Arabic manuscripts, 50 maps, and more than 275 books gathered from Harvard's library and museum collections. In total, Harvard offers 156,000 of ancient materials (e.g.

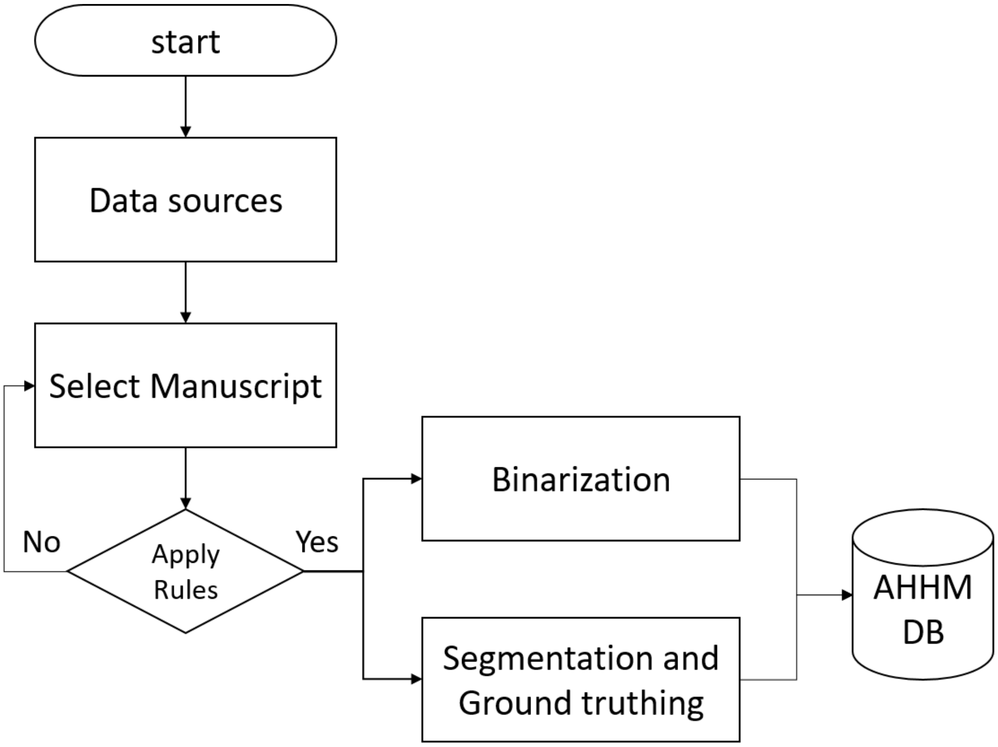
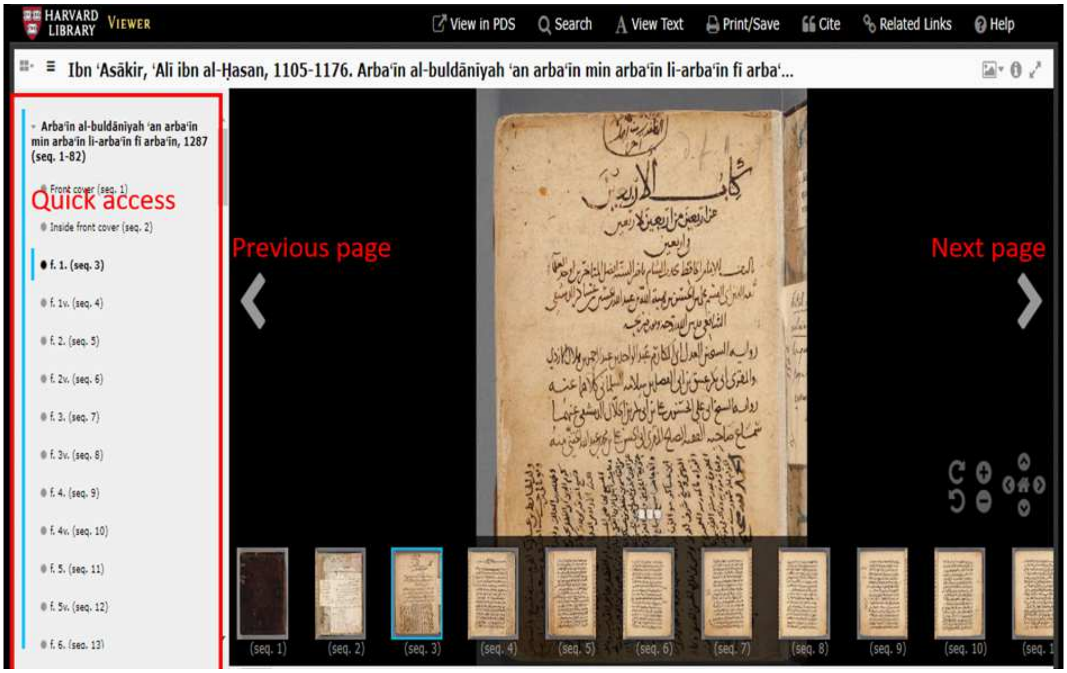


Figure 3.2: AHHM framework phases

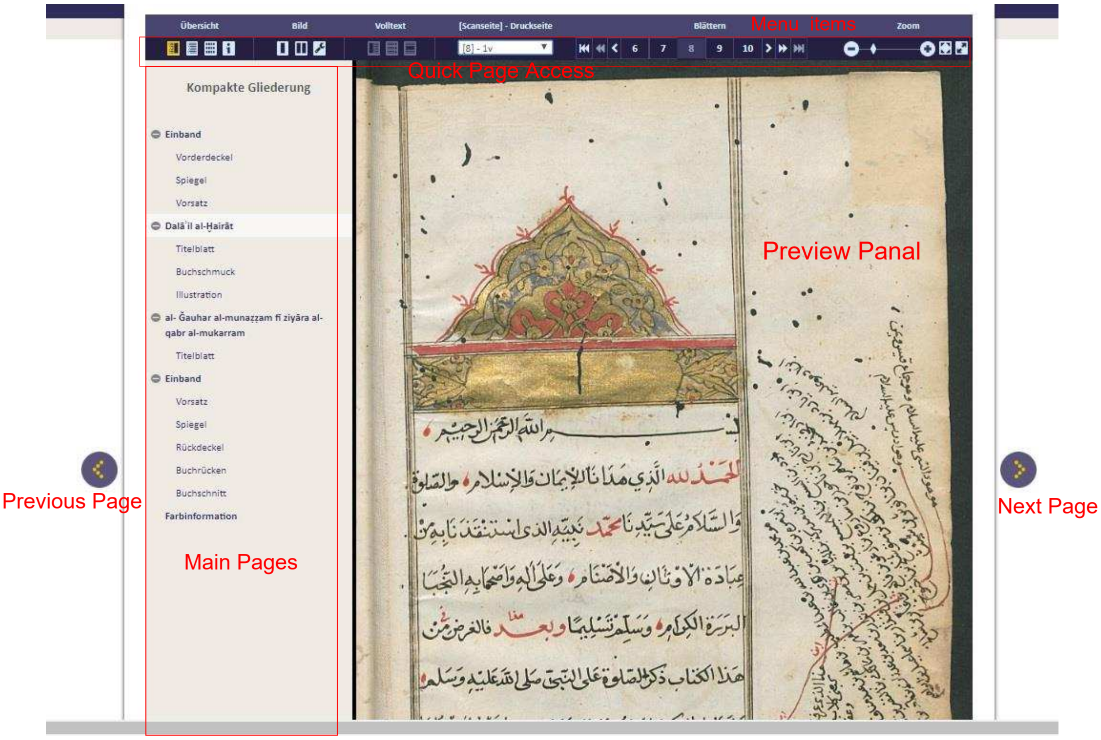
manuscripts, maps etc) dated from the 10^{th} to the 20^{th} centuries. Secondly, SLB offers diversity of ancient materials including Arabic historical manuscripts. Approximately, SLB has 8,000 manuscripts that include text, miniatures, illustrations, and drawings.

Unlike other digital libraries, these libraries allow non-commercial usage of manuscripts, and there is no watermarking on these manuscript pages. Moreover, the libraries are utilized with a software that allows visitors to read, surf, search, and download pages. However, the search facility is not working on historical manuscripts due to unavailable text. Therefore, visitors need to search through reading manuscript pages (i.e., manual search). Figures 3.3.(a) and (b) show front-page of these softwares.

The developed AHHM dataset is based on four books. So, at least four writers have



(a) Harvard Library



(b) State of Berlin Library

Figure 3.3: Example of digital libraries software interface

written the main content of these books because ancient manuscript may be written by several writers [188]. Table 3.1 outline page samples taken from each source book. Transliterated titles and author names of these books are listed below:

- Book 1: title ”كتاب التعريف لمذهب التصوف” which is transliterated to ”Kitab al-Taarruf li-madhhab ahl al-tasawwuf” written by Ahmad ibn al-Husayn. The book is categorized as Sufism doctrines book.

- Book 2: title ”كتاب بحر الكلام في التحويد” which is transliterated to ”Kitab Bahr al-kalam fi ilm al-tawhid” written by Nasafi, Maymun ibn Muhammad. The book discusses Islamic doctrines.

- Book 3: title ”كتاب بغية الطالب في معرفة المفروض و الواجب” which is transliterated to ”Bughyat al-talib fi marifat al-mafrud wa-al-wajib ” written by Jafar ibn Khadir Janaji. The book discusses Islamic doctrines.

- Book 4: title ”كتاب دليل الخيرات” which is transliterated to ”Kitab Dalaeil al-Khairat ” written by Muhammad ibn-Sulaiman Jazuli. The book is categorized as Sufism.

Pages are selected from each book based on their layout characteristics. The desired layout should have main-content and some side-note texts. The amount of side-note text may vary from one page to another to allow testing different levels of document layout analysis algorithms. Figure 3.4 shows one sample from each book.

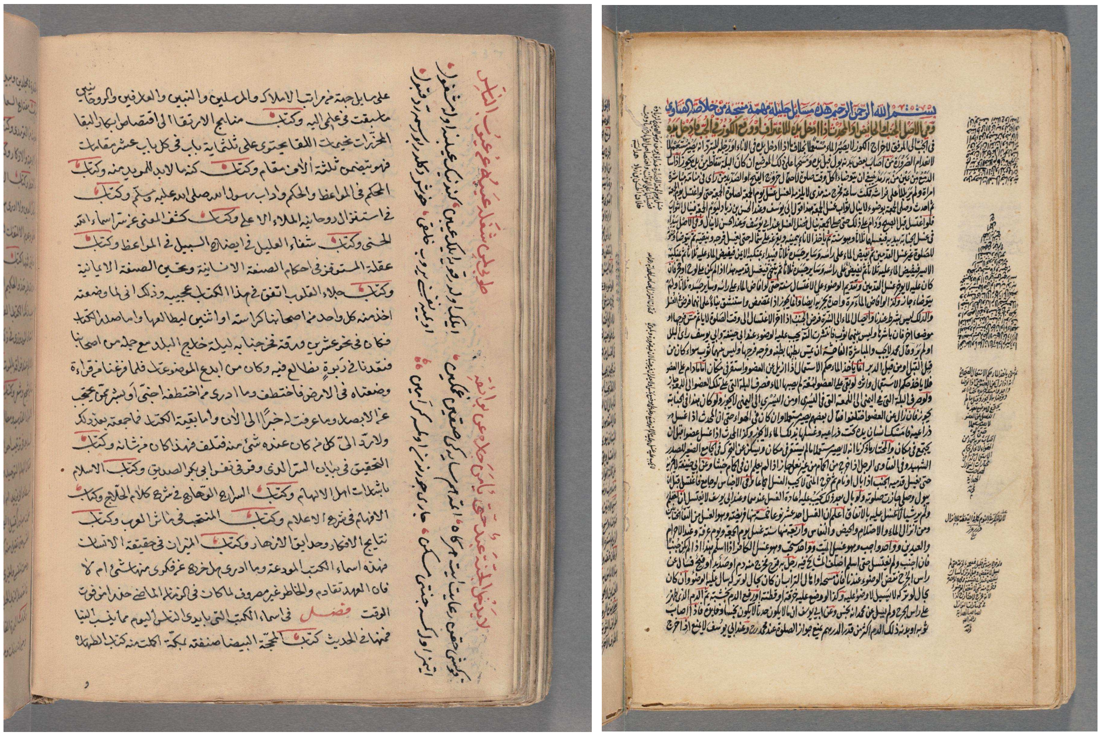
Table 3.1: AHHM dataset resources

Book	**Date**	**Source**	**Pages**	**Subject**
Book 1	10th Cen.	Harvard	Smaple1 - Sample9	Sufism doctrines
Book 2	Undated	Harvard	Sample10 - Sample39	Islam doctrines
Book 3	1247 Hj	Harvard	Sample40 - Sample51	Islam doctrines
Book 4	1729 Gn	SLB	Smaple52 - Sample108	Sufism

- Hj: Hijri, Gn: Gregorian, Cen:Century

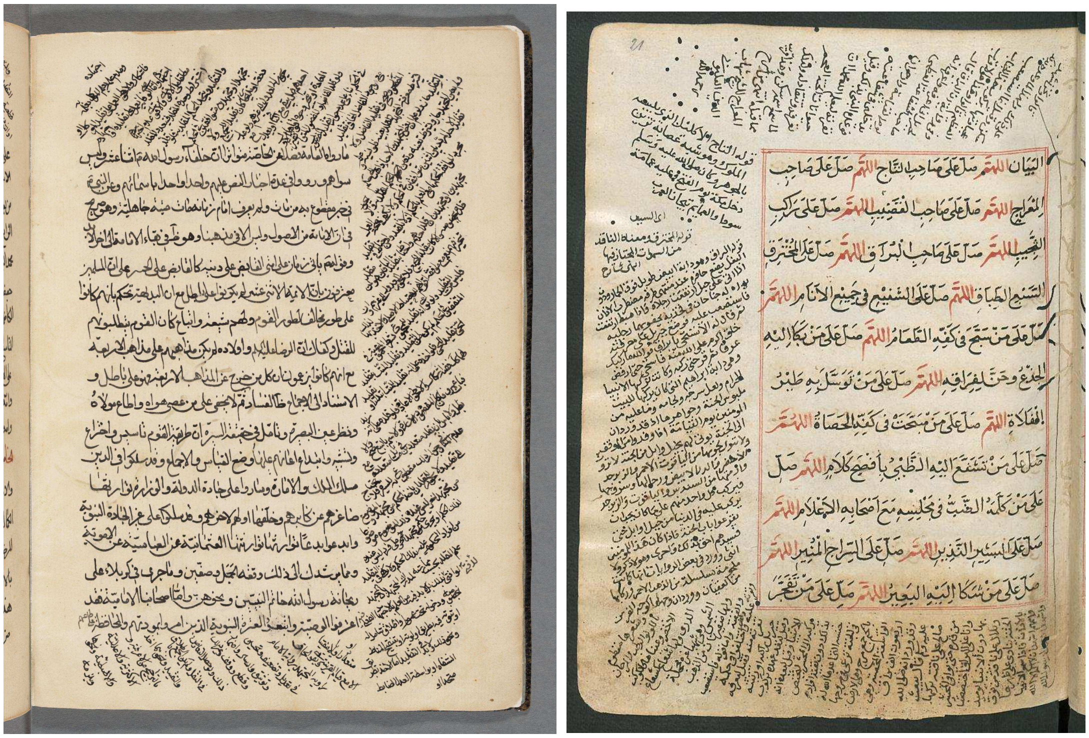
3.3 Data Format and Properties

The AHHM dataset consists of 108 page-images extracted from four books. The pages are mostly text. There are no drawings, figures, or decorations. The amount of text on each page vary based on the writing style. Moreover, images' sizes are different. Manuscript pages that are extracted from SLB [200] have resolutions between 277 to 290 dots per pixel (DPI). There are 57 pages extracted from one book (Book 4). Book4 pages contains 11 text lines per page on average. SLB pages make 52.7% of AHHM dataset that allow evaluation of moderate-to-hard document analysis algorithms. For complex document layout analysis and classification algorithms, Harvard manuscript samples can be used. They form 47.2 % of the AHHM pages with different writing styles, quality conditions, and image sizes. Harvard manuscript samples have fixed resolution at 300 DPI. The writing in these manuscripts is dense and may suffer from text-touching (see Figure 3.1). Number of text lines per page vary from one book to another (24 text lines on average). This makes these manuscripts challenging for document analysis. Figure 3.4 shows a page example taken from each book.



(a) Sample 03 [12]

(b) Sample 24 [12]



(c) Sample 44 [12]

(d) Sample 83 [200]

Figure 3.4: AHHM manuscript samples

3.3.1 Ground truth methodology

To prepare the ground truth of AHHM, we used MATLAB software to select polygon points manually around the desired text-objects. A user is asked to mark main content and 20 words on each manuscript page. The segmentation of a block requires at least four polygon points. Based on the manuscript layout conditions, more polygon points may be selected to mark some text blocks. Similarly, words may require at least three polygon points for segmentation. The word segmentation could be harder than larger blocks because of dense-text and text-touching issues. Figure 3.5 shows an example

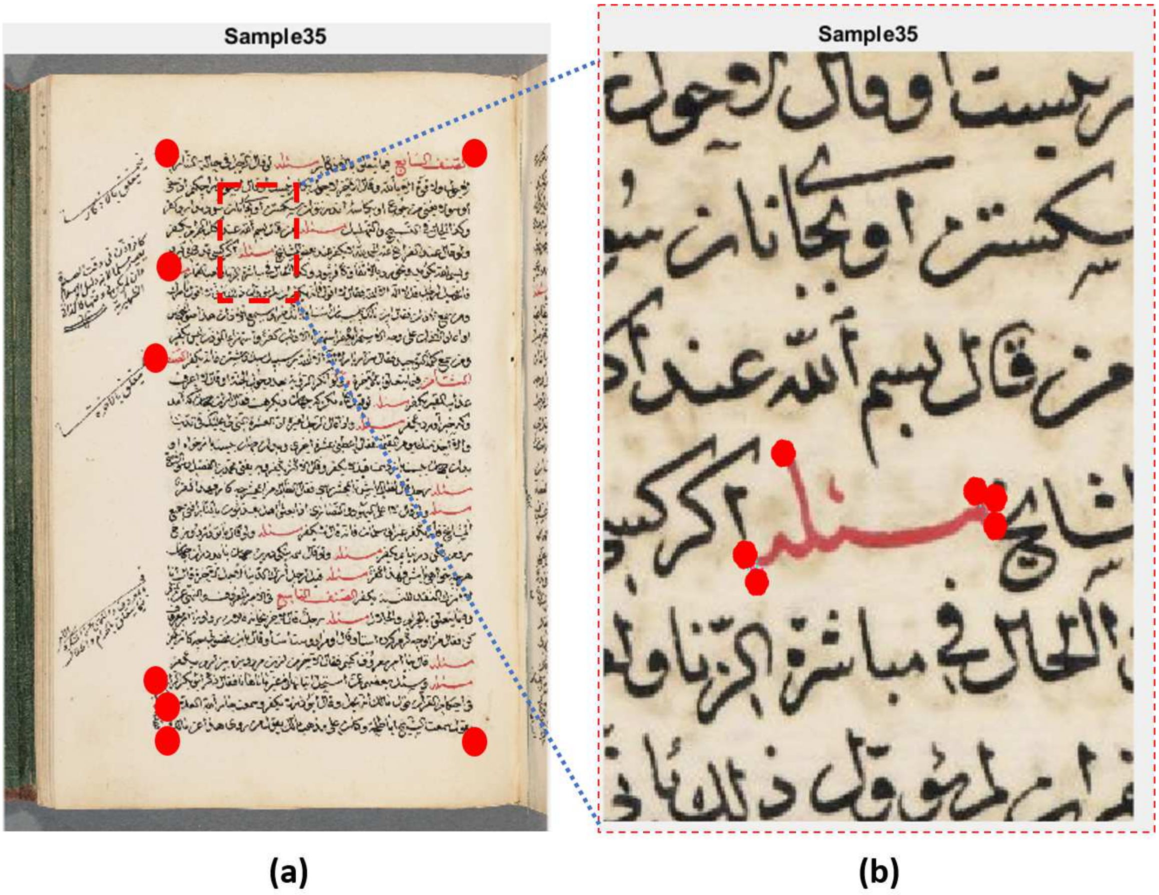


Figure 3.5: Examples of block and word segmentation; a) Block segmentation, b) Word segmentation

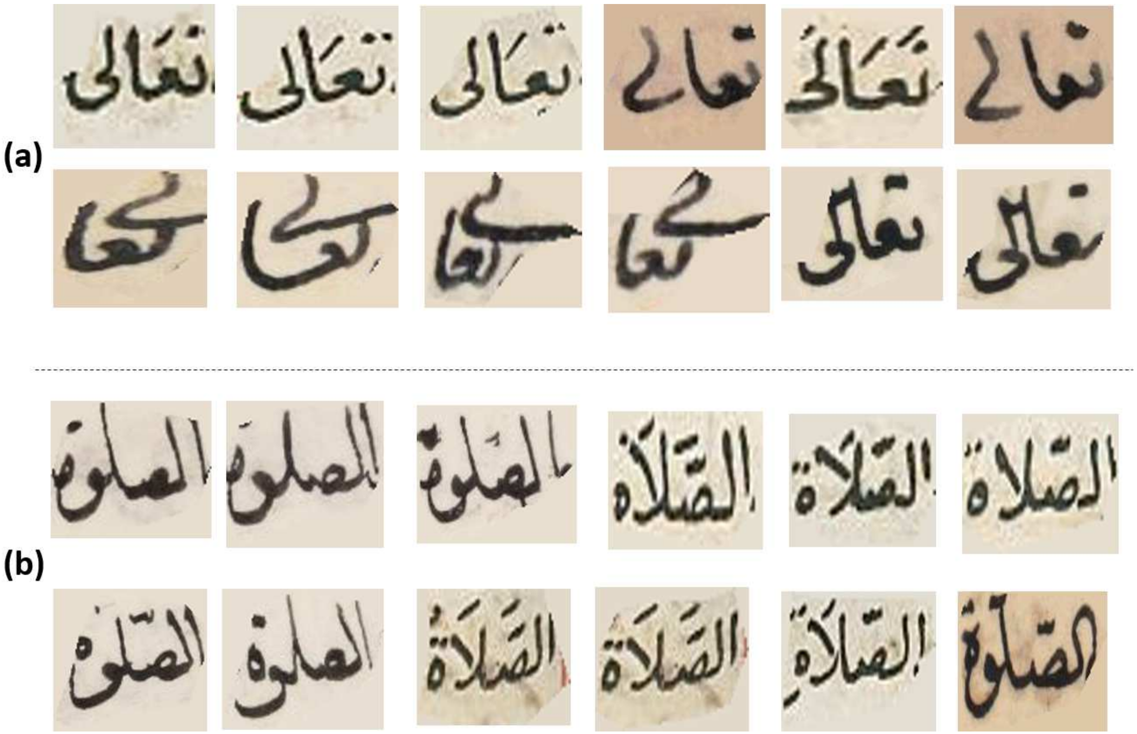


Figure 3.6: Two examples of keywords that have different writing fonts, and styles, a) KW06, b)KW24

of defining "Sample35" ground truth. In Figure 3.5.(a) main content is defined by locating eight points(red dots around main content), and a word is segmented by selecting six points in Figure 3.5.(b). The text of Harvard manuscript-pages are denser than SLB manuscript-pages (see Figure 3.4.(b) and (d)). Therefore, word segmentation could be error prone and requires careful segmentation. Examples of segmented words are shown in Figure 3.6.

3.3.2 Ground Truth Format

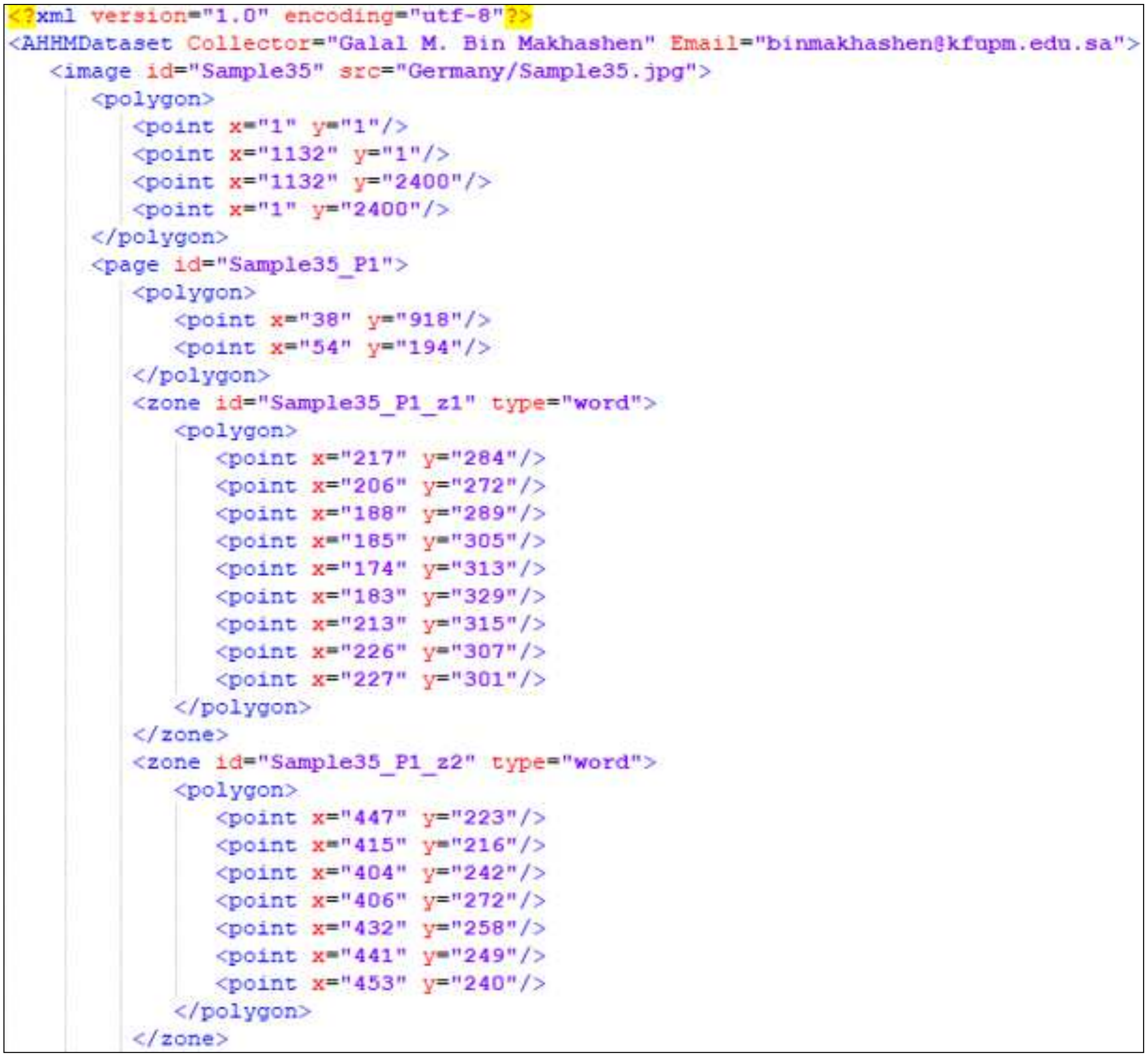
Each page is associated with an eXtensible Markup Language (XML) [201] file that contains its ground truth. The selection of XML file format to store dataset ground-truth has two main benefits; 1) It allows both segmentation, and segmentation-free document retrieval, 2) Both binarized and colored AHHM dataset can use the same ground truth XML file.

Each XML file is divided into three main tags; image, page, and zone. The image tag represents the actual size of a manuscript page. It stores the X-Y coordinates of the image associated with the file ID, and data-source. The Page tag is used to store the text blocks coordinates. It gives each block an ID such as "Sample35_P1" as shown in Figure 3.7. Finally, a zone tag is used to store word coordinates in each block. A sample content of an XML file is shown in Figure 3.7.

3.4 Keywords

Besides providing the ground truth, the AHHM dataset includes 25 selected keywords. Table 3.2 shows a reference word of each keyword.

The keywords can be used to develop information retrieval systems such as keyword spotting. Unlike other datasets that allow using every word as a query to evaluate word spotting algorithms[184]. Usually, these datasets were not designed for keyword spotting, instead they were built for word recognition. For example, GW20 dataset was used in [161] with 1090 keywords, [21] with 105 keywords. It is obvious that using all dataset keywords is computationally not feasible as reported in [188]. Moreover, using different number of keywords does not support comparing different algorithms. We selected 25 keywords to make AHHM comparable to HADARA80P dataset. Moreover, it extends HADARA80P dataset by introducing further challenging issues such as multi-writer and different writing styles. Figure 3.6 shows two



```
<?xml version="1.0" encoding="utf-8"?>
<AHHMDataset Collector="Galal M. Bin Makhashen" Email="binmakhashen@kfupm.edu.sa">
   <image id="Sample35" src="Germany/Sample35.jpg">
      <polygon>
         <point x="1" y="1"/>
         <point x="1132" y="1"/>
         <point x="1132" y="2400"/>
         <point x="1" y="2400"/>
      </polygon>
      <page id="Sample35_P1">
         <polygon>
            <point x="38" y="918"/>
            <point x="54" y="194"/>
         </polygon>
         <zone id="Sample35_P1_z1" type="word">
            <polygon>
               <point x="217" y="284"/>
               <point x="206" y="272"/>
               <point x="188" y="289"/>
               <point x="185" y="305"/>
               <point x="174" y="313"/>
               <point x="183" y="329"/>
               <point x="213" y="315"/>
               <point x="226" y="307"/>
               <point x="227" y="301"/>
            </polygon>
         </zone>
         <zone id="Sample35_P1_z2" type="word">
            <polygon>
               <point x="447" y="223"/>
               <point x="415" y="216"/>
               <point x="404" y="242"/>
               <point x="406" y="272"/>
               <point x="432" y="258"/>
               <point x="441" y="249"/>
               <point x="453" y="240"/>
            </polygon>
         </zone>
```

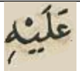
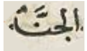
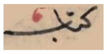
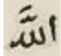
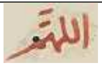
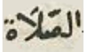
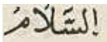
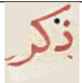
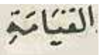
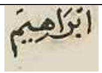
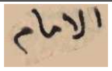
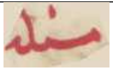
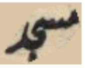
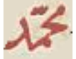
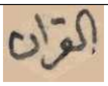
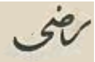
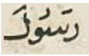
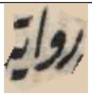
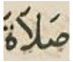
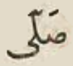
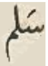
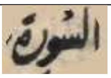
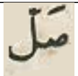
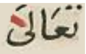
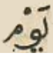
Figure 3.7: XML ground truth example

keyword examples that have instances in different writing styles.

In total, AHHM dataset has 2135 extracted words from 108 manuscript pages. There are 1061 keyword instances extracted from all manuscripts. This is because some extracted words have been rejected due to segmentation errors.

The extracted keyword samples are shown in Table 3.2 as rectangular images. Each keyword-image's background is filled with median of all pixel colors present in the keyword zone. Moreover, shown keywords in Table 3.2 are resized and modified, be-

Table 3.2: AHHM selected keyword samples

Code	TRANS	Keyword	NS	Code	TRANS	Keyword	NS
KW01	Aaliah		51	KW02	Al-Janah		9
KW03	Ketab		100	KW04	Allah		165
KW05	Allahm		61	KW06	Al-Salaah		29
KW07	Asslam		35	KW08	Dekr		18
KW09	Al-Gaiamah		6	KW10	Ibrahim		25
KW11	Al-Imam		21	KW12	Msalah		37
KW13	Masjed		7	KW14	Mohammed		230
KW15	Al-Quraan		10	KW16	Radhi		6
KW17	Rasoul		26	KW18	Rewaih		5
KW19	Salaah		11	KW20	Sali		45
KW21	Slm		65	KW22	Al-Sourah		7
KW23	SsL		49	KW24	Ta-ala		33
				KW25	Youm		9

- TRANS: Transliteration, NS: Number of instances

cause extracted words have different sizes due to writing-styles and fonts sizes. Table 3.3 summarizes the content of the dataset, and indicates the number of words per manuscript sample page.

Table 3.3: AHHM Dataset Summary

Page	NW	Keywords	Other	Page	NW	Keywords	Other
Sample01	20	20	0	Sample41	20	8	12
Sample02	19	13	6	Sample42	19	19	0
Sample03	20	16	4	Sample43	20	7	13
Sample04	20	15	5	Sample44	20	0	20
Sample05	20	14	6	Sample45	20	9	11
Sample06	18	15	3	Sample46	19	12	7
Sample07	20	15	5	Sample47	20	12	8
Sample08	20	17	3	Sample48	20	16	4
Sample09	18	13	5	Sample49	20	6	14
Sample10	20	11	9	Sample50	19	7	12
Sample11	20	9	11	Sample51	16	10	6
Sample12	20	10	10	Sample52	20	15	5
Sample13	20	6	14	Sample53	20	7	13
Sample14	20	16	4	Sample54	20	7	13
Sample15	20	14	6	Sample55	20	13	7
Sample16	20	18	2	Sample56	20	17	3
Sample17	20	19	1	Sample57	20	16	4
Sample18	20	4	16	Sample58	18	10	8
Sample19	20	5	15	Sample59	20	8	12
Sample20	20	12	8	Sample60	20	10	10
Sample21	20	15	5	Sample61	20	18	2
Sample22	20	8	12	Sample62	20	18	2
Sample23	20	10	10	Sample63	20	15	5
Sample24	20	12	8	Sample64	20	20	0
Sample25	20	6	14	Sample65	20	16	4
Sample26	20	16	4	Sample66	20	14	6
Sample27	19	8	11	Sample67	18	6	12
Sample28	20	12	8	Sample68	20	14	6
Sample29	20	14	6	Sample69	20	3	17
Sample30	20	13	7	Sample70	20	2	18
Sample31	20	10	10	Sample71	20	3	17
Sample32	20	4	16	Sample72	20	2	18
Sample33	20	9	11	Sample73	20	5	15
Sample34	20	11	9	Sample74	20	5	15
Sample35	20	7	13	Sample75	20	2	18
Sample36	20	14	6	Sample76	20	11	9
Sample37	20	13	7	Sample77	20	5	15
Sample38	20	13	7	Sample78	20	7	13
Sample39	20	10	10	Sample79	20	6	14
Sample40	20	11	9	Sample80	20	7	13

- NW.: Total number of extracted words

Page	NW	Keywords	Other	Page	NW	Keywords	Other
Sample81	19	8	11	Sample95	19	17	2
Sample82	20	2	18	Sample96	20	10	10
Sample83	19	9	10	Sample97	20	10	10
Sample84	19	5	14	Sample98	20	05	15
Sample85	20	9	11	Sample99	20	6	14
Sample86	18	8	10	Sample100	19	4	15
Sample87	20	4	16	Sample101	18	1	17
Sample88	20	10	10	Sample102	20	7	13
Sample89	20	1	19	Sample103	20	0	20
Sample90	20	5	15	Sample104	20	2	18
Sample91	20	16	4	Sample105	20	6	14
Sample92	20	15	5	Sample106	20	6	14
Sample93	20	20	0	Sample107	20	14	6
Sample94	20	8	12	Sample108	20	7	13

- NW: Total number of extracted words

3.5 Conclusions

In this Chapter, we present an Arabic Historical Handwritten Manuscript (AHHM) dataset. The dataset consists of 108 manuscript pages dated from 10^{th} to 18^{th} centuries. AHHM manuscript pages are selected from different books to allow studying the effects of multi-writer, writing-styles, and multi-font scenarios on document layout analysis, classification, and retrieval techniques.

AHHM dataset offers two types of manuscripts; colored and binary. It can be used in segmentation or segmentation-free document classification and retrieval scenarios. Its ground truth is represented by XML files to allow algorithms free adaptation. Finally, AHHM dataset contains 108 extracted main, and side note blocks. In addition, it includes 2135 extracted word instances from all pages, and 25 words are selected as keywords.

CHAPTER 4

A HYBRID LAYOUT ANALYSIS FOR ARABIC HISTORICAL MANUSCRIPTS

A vast number of historical manuscripts is available in digital archives that require automatic categorization, indexing, and retrieval. Such applications depend heavily on the quality of the layout analysis process. Naturally, historical manuscripts possess complex and challenging layouts due to aging, free-writing style, marginal notes, ornamentation, ink-bleeding, etc. Therefore, the quality of extracting the main content from a manuscript page is of great importance for these applications.

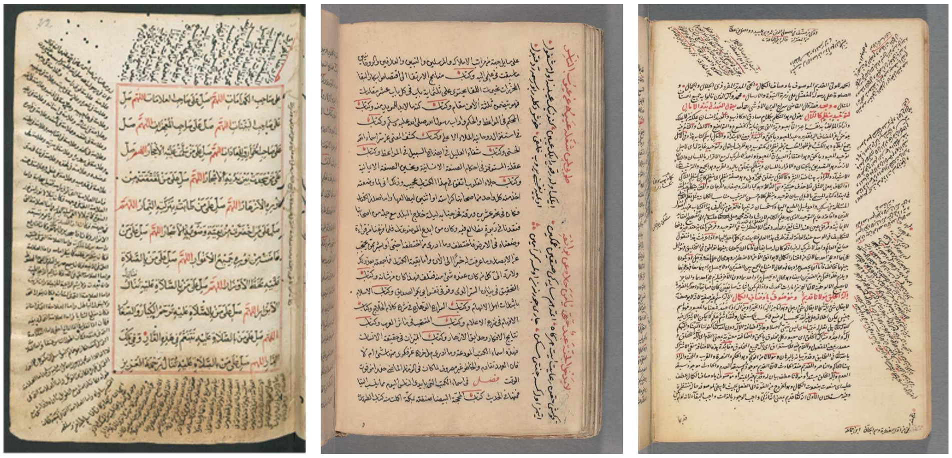
In this chapter, a learning-free hybrid analysis approach is presented. It is designed to address mostly-text historical manuscripts and locates main content region. To achieve this target, the proposed algorithm integrates bottom-up and top-down analyses.

A general framework of the algorithm is presented in 4.1. The algorithm starts by preprocessing manuscripts, which is described in Section 4.2. In Section 4.3, manuscript characterization phase is outlined. The main aim of this step is to estimate initial coordinates of main content region. In Section 4.4, we describe features and parameters extraction procedure. The hybrid analysis is presented in Section 4.5. Finally, a summary and final remarks of the proposed approach is given in Section 4.6

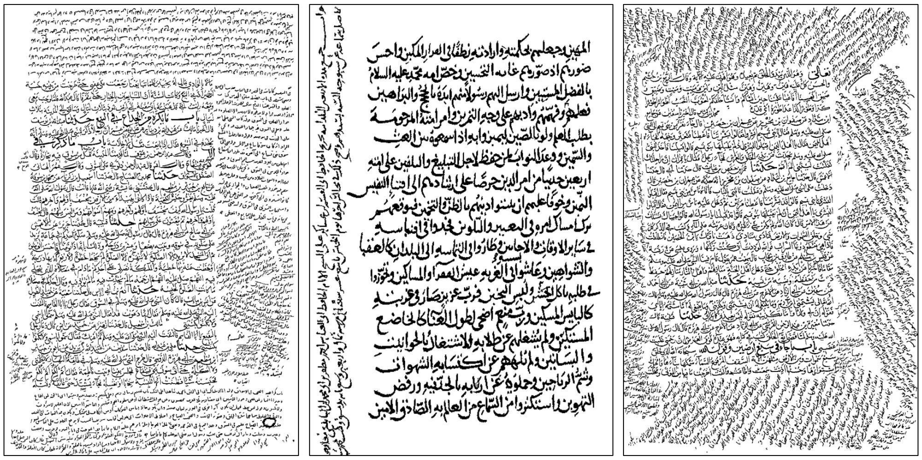
4.1 Historical Manuscript Analysis Framework

Document layout analysis is an essential research field of document understanding systems for decades [11], [202]. DLA methods (at early stages) were conducted on simple and regular document layouts. Therefore, it was considered as a sub-task of document understanding systems [84]. Eventually, researchers have encountered more complex and diverse document layouts that require robust DLA algorithms. Consequently, DLA has been recognized as a separate area of research [203].

Complex document layouts have many issues such as using multi-writing styles, quality degradation, lighting conditions, blocks organizations etc. In other words, a complex layout is a structure that has arbitrary blocks. The handwritten historical manuscripts are often considered to have complex layouts [204]. Usually, these documents are not following unique standard writing style, font type, or font size. Moreover, these documents use irregular spacing among words, text-lines, or paragraphs. In addition, they may contain marginal text (i.e., side-notes), and suffer from aging, ink bleeding, text degradation, etc. These challenging factors have made the



(a) AHHM dataaset Samples



(b) Bukhari et. al [197] dataset

Figure 4.1: Manuscript samples

analysis of historical manuscripts a hard task.

Our research objective is to analyze historical manuscripts that contain mostly text content. It could be written by different writers, in free writing style, and with dense text (some examples are shown in Figure 4.1). The proposed algorithm has two outcomes; **1)** Fast whitespace analysis for handwritten documents using an anisotropic diffusion filtering (ADF) to initially locate main content. To-our-knowledge, white space analysis has not been employed in analyzing handwritten document layouts [25]; **2)** A hybrid technique that integrates global and local analysis to extract manuscript

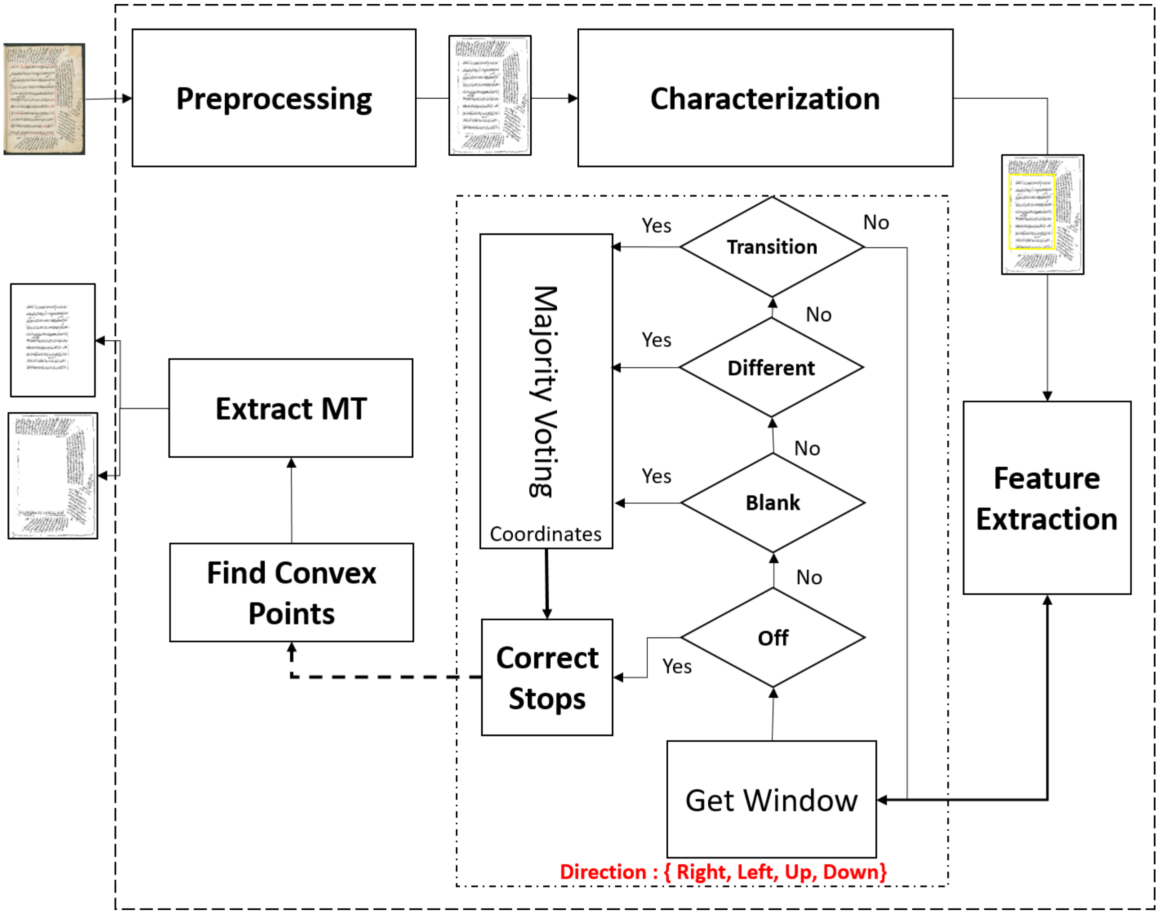
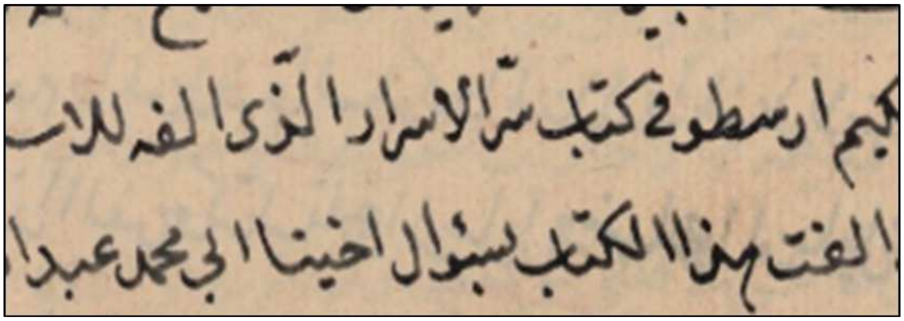


Figure 4.2: General overview of the proposed algorithm.

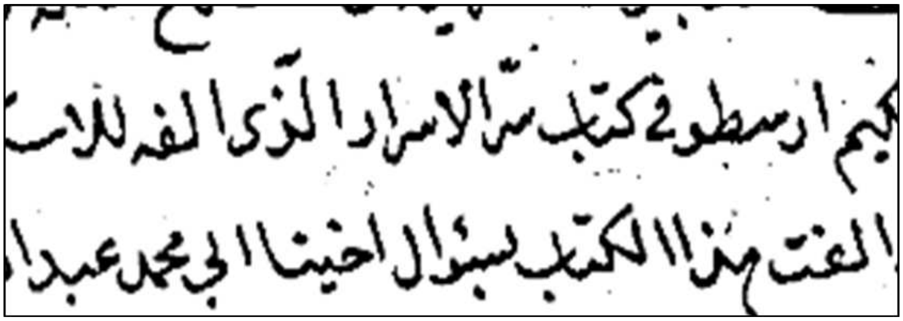
main content. A general flowchart of the proposed hybrid algorithm is illustrated in Figure 4.2.

4.2 Preprocessing

The preprocessing phase consists of two tasks; manuscript binarization, and noise removal. These two steps are presented in the following subsections.



(a) Part of manuscript page



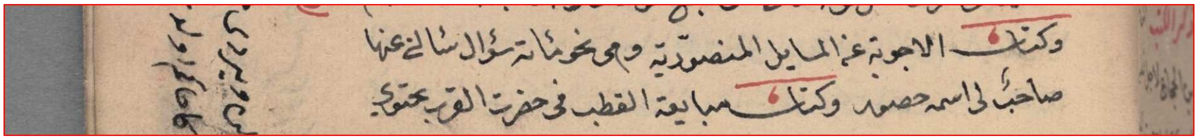
(b) Binary result

Figure 4.3: Binarization example

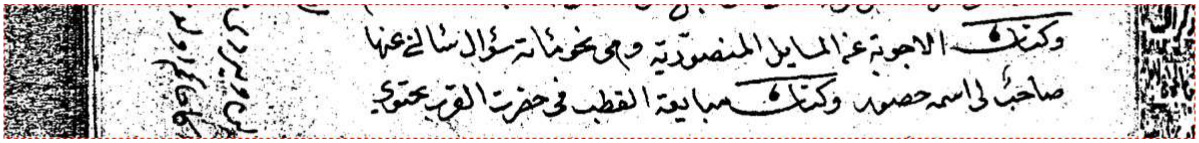
4.2.1 Binarization

Although binarization may yield undesired artifacts on the binary image, it reduces analysis time on subsequent phases. The document analysis on binary images will work only on one layer of a manuscript image. In addition, it reduces computation range of values from 256 grayscale intensities to two values [0 or 1]. Moreover, the binarization may automatically address some issues such as show-through, unbalanced illumination, and shadows that could disturb manuscript's textual content. Figure 4.3 illustrates an example of binarization.

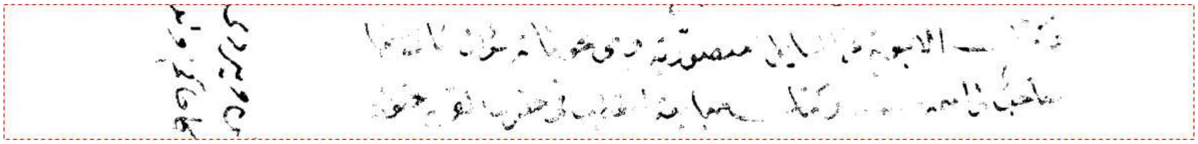
As discussed in Chapter 2, there are two main types of binarization methods; global and local. Due to manuscript degradations, global binarization may produce noise or affect the text regions because of the local background variations. Figure 4.4.(b) shows a global binrization result using Otsu method on part of the historical manuscript page shown in 4.4.(a). It can be observed that the amount of binarization leftover (i.e. noise) is larger in Figure 4.4.(b) than the noise produced by contrast-based local binarization in Figure 4.4.(d). Although contrast-based local binarization produces less noise (see Figures 4.4.(d)), it may affect the text content as shown in Figure 4.4.(c). This effect is due to computing dynamic thresholds from each image-block that makes the binaization behaves differently on each block and produces unaccepted results. In this case (Figure 4.4.(c)), the binarization removes parts of page's text. To solve this issue, dynamic binarization thresholds should be normalized to reduce this effect as suggested by Sauvola and Pietikainen in [29]. Figure 4.4.(d) shows an example of the Sauvola and Pietikainen binarization result.



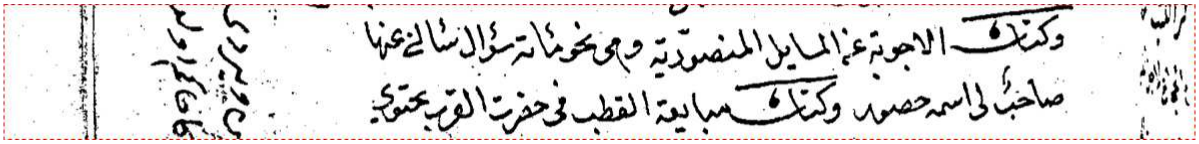
(a) Part of a manuscript page



(b) Otsu Binarization (Global) [27]



(c) Adaptive Binarization (Local) [37]



(d) Sauvola & Pietikainen Binarization (Local) [29]

Figure 4.4: Global and Local Binarization Comparison

In this work, we adopted Sauvola and Pietikainen binarization method [29]. Sauvola's algorithm performs the binarization locally based on image block statistics. Therefore, each pixel is classified as foreground or background based on its neighborhood characteristics. Sauvola's method is a modified version of Niblack [28]. It normalizes the standard deviation with its dynamic range and makes the local mean contribute less in the binarization equation (4.1). Thus, local illuminations are treated adaptively.

$$B(x,y) = \quad \mu(x+d, y+d) \times \left(1 + k\left(\frac{\sigma(x+d,y+d)}{DR} - 1\right)\right) \tag{4.1}$$

where μ and σ are the mean and standard deviation of the region (x, y), k is a small constant value computed empirically, DR is the dynamic range of the standard deviation.

4.2.2 Noise removal

The main aim of the noise removal step is to make sure that the subsequent analysis will consider significant PAWs in the analysis. Therefore, small elements such as diacritics, dots, commas, and binarization's leftover artifacts are considered as noise. The noise removal is based on the geometric characteristics of the connected components (CC). Let C_a, and C_{avg} be CC's area and the average area respectively. The average area is computed as:

$$C_{avg} = \frac{1}{n}\sum_{i=1}^{n} C_a^i$$

where n is number of significant CCs in a manuscript page. The cleaned image is computed by removing all relevant small CCs that are less than the C_{avg} as:

$$I_c = \begin{cases} C_a \in I_c \quad \text{if} \quad C_a \geq C_{avg} \\ C_a \notin I_c \quad Otherwise \end{cases}$$

where I_c is the cleaned manuscript image.

4.3 Manuscript characterization

Manuscript characterization involves fast detection of main-content region, and setting initial analysis parameters. The main-content region detection utilizes an adaptive whitespace analysis based on ADF and static whitespace localization (SWL). The algorithm generates two masks of the main-content region using ADF and SWL. Then, an integration of the masks is carried out to define an initial main-content region.

4.3.1 Adaptive main-content localization

Adaptive localization of main-content region is an initial step in estimating main-content writing characteristics. It helps in exposing the difference between main-content text and other text that could be found in a manuscript page. The adaptive main-content localization utilizes whitespace analysis using two techniques ADF and SWL.

4.3.2 Whitespace localization using anisotropic diffusion

Traditional whitespace analysis is usually applied on type-written document layout analysis such as [193, 205, 206, 207]. In traditional whitespace analysis, the segmentation is realized by detecting maximal whitespace rectangles that are located between different layout blocks such as text-columns, figures etc. [208]. Unlike previous studies, we utilize the whitespace analysis on handwritten historical manuscripts to extract the main-content region. The major whitespaces that separates main content are either long horizontally or vertically. By long we mean the major whitespace may not be rectangular shaped.The whitespace should have width or height of at least one third the length of the image I_c width or height respectively. These large whitespaces are located at the transition area between the main-content and side-notes. Therefore, small whitespaces that usually appear within the main-content or side-note areas can be avoided. This is a challenging task because historical manuscripts are mostly text with dense and unconstrained written text. Moreover, side-notes text may touch the main-content text, i.e., text touching issue (see Figure 4.1). To overcome this problem, an anisotropic diffusion filtering is applied on manuscript images to emphasize foreground/background separation and boosts the segmentation process. Then, to locate the major whitespace gaps between main-content and side-note regions, the algorithm computes the second derivative of the ADF response.

The standard form of an oriented anisotropic Gaussian filter is written as:

$$g_{(u,v,\sigma_u,\sigma_v,\theta)} = \frac{1}{\sqrt{2\pi\sigma_u}} e^{\left(-\frac{u^2}{2\sigma_u^2}\right)} * \frac{1}{\sqrt{2\pi\sigma_v}} e^{\left(-\frac{v^2}{2\sigma_v^2}\right)} \tag{4.2}$$

where $*$ is the convolution, σ_u and σ_v are the standard deviations of both frequency components u and v that are representing the direction of the angle θ and the orthogonal to θ respectively. They are defined in the following equation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where the u-axis being in the direction of θ, and the v-axis being orthogonal to θ . To make the application faster, a 2d-filter can be separated into two filters. In other words, it can be rearranged into a filter along x-direction, followed by another along a line $t = x.\cos\varphi + y.\sin\varphi$.

$$g_{(x,y,\sigma_u,\sigma_v,\theta)} = \frac{1}{\sqrt{2\pi\sigma_x}} e^{\left(-\frac{x^2}{2\sigma_x^2}\right)} * \frac{1}{\sqrt{2\pi\sigma_\varphi}} e^{\left(-\frac{t^2}{2\sigma_\varphi^2}\right)} \tag{4.3}$$

with the impulse response as:

$$g_{(x,y;\sigma_x,\sigma_y,\theta)} = \frac{1}{2\pi\sigma_x\sigma_y}.e^{\left(-\frac{1}{2}\left(\frac{\left(\frac{x-y}{\tan\theta}\right)^2}{\sigma_x^2}+\frac{\left(\frac{y}{\sin\theta}\right)^2}{\sigma_y^2}\right)\right)}$$

To yield the decomposition, Equation (4.2) can be written as:

$$\begin{aligned} g_{(x,y;\sigma_u,\sigma_v,\theta)} = \\ \frac{1}{2\pi\sigma_u\sigma_v}.e^{\left(-\frac{1}{2}\left(\frac{(x\cos\theta+y\sin\theta)^2}{\sigma_u^2}+\frac{(-x\sin\theta+y\cos\theta)^2}{\sigma_v^2}\right)\right)} \end{aligned} \tag{4.4}$$

Expanding the quadratic terms in Equation (4.4) yields the following system of equations:

$$\frac{x^2}{y^2} = x^2\frac{\cos^2\theta}{\sigma_u^2} + x^2\frac{\sin^2\theta}{\sigma_v^2} \tag{4.5}$$

$$\frac{y^2}{\sigma_x^2\tan^2\theta} + \frac{y^2}{\sigma_y^2\sin^2\theta} = y^2\frac{\cos^2\theta}{\sigma_v^2} + y^2\frac{\sin^2\theta}{\sigma_u^2} \tag{4.6}$$

$$-\frac{2xy}{\sigma_x^2\tan\theta} = 2xy\cos\theta\sin\theta\left(\frac{1}{\sigma_u^2} - \frac{1}{\sigma_v^2}\right) \tag{4.7}$$

By solving equations (4.5, 4.6 and 4.7) it yields the decomposition of the filter along x-axis with standard deviation:

$$\sigma_x = \frac{\sigma_u\sigma_v}{\sqrt{\sigma_v^2\cos^2\theta + \sigma_u^2\sin^2\theta}}$$

and along the line $t : y - x.\tan\theta = 0$, with standard deviation:

$$\sigma_y = \frac{1}{\sin\theta}\sqrt{\sigma_v^2\cos^2\theta + \sigma_u^2\sin^2\theta}$$

and intercept:

$$\tan\theta = \frac{\sigma_v^2\cos^2\theta + \sigma_u^2\sin^2\theta}{(\sigma_u^2 - \sigma_v^2)\cos\theta\sin\theta}$$

hence, the filter in Equation 4.2 can be separated into two 1-D Gaussian filters at arbitrary orientation φ as in Equation 4.3. In other words, to perform ADF, a 1-d Gaussian filter is convolved with an image on the x-direction followed by an application of a 1-d Gaussian filter in the φ-direction [230].

The ADF boosts the separation of foreground/background regions by steering the filter locally against text-strokes in some specific directions. In this sense, the ADF scale parameters should be set based on measurements of text strokes and gaps of a manuscript. Therefore, widths and heights of main-content words, PAWs, or characters can be used to estimate the proper ADF scale parameters.

For the vertical direction of ADF, the scale can be computed based on the average whitespace gaps of text lines. Hence, the S_x and S_y are computed as follow:

$$S_x = \alpha \times \mu_{H_{PAW}} + \beta \times \mu_{Lgaps} \tag{4.8}$$

where $\mu_{H_{PAW}}$ is the average height of the main-content dominant PAWs, μ_{Lgaps} is the average gap height between text lines, and α, β are two weights to reduce the variability effect of PAWs hight and vertical gaps,

$$S_y = \alpha \times \mu_{W_{PAW}} + \beta \times \mu_{PAWgaps}$$

where $\mu_{W_{PAW}}$, and $\mu_{PAWgaps}$ are the average width of the main-content dominant PAWs and gaps respectively.

The ADF is applied on a range of angles $[10^o \pm \frac{\pi}{2}]$ and $[10^o \pm (\pi \times 2)]$ for vertical and horizontal directions respectively. The application of ADF at this range may boost the vertical and horizontal whitespaces of main-content region against other document regions.

Arabic language is written horizontally from right to left. In historical Arabic manuscripts, written text-notes appear on the margin space around the main-content. In case of dense written notes, the main-content and side-notes may not be clearly separated. Because some text components of side-notes and main-content might be touching at the boundary region. Figure 4.1 shows manuscript pages with different levels of dense text. First example of Figure 4.1.(b) represents a situation of a manuscript that has its left-side of the main-content region touches the side-note. Due to this issue, whitespace analysis may not be suitable for finding a separating path between the main-content and side-notes. Hence, in this situation, ADF may not be successful to find clear cut whitespace at the left-side of the main-content region. However, still some clues of main-content region left-boundary can be estimated from its top and bottom boundaries. In general, left and right sides of main-content region are more exposed to text-touching issue than top and bottom boundaries. This could be attributed to the horizontal writing style of Arabic language, and writers' behavior.

Since left and right boundaries of the main-content region are difficult to be detected, we discuss ADF vertical response with some illustrative examples (see Figure 4.5, first row). Moreover, the effects of ADF scale parameter S_x has been analyzed.

The white shapes in Figure 4.5 (first row) represent text entities in a binary image. Figure 4.5 column (a) shows a long component (in white color) that represents very dense handwriting of the main-content where the text line gaps are very small. On one hand, the max response of the ADF using an estimated S_x from equation (4.8) is showed in Figure 4.5.(a) second row. The ADF response(in magenta color), which appears on the left and right sides of the middle long-component, represents major whitespace locations. However, this major-whitespace leaves wide black gap between the long-component and the ADF response. These black gaps are safe zones that the major-whitespace leaves to allow safe segmentation. The safe zone width is related to the ADF scale S_x parameter. In this example, the safe zone is wide because the estimated scale is large $S_x \approx 153$ pixels.

The safe zone is important to define the main-content boundary that does not interfere with other regions. The desired safe zone should be small enough to surround the main-content region. Therefore, we fix $S_x = 15$ pixels in the second experiment to investigate the effects of the scale on ADF response as shown in Figure 4.5.(a) third row. The safe zones are shrunk and the ADF response appears very close to

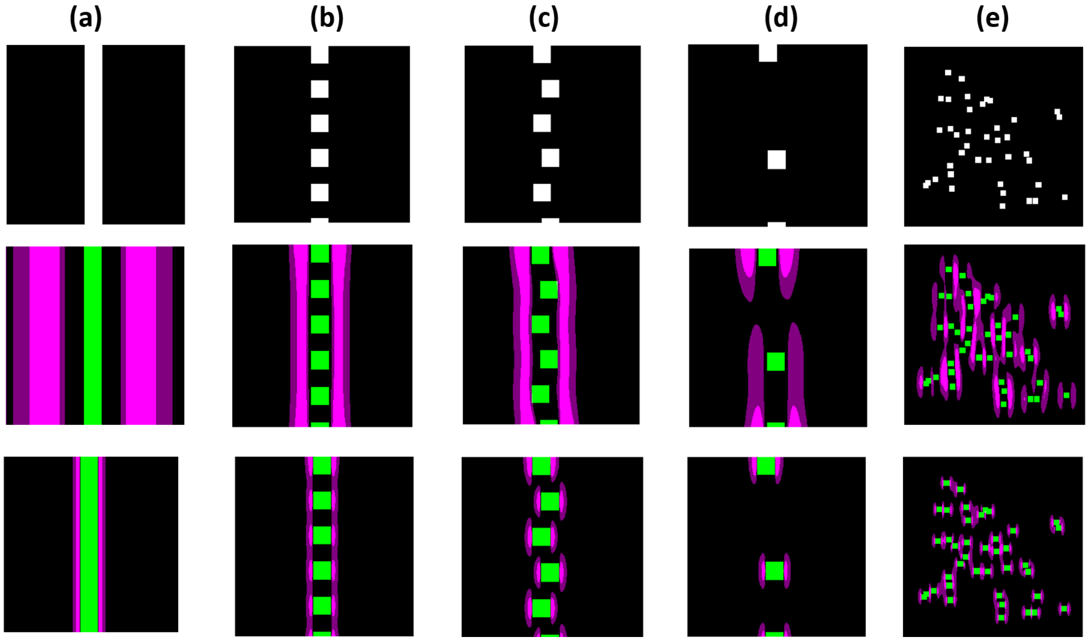


Figure 4.5: ADF illustration on different simulated text situations; a) Touching text lines, b) Normal text lines, c) Regular spaced, not aligned text lines, d) Irregular aligned and spaced text lines, e) Scattered text components

the long-component.

As the first example may not be realistic that all text lines are touching each other(i.e., extreme example). The behavior of ADF is investigated further in other normal situations.

The ADF is applied on several simulated situations where vertical gaps are getting larger as shown in Figure 4.5.(b), as well as white components may not be vertically aligned as in Figures 4.5.(c) and 4.5.(d). The results of these illustrative experiments show that ADF max responses can be preserved along the perpendicular direction of the strong gradient between the edges of foreground and background. Moreover, the estimated S_x based on white components and gaps resulted in better whitespace localization (see Figure 4.5.(b),(c), and(d) second row). Hence, the ADF maximum response can be used to indicate a whitespace separator between different regions in such similar situations.

It is very challenging to find the main content in a situation where the main-content and side-note texts are similar and scattered heterogeneously all over a manuscript page. Figure 4.5.(e) shows the second extreme situation. The ADF filtering shows no major response as in the previous examples. In this situation, the algorithm may conclude that all text components are of one type. To explain this effect further, larger white components are added to the same example to illustrate two types of text situation as shown in Figure 4.6.(a). In this case, the ADF produces major responses that can be used to separate these two types of text (see Figure 4.6.(b) and (c)). Moreover, the results in Figure 4.6.(b) is more suitable for main content

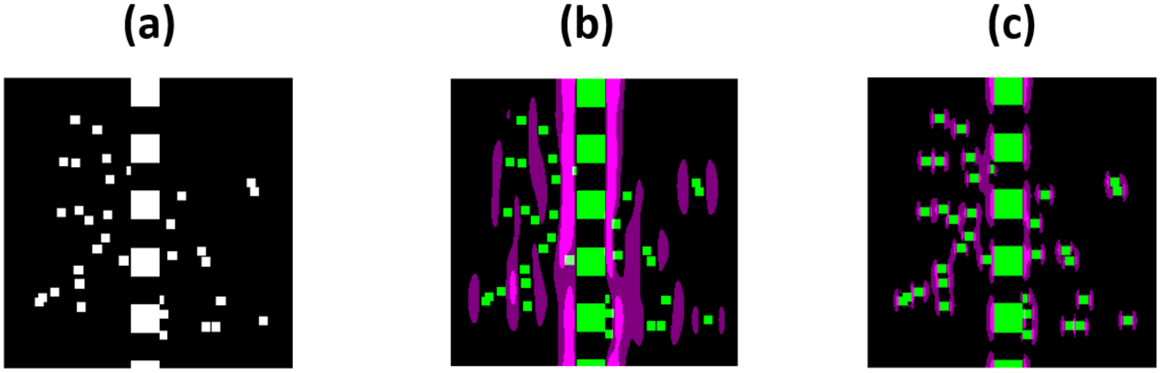


Figure 4.6: An example of two types of white components, **a)** simulated input-text, **b)** ADF response using estimated S_x, **c)** ADF response with Fixed scale $S_x = 15$.

separation due to estimating the scale objectively.

Figure 4.7 depicts the pipeline of the initial main-content detection using ADF. The detection pipeline starts by computing vertical and horizontal ADF responses (see Figure 4.7.(b) first row and second row respectively). In Figure 4.7.(c), these results are thresholded to remove weak ADF responses. Then, an integration step is carried out to combine both responses. In this step, the algorithm finds clues for estimating the hight of the main-content region by locating the longest whitespace responses on the top and bottom of the page. Similarly, it estimates the width of the main-content region by locating the longest whitespace responses on the left and right sides of the page. So, any top or bottom whitespaces that exceed the estimated width of main content region are cleaned, and similarly for left and right whitespaces in relation to the estimated hight of main content region. Formally, Let $Vmask, Hmask$ be the ADF vertical and horizontal responses respectively. The integrated mask $Cmask$ is

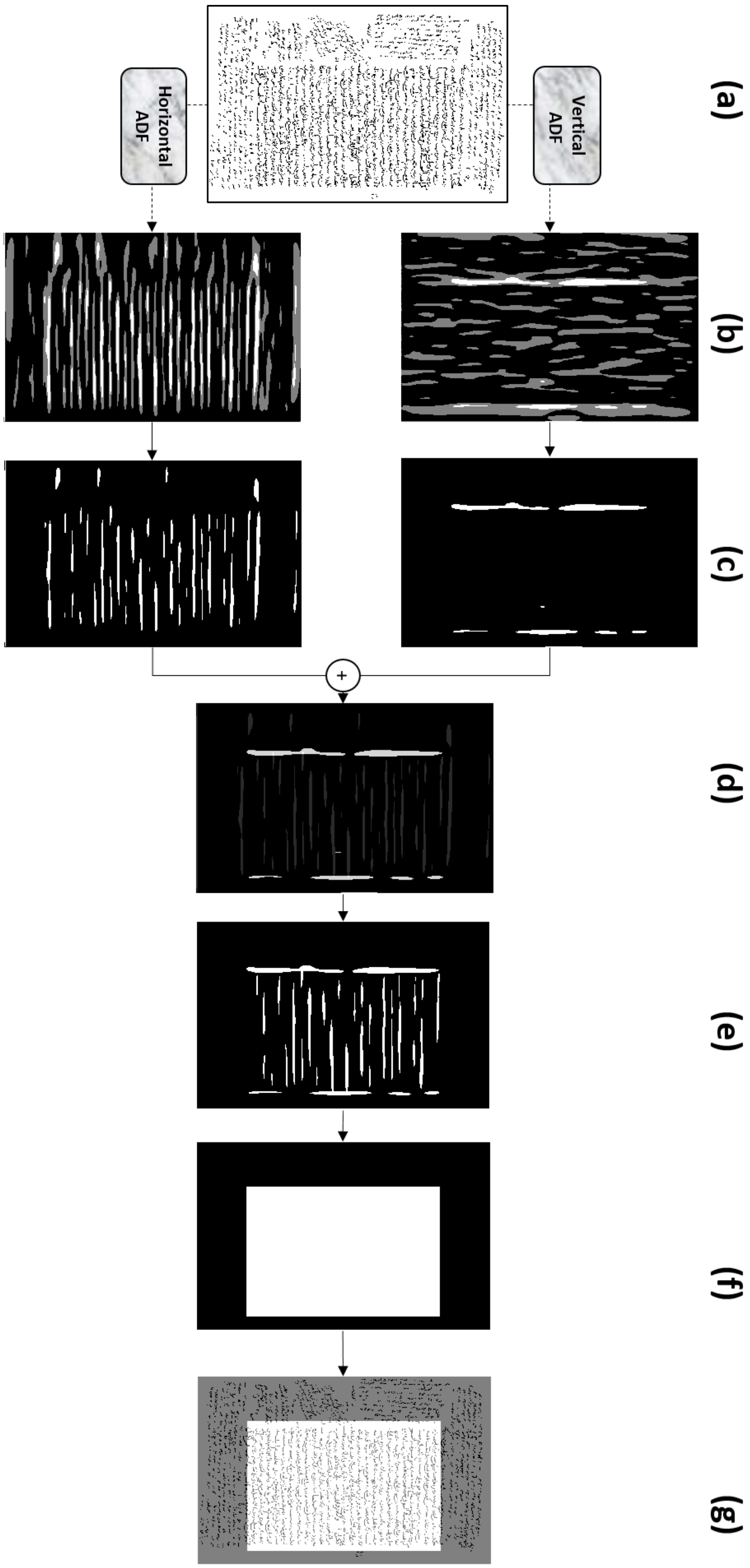


Figure 4.7: ADF main-content detection pipeline. a) An input manuscript; b) Vertical and horizontal responses; c) Removing weak response; d) Integrating vertical and horizontal preprocessed responses; e) Adjust left and right responses based on top and bottom clues and vice versa if needed; f) Generate ADF mask; g) Main-content result

generated as follows:

$$Cmask = \begin{cases} Hmask_i \in Cmask & if\, \min(W_v) \leq Hmask_i \leq \max(W_v) \\ Vmask_i \in Cmask & if\, \min(W_h) \leq Vmask_i \leq \max(W_h) \\ \\ Hmask_i, Vmask_i \notin Cmask & Otherwise \end{cases}$$

where W_h is a major whitespace coordinates of ADF horizontal response, W_v is major whitespace coordinates of ADF vertical response, min(.) and max(.) locate the coordinates of the major whitespace responses. Figure 4.7.(e-f) show the final generated ADF mask.

4.3.3 Static whitespace localization

The ADF whitespace analysis may fail to find the main-content boundary due to the situation discussed in the previous section and shown in Figure 4.5.(e). Therefore, our initial main-content boundary detection includes SWL. The SWL scans through a manuscript page vertically and horizontally. It marks gaps that are greater than a predefined threshold ω_s as whitespace. The scans are on regular fixed intervals ($s = \omega_d$). The aim of SWL scans are to generate whitespace masks that represent general boundaries of the in-between regions horizontally and vertically. Figure 4.8 illustrates an example of the generated SWL masks. The integration of SWL horizontal and vertical masks is computed by multiplying the vertical and horizontal masks.

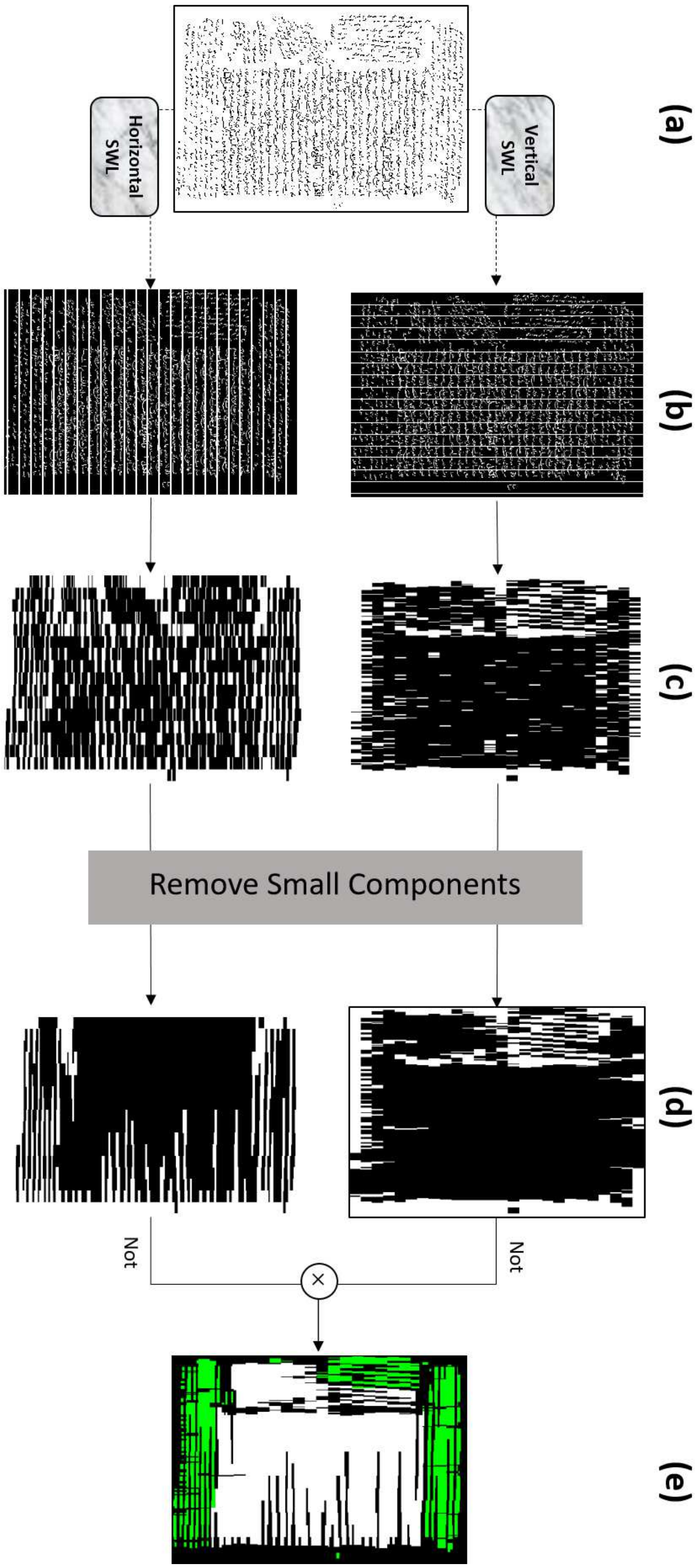


Figure 4.8: Static whitespace mask generator; a) An input manuscript, b) SWL scans; vertical (first row), and horizontal(second row), c) SWL outputs, d) Cleaned outputs, e) Combined vertical and horizontal outputs.

4.3.4 ADF and SWL integration

The resultant masks of both ADF and SWL are integrated to define the main-content initial region. The integration process is considered only if the ADF fails to define the main-content region due to the limitations discussed previously. The failure situation is detected if the ADF responses have no maximum peaks.

Comparing the two techniques, we found that ADF technique is robust at detecting horizontal whitespaces, while SWL is good at detecting vertical whitespaces. This observation is exploited to combine both masks and generate a robust integrated mask. Figure 4.9.(b) shows both vertical and horizontal responses of SWL and ADF respectively. The SWL vertical response helps in defining the width of the mask while the horizontal response of ADF is defining the hight of the mask.

4.4 Feature Extraction

Once the initial main-content region is identified, a set of essential handwriting characteristics is extracted. The writing characteristics are mainly geometric measurements. The algorithm randomly selects eight frames over the initial main-content region. A frame w_f has a square shape and its size is calculated so that at least two text lines can fit inside it. The length of each side is computed as follows:

$$w_f = (\alpha \times W_p) + (\beta \times G_l) \tag{4.9}$$

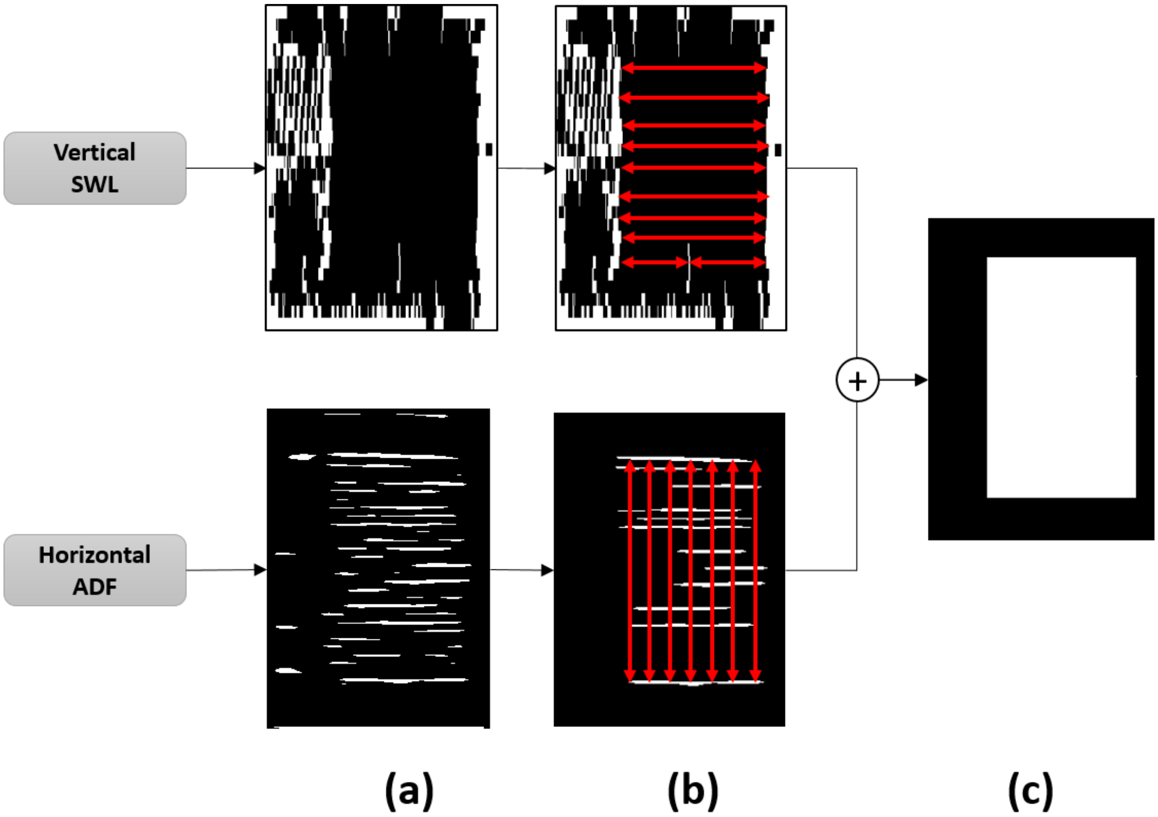


Figure 4.9: An integration mask using ADF and SWL

where α and β are two free parameters set to three and four respectively to create a frame of 3 times the average PWA width W_p and 4 times the average vertical gaps G_l between PAWs.

4.4.1 Geometric features

Eight geometric features are extracted from each frame as follows:

- Average height:

  This feature measures vertical stroke style of the main-content text. In comparison to side-notes, the vertical strokes of the side-notes are smaller due to restricted writing-space. Therefore, PAWs' heights can help in distinguishing main-content vertical strokes from side-notes.

- Average area:

  Since, some side-notes text are written vertically, the height feature alone may not be enough to distinguish main-content elements from side-notes. Therefore, a width feature of these elements is as important as the hight. To handle this confusion and increase the understanding of main-content characteristics, the average area of PAWs is considered as a feature point.

- Foreground to background ratio:

  This feature estimates how foreground pixels are distributed against background pixels in the main-content region. The large writing style of main-content leaves large gaps around, which minimizes the foreground to background ratio.

- Pixel density:

  Computes how dense is the foreground pixels at a given frame. It simply counts the number of foreground pixels located in a given frame. This feature and the foreground to background ratio feature indicate the density of written text in the main-content region.

- Average horizontal gaps:

  This feature estimates the regular horizontal spacing among the main-content's PAWs. Although the text in the main-content is handwritten, still a writer can maintain reasonable spacing between words. Usually, the main-content horizontal spacing differs from side-note spacing due to writing conditions and style.

In the main-content, a writer has the whole page to start writing comfortably, while in side-notes a writer has a small area to write.

- Average text line gap:

  The text-line gaps have some regularity in the main-content region than side-notes. Therefore, the average vertical whitespace between text-lines is considered as a region feature.

- Distance transform(DT):

  It computes a binary-to-gray value of each foreground pixel to its nearest neighbor foreground pixel. So, for each foreground pixel FP_i, the DT finds a linear location of its closest foreground pixel FP_j, and stores FP_j location as a pixel value in FP_i [209]. In other words, it transforms a binary FP_i value to a location value $\in \mathcal{R}$. This feature computes the transformed intensity of the main-content text. Figure 4.10.a shows DT response of the main-content text in comparison to DT response of side-notes text in Figure 4.10.a-b.

- Text orientation:

  The estimation of the main-content text orientation is computed locally. For each frame, all PAWs centroids are determined. Then, the algorithm finds the right neighbors of each PAW. Let v_{nm} a vector between two neighboring PAWs n and m, and v_h is a reference horizontal vector on the x-axis. The orientation

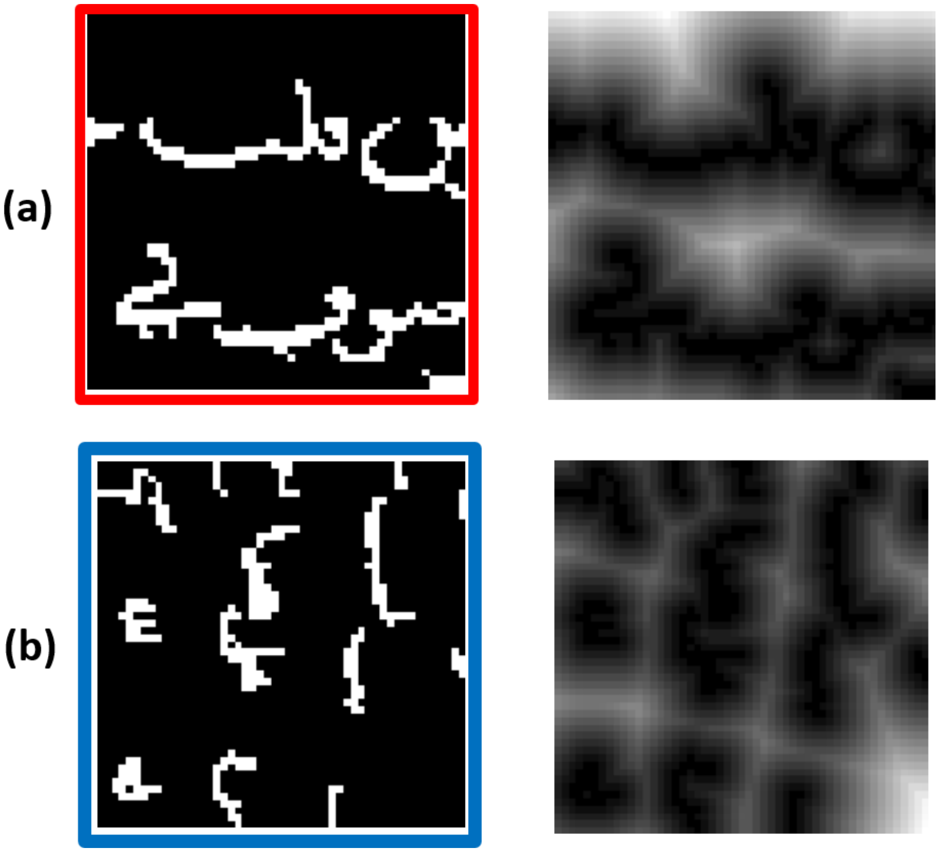


Figure 4.10: Distance Transform response. a) A patch taken from Main-content region; b) A patch taken from side-notes left part

is estimated by computing the angle between the two vectors as follows:

$$\theta = \cos^{-1}\left(\frac{v_h \bullet v_{nm}}{\|v_h\| \times \|v_{nm}\|}\right)$$

where $\bullet$ is the dot product of two vectors and $\|.\|$ is the magnitude of the vectors.

4.5 Moving window analysis

The segmentation of the main-content text has two main phases; global analysis that coarsely defines an initial main-content region, and local analysis using a moving-window approach. The moving window uses dynamic sizes that are estimated using Equation 4.9. Algorithm (1) describes the moving window analysis main steps. The moving window analysis is detailed in the following sub-sections.

4.5.1 Feature based main-content segmentation

The initial main-content detection is error prone, it may fail to find the left or right boundaries due to an absence of enough whitespace gaps between the main-content and side-note regions. In this case, neither SWL nor ADF can yield perfect mask of the main-content region. However, the main-content characteristics can be extracted from the estimated main-content region.

To address this issue, local analysis using moving windows is employed to define the manuscript main content. Three windows centered at the initial main-content region are moved towards left, right, up, and down directions of a manuscript page. The algorithm stops the moving windows analysis at a particular direction if two of the windows have met stop conditions.

There are four conditions to stop a moving-window analysis, namely; blank-zone, off-boundary, transition-zone, or different-zone. Figure 4.11 shows an example of each stopping condition. A blank-zone and off-boundary conditions are intuitive stop conditions. Once a moving window steps on a large empty background, or reaches borders of a manuscript page it must be stopped. In other words, the processed manuscript page must contain only main-content text in that particular direction.

As features are extracted from the windows on each move, a score CS_i is computed

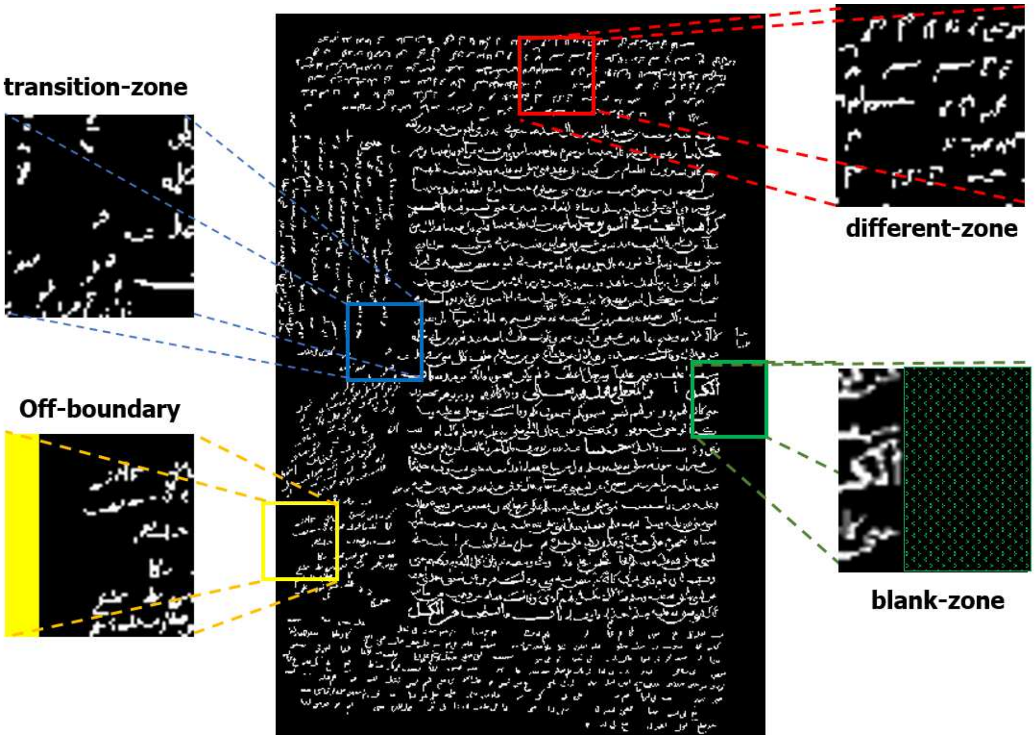


Figure 4.11: Moving Window stop conditions.

by matching these features of the current windows with the initial predefined features of the main-content region using Euclidean distance. If the score $CS_i > th_1$, then, the current moving window i is marked as a stop window because of a reasonable change in its characteristics compared to the predefined characteristics of the main-content region. Similarly, for the last stop condition, but it uses another pre-computed threshold th_2. Thresholds th_1 and th_2 are empirically computed in the initial step of manuscript characterization.

Illustrative stop windows are shown in Figure 4.11. An example of moving window analysis on a manuscript page is shown in Figure 4.12.(a). In this example, only the right moving-window has stopped because of a blank-zone condition, and for other directions, the moving windows are stopped because of the transition-zone condition.

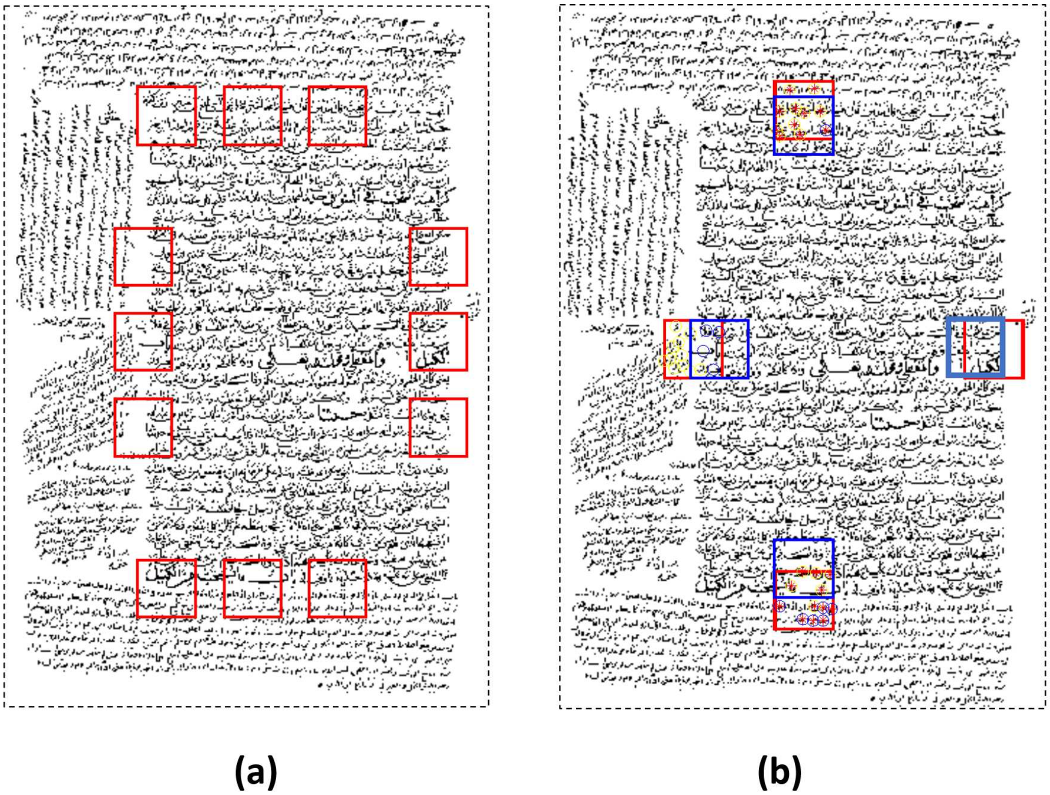

(a) (b)

Figure 4.12: Main-content region boundary detection; a) Moving stops by different conditions, b) Stop window correction

4.5.2 Local main-content/side-note analysis

Once moving windows reach stop conditions, the algorithm finds possible boundary cuts that separates main content from side notes. The off-boundary, and blank-zone stop conditions are corrected by moving the stop windows back and return coordinates of boundary foreground-pixel locations. In other words, if the stop window is located on the right direction, then, the algorithm returns the rightmost foreground-pixel coordinates. Similar procedure is carried out for other directions.

In Figure 4.12.(a), left, upper, and lower stop windows are stopped because of

Algorithm 1 Main-content boundary detection

Input: Feature vector F_{ini} and manuscript I_c **Output:** Four stop windows

```
procedure MOVING WNIDOW ANALYSIS
 Loop for directions (Right, Left, Up and Down):
Loop while not(off-boundary)
    Winfeat_i ⟵ getNextWindow($I_c$, step)
    score ⟵ Euclidean (Winfeat, $F_{ini}$)
    if ( $score > th_1$ ) then
        FlagTransition ⟵ True
        return StopWin ⟵ Change in characteristics
    else
        if ($score > th_2$) then
            FlagDifferent ⟵ True
            return StopWin ⟵ Different characteristics
        else
            if ($score = \phi$) then
                FlagBlank ⟵ True
                return StopWin ⟵ Blank-zone
Loop while (until off-boundary)
Loop next direction
```

transition-zone condition. In transition-zone condition, it is more challenging to define a boundary between the main-content and side-notes. The whitespace gaps in these stops could be very tight. Therefore, the algorithm performs unsupervised classification of the local connected component.

First, the algorithm extracts six geometric features per connected component. The features are the hight, area, foreground/background ratio, pixel density, distance transform, and orientation. Then, K-nearest neighbor algorithm is used to find two classes of components; main-content or side-notes. Figure. 4.12.(b) shows the clustering results on left, upper and bottom stop windows. After that, a boundary cut is defined on the mid distance between the clusters' centers as follows:

$$S_d = \frac{dist(Cn_i, Cn_j)}{2}$$

where Cn_i and Cn_j are the clusters' centers of the main-content and side-notes, and $dist(.)$ is the Euclidean distance. Finally, the stop windows are corrected by moving them back S_d pixels. Figure 4.12.b shows the three stop windows (in red color) have been moved to their new coordinates(in blue color).

Main content segmentation

The algorithm converts corrected stop windows to boundary points. It selects foreground pixels of PAWs along the boundary of the main-content region. Figure 4.13.(a) shows an example of the selected boundary points. As the number of the selected points is large, the algorithm reduces them and identifies the end-points using a convex hull algorithm. Figure 4.13.(b) illustrates the convex hull points in red color. After that, a convex mask of the main content is generated by tracing these points. Finally, the manuscript's main content is segmented using connected components. Each connected component that have 80% of its foreground in the convex region is labeled as main content, otherwise it is labeled as side-note. Figure 4.13.(e) shows an example of the main content segmentation.

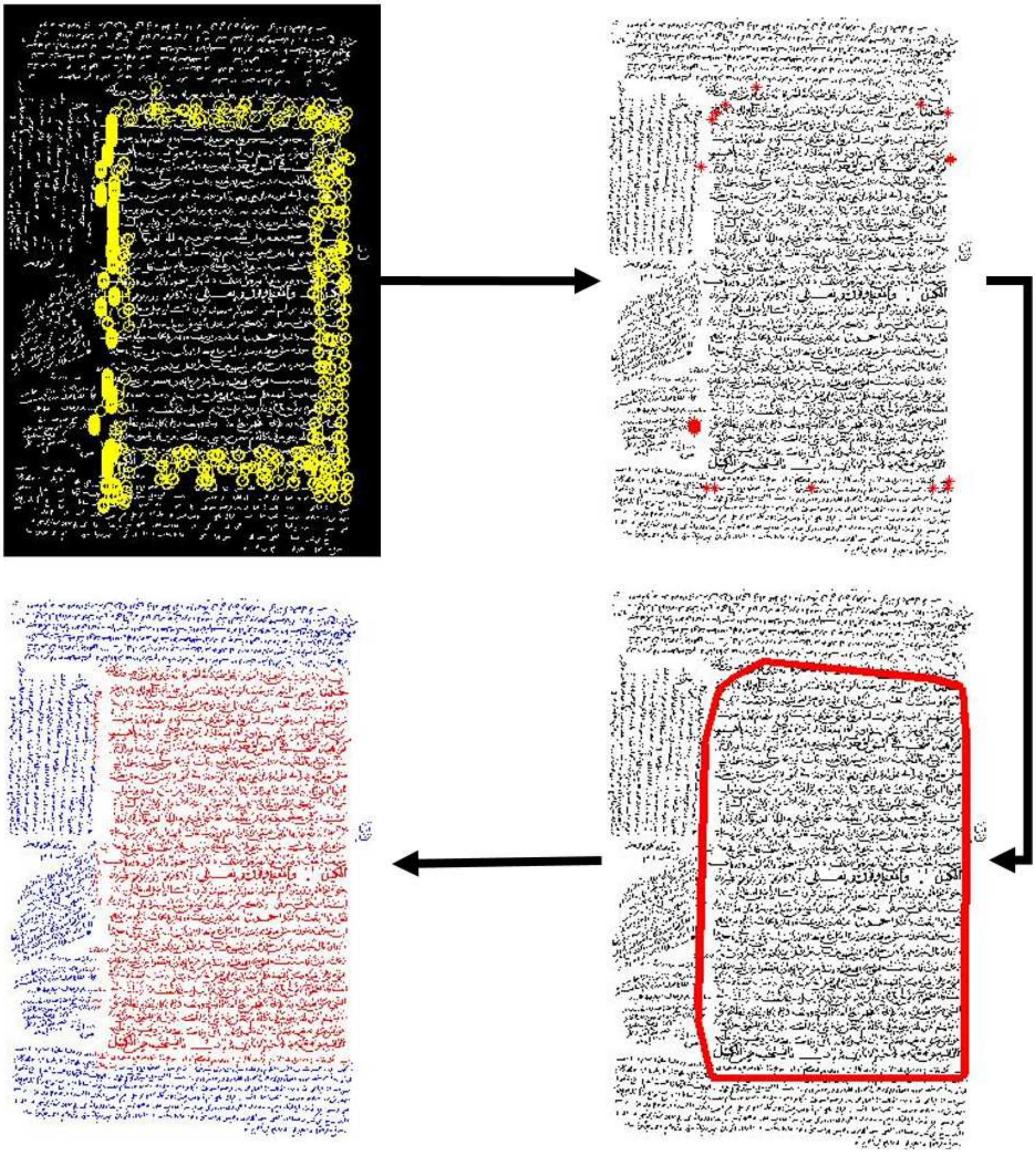


Figure 4.13: Main-content segmentation

4.6 Conclusions

In this Chapter, a learning-free hybrid DLA algorithm is presented. The hybrid algorithm is designed to address Arabic historical manuscripts that are mostly text. The main aim is to locate and extract the main contents of the manuscripts that can be used in categorization and indexing applications. The algorithm has four main steps; **1)** A preprocessing phase that computes a binary version of each manuscript and cleans them from noise, **2)** Manuscript characterization phase which detects the main region initially to extract a set of features, **3)** Feature extraction phase; geometric features are extracted to describe the content and writing style of a manuscript's main-content, **4)** Moving window analysis that extracts the manuscript main content.

The proposed method integrates whitespace analysis (top-down strategy) to define initial region of interest and analysis parameters. The whitespace analysis is usually adopted in printed and clean document layouts. However, by introducing anisotropic diffusion in this work, the whitespace separation among regions are boosted, and hence regions can be detected. The application of ADF can be extended further to analyze contemporary documents and extract document blocks such as figures, text paragraphs, text columns etc. which makes the proposed method scalable.

Finally, the detected main-region is verified and the segmentation is enhanced by moving window analysis (bottom-up strategy). The moving window procedure utilizes the automatic parameters that are computed in the previous phases to extract the manuscript main content.

CHAPTER 5

DOCUMENT CLASSIFICATION BY KEYWORD SPOTTING

Information retrieval of scanned handwritten documents is becoming vital due to the rapid increase in digitized documents. This process is challenging in the domain of historical manuscripts retrieval due to document degradation that are preventing the extraction of accurate transcripts. Keyword spotting systems are developed to search for words within scanned documents usually without word-transcription. These systems can be either template matching or learning-based algorithms.

In this chapter, we present a learning-based keyword spotting technique using the word-skeleton and Speeded-Up Robust Feature (SURF) descriptors. The proposed method detects interest points of handwriting using the word-skeleton. Unlike SURF detector that could locate interest points off-writing strokes due to ink-bleeding, background noise, or auxiliary diacritics etc. In word-skeleton, the interest points are located by tracing the word skeleton and identifying important interest points locally. Therefore, it selects interest points on main handwritten strokes. Then, these selected interest points are described by Bag-of-Visual-Words (BoVW) powered with SURF features. After that, a set of support vector machines (SVM) are trained and validated using a set of keywords. Then, the behavior of these SVMs is modeled by observing their responses on matching, mismatching, and rejection decisions. Finally, the three models are analyzed to find the proper decision thresholds for keyword spotting.

5.1 Introduction

Recently, the amount of digitized documents, which includes printed and handwritten documents, is tremendously increasing. Printed documents are structured and easy to be indexed, retrieved and stored using standard tools such as OCR as they are usually represented by their transcriptions [10]. On the other hand, handwritten documents are hard to process because of the variations in writing and issues related to their writing styles. These issues may impede transcription extraction methods. Therefore, keyword spotting systems(KWS) are usually adopted to search and retrieve required documents based on queried information [210].

The KWS is a task that performs retrieval of all word instances of a particular keyword query in a collection of documents. Hence, classifying those documents that contain word matches to the queried keyword. Usually, this process is carried-out without document transcription because of manuscript degradations that are caused by aging, handwriting styles, unrecognized fonts, unstructured writings, etc.

The proposed approach contains two main modes, namely, configuration, and operation modes. In the configuration mode, the approach trains the support vector machines (SVM) and models its decision behaviors. This modeling yields three distributions of matching, mismatching, and rejection SVM responses. Then, based on Bayesian decision theory, the matching distribution is analyzed against mismatching and rejection distributions to estimate three spotting thresholds. The modeling is done off-line to reduce keyword spotting time in the operation mode. Figure 5.1 illustrates a general overview of the proposed KWS framework.

The main contributions of this work includes:

A. Analyzing keyword's interior structure and locating interest points using keywords' skeletons. Unlike automatic detectors such as SIFT or SURF, which are designed originally to detect salient regions of natural scene objects. In other words, detecting interest points using automatic detectors on handwritten objects may select undesired interest points. Figure 5.6 shows interest points selected by the different approaches.

B. Proposing a novel approach to estimate spotting thresholds based on modeling the SVMs matching, mismatching and rejection behaviors.

C. Investigating the integration of skeleton , SURF, and dense key-points sampling for keyword spotting and word recognition tasks.

The rest of this chapter is organized as follows: Section 5.2 discusses related works. The proposed KWS is outlined in Section 5.3. In Section 5.4 the conclusions and closing remarks are presented.

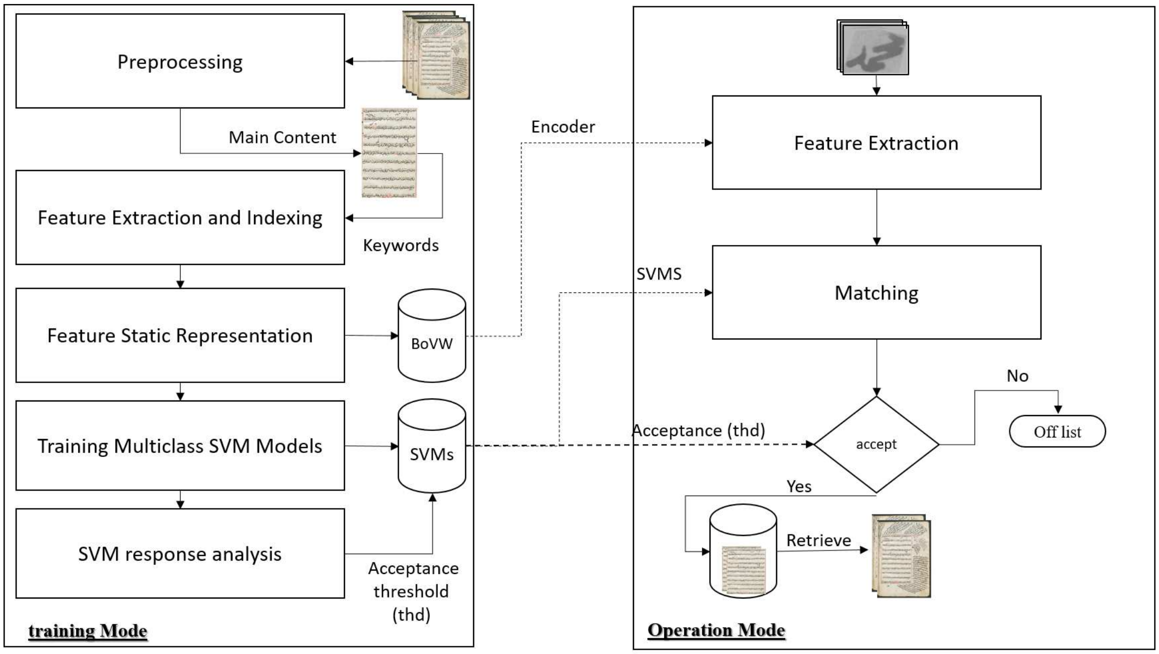


Figure 5.1: Overview of the proposed Keyword Spotting

5.2 Related Work

In general, there are two main types of features that are extracted from a word image for KWS; global and local. The handwritten words are incorporating large variabilities that make the design of global feature extraction, to address KWS, a hard task. Due to that, most of the studies in the literature suggested local feature extraction techniques. Scale Invariant Feature Transform (SIFT) [211], and Speeded-Up Robust Features (SURF) [7] are among the well-known and frequently used features and interest point detectors in computer vision.

SIFT is successfully used in computer vision applications such as image analysis, categorization, retrieval, and recognition[212, 153]. Recently, computer vision based features are borrowed into handwritten recognition and spotting tasks [11]. It has two sub-tasks; feature detection and description.

The detection in SIFT employs multi-level Laplacian of Gaussian (LoG), and is implemented using Difference of Gaussian (DoG). To reduce the computations required for calculating LoG, in some applications, dense key-point sampling is used instead. In dense sampling, the image is divided into regular patches and the center of each patch is treated as a key-point [213].

In feature extraction, SIFT descriptor computes weighted Histogram of Gradients (HoG) by dividing each key-point patch into (4×4) sub-regions and quantizing their gradient orientations into eight bins. Then, the weighted gradients of each sub-region are accumulated into a histogram of eight bins. So, SIFT feature vector will have $16(blocks) \times 8(feature \quad values) = 128$ feature points in total.

The discriminant power of SIFT has attracted researchers to address KWS using SIFT approach. For instance, a segmentation based KWS using SIFT detector and descriptor is suggested in [212]. Their algorithm divided each word image vertically and used SIFT to detect and describe interest points on these sub-images. Then, Euclidean distance was used for interest point matching. Also, SIFT was used for printed document retrieval in [214]. The later method may suffer from detecting different number of interest points in the template-images against queried word-images. Therefore, simple matching may not be proper for keyword spotting.

In some cases, the SIFT detector is used alone with non-SIFT descriptors because of its sensitivity towards noise (see Figure 5.2). In general, the detector locates salient written zones on keyword image as in [215, 216]. These methods relied on SIFT to pinpoint possible word-match zones on documents directly without segmentation.

Hence, they avoided processing a document word by word. Even though this method reduces KWS searching time via employing fast localization of possible word matches, it may skip words that match the queried word. Thus, the spotting task, according to definition, is not fulfilled. Another study that involved SIFT detector and used a similar framework of [216] with document dependent local features is reported in [213]. Due to writing variations, it is arguable that invariant properties of SIFT may not be desired to describe local features. The authors in [216] have shown that document dependent local features can outperform SIFT.

SIFT detector is one important part of the algorithm that automatically detects salient image regions. However, this characteristic of SIFT detector may suite object recognition more than handwritten recognition due to the dynamic variations in handwriting [217]. Therefore, some methods have replaced the SIFT detector by dense sampling as in [218, 219, 220].

In [218] local features such as Histogram of Oriented Gradients(HOG), SIFT and Local Binary Patterns (LBP) were extracted from densely sampled key-points. Since the number of key-points can vary depending on the size of the keyword image, the authors suggested normalizing the image sizes and extract the same number of key-points from each image. Then, their algorithm performed keyword matching using two-directional Dynamic Time Warping (DTW). The study reported that the HOG features had the highest mean average precision in comparison to others. In the same direction, Gradient Local Binary Patterns (GLBP) was proposed for automatic keyword spotting in handwritten documents [219]. The GLBP is a gradient feature that

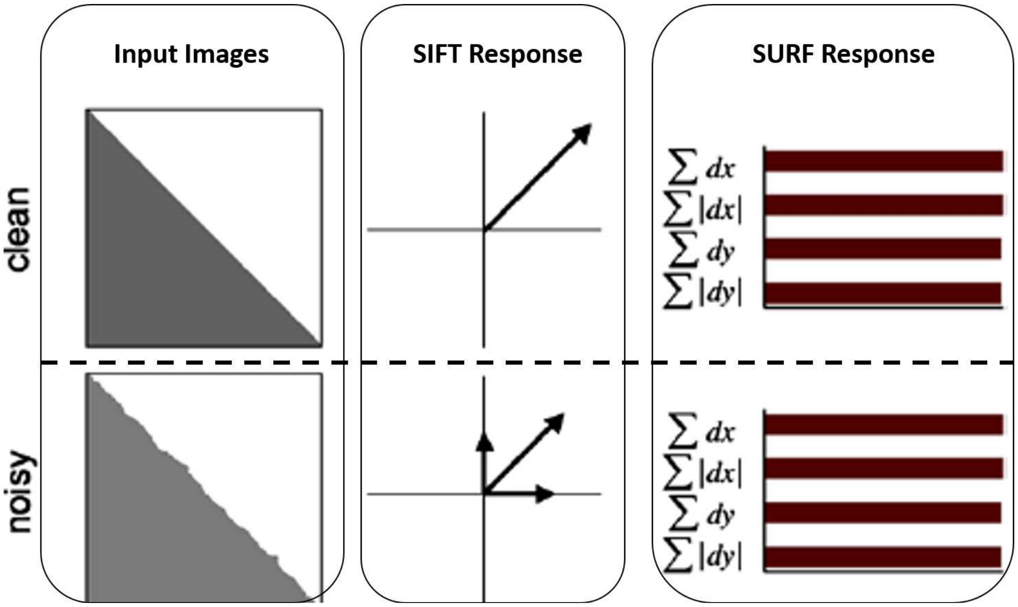


Figure 5.2: Comparison between SIFT and SURF responses [7]

computes the gradient information at the transitions of the Local Binary Pattern (LBP) code.

Although SIFT approach is reported successful for many applications, researchers are concerned about its speed in KWS. This issue involves two aspects, the number of detected key-points and the descriptor space. A recent study reported in [217] suggested reducing SIFT feature space to half-size via considering each symmetric gradient orientation into one bin for example, -90^o and $+90^o$ are placed in the same bin. They reported that the handwritten recognition system using half SIFT features space can perform at the same level of full SIFT features. Although this idea is similar to unsigned gradient of HOG reported in [172], their observation is important to study the modifications to local features adopted in handwritten applications such as spotting.

Bag of Visual Words (BoVW) is another solution to the variation in the number of key-points detected in word images. The BoVW framework represents images by frequencies of occurrences of their local features quantized into fixed number of code-words. There are several examples of BoVW powered by SIFT for keyword spotting such as [221, 222, 223].

The BoVW model has numerous important advantages such as increasing robustness to occlusion, image deformations and provides invariance to changes due to using local features. Moreover, the framework of BoVW model is simple and allows fast keyword matching. Several studies reported good performance of BoVW in comparison to other complex spotting algorithms [223, 144].

In this chapter, we propose keyword spotting for historical handwritten manuscripts using BoVW framework powered by local robust features. The proposed framework follows the Query By Example (QBE) paradigm. In feature extraction, related key-points are sampled using word skeleton. The skeletonized word-image allows localization of important word interest points that includes word joint-points, end-points, and connected paths between them. These areas of a word image contain informative writing behavior, and emphasize interior writing styles. Unlike the study in [220] which extracts shape context features using word skeleton. Our proposed approach locates skeletonized interest points to enhance local feature extraction. We investigate skeletonized interest-points using SURF descriptor. Therefore, we compare the proposed approach to SURF detector and dense sampling.

We adopt SURF descriptor as local feature extraction. The selection of SURF in this work has two advantages; **1)** It extracts half the size of SIFT features space, **2)** It is less sensitive to shape deformations (see Figure 5.2). In addition, SURF is rarely studied in the domain of keyword spotting. In the previous literature, we found few studies that compared the performance of SURF to other techniques for keyword spotting as in [155, 224].

5.3 Proposed word spotting system

The proposed KWS algorithm has two main modes; configuration, and operation. Each mode consists of three main phases; 1) preprocessing, 2) feature extraction and representation, 3) matching.

In the configuration mode, the preprocessing includes locating and segmenting main content of each manuscript page(as described in Chapter 4), and selecting keywords (as described in Chapter 3). The feature extraction phase is responsible for describing salient image key-points using SURF descriptor. Then, BoVW is used to define fixed length representation of SURF features. The BoVW framework has a key impact on generating trained SVM models for better keyword matching [156]. The main steps of the proposed KWS framework is illustrated in Figure 5.1.

5.3.1 Feature Extraction

In this subsection, detailed feature extraction and representation including interest region selection are presented.

Speeded-Up Robust Features

SURF is a computer vision algorithm designed to detect and describe salient image regions in fast and robust sense [7]. Unlike SIFT, the SURF relies on image integrals and Hessian matrix for computing salient points and Haar wavelet transform for feature description. Both SURF detector and descriptor are discussed below.

A. **SURF Detector**

The integral images allow fast computation of box type convolution filters [225]. The entry of the integral image $I_{int}(X)$ at a location $X = (x, y)^T$ represents the sum of all pixels in the input image I within a rectangular region formed by the origin and X,

$$I_{int}(X) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j).$$

Figure 5.3 illustrates the calculation of the image integral and shows that it can be reduced to a summation of four points for each image block. Given a point $px = (x, y)$ in an integral image I_{int}, the Hessian matrix is approximated by computing the convolution of the Gaussian (with scale σ) second order derivative in horizontal direction $D_{xx}(px, \sigma)$, vertical direction $D_{yy}(px, \sigma)$ and diagonal directions $D_{xy}(px, \sigma) = D_{yx}(px, \sigma)$ with an image I_{int}.

$$D_{xx}(px, \sigma) = \frac{\partial^2}{\partial^2 x} g(\sigma) * I_{int}(px),$$

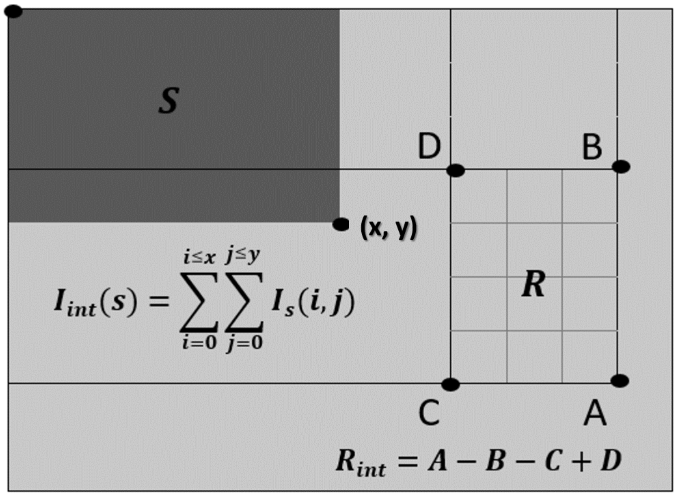

S
D
B
(x, y)
R
$I_{int}(s) = \sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I_s(i,j)$
C
A
$R_{int} = A - B - C + D$

Figure 5.3: Integral image calculation

$$D_{yy}(px,\sigma) = \frac{\partial^2}{\partial^2 y} g(\sigma) * I_{int}(px),$$

$$D_{xy}(px,\sigma) = \frac{\partial^2}{\partial^2 xy} g(\sigma) * I_{int}(px)$$

This calculation is done using convolution of the source image with various Gaussian-related box filters for fast calculation (as shown in Figure 5.4). A Hessian matrix $\mathcal{H}$ is approximated as follows:

$$\mathcal{H}_{approx}(px,\sigma) = \begin{bmatrix} D_{xx}(px,\sigma) & D_{xy}(px,\sigma) \\ & \\ D_{xy}(px,\sigma) & D_{yy}(px,\sigma) \end{bmatrix}$$

Basically, the Hessian determinant decides whether the current point at location px of the image is a key-point if and only if the computed determinant by px

shows a high peak in comparison to its neighborhood points. The Hessian determinant by px is calculated by:

$$det(\mathcal{H}_{approx}(px, \sigma)) = D_{xx}D_{yy} - (w \times D_{xy})^2$$

where w is a relative weight to balance the expression of the Hessian's determinant.

To provide multi-scale analysis, SURF detector is computed at different scales by scaling up the box filter, instead of down sampling the image. A description of SURF detector is given in the following Algorithm.

B. **Dense key-point sampling**

Dense key-point sampling technique is adopted successfully in several applications such as [226, 227] including document retrieval [223]. It simply divides the image into several sub-regions in size of (4×4) or (8×8) with or without overlapping. The center of these sub-regions are used as key-points. Figure 5.6

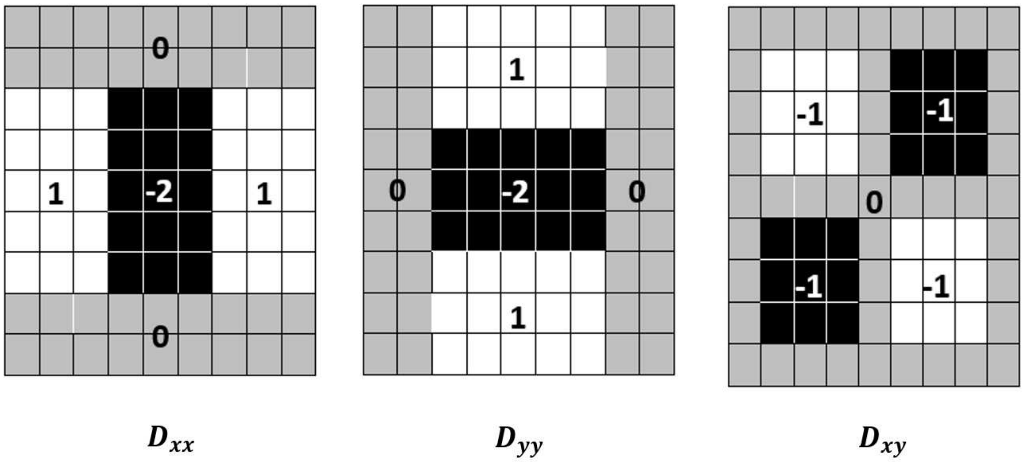


Figure 5.4: Gaussian approximation (Box filter)

Algorithm 2 SURF Detector [7]

```
procedure DETECTOR
    Calculate integral image $I$
Loop for each pixel $px$ at octave $i$
    Calculate $D_{xx}, D_{xy}, D_{yy}$ using box filter
    Normalize responses
    Calculate determinant of Hessian matrix
    if determinant > Threshold then return $px$ as key-point
Loop next
    Suppress non maximum key-points
    Interpolate keypoints between octaves
```

shows example of dense key-point sampling. Usually, the number of key-points is large which normally affects the algorithm speed. Moreover, some key-points may lie outside the handwritten region, and they might be useful in some applications to recognize similar locations or behavior. In document analysis, these off-writing keypoints might be informative to capture writing style and font types [223].

C. SURF Descriptor

Unlike SIFT descriptor, SURF features describes the change of intensities around each keypoint region in square neighborhood. It builds a feature space using the first order Haar wavelet transform in x and y directions rather than the Laplacian of Gaussian (LoG) used in SIFT. It exploits integral images for speeding up the calculation of Haar responses, and produces only 64 dimensional feature space which is half the size of SIFT feature space. Moreover, key-point orientations in SURF are optional and application specific which means, some applications may ignore them [7]. Hence, a reduced time for feature computation and matching.

To extract SURF features, it constructs a square region centered around a key-point with size of (20 × 20) pixels. Then, it splits the region regularly into smaller sub-regions of (5 × 5). Finally, Haar wavelets is convolved with each region and weighted by Guassian filter to produce its final responses L_x and L_y [7]. After that, it sums up all these responses L_x, and L_y of each sub-region to form the first two entries of the feature vector. Moreover, the sum of the absolute values of these responses $\sum |L_x|$ and $\sum |L_y|$ are computed and included in the feature vector as the third and fourth entries. The absolute values of the responses are representing the polar change in the image intensities. So, each sub-region is represented with a four-valued feature vector $v = [\sum L_x, \sum L_y, \sum |L_x|, \sum |L_y|]$. Therefore, the total length of the SURF feature vector is $4(features) \times 16(block) = 64$ feature points.

It is important to note that SURF is similar in concept to SIFT in focusing on the spatial distribution of the gradient information[7]. Nevertheless, SURF integrates the gradient spatial information, whereas SIFT depends on the distribution of the gradient orientations. This makes SURF less sensitive to noise which may help in handwritten recognition and spotting tasks. Figure 5.5 illustrates the process of feature extraction using SURF descriptor.

5.3.2 Skeleton-based interest points

Skeletonization of a binary object is a reduction of the foreground part of that object into a skeletal remnant that preserves the connectivity of the original object. This

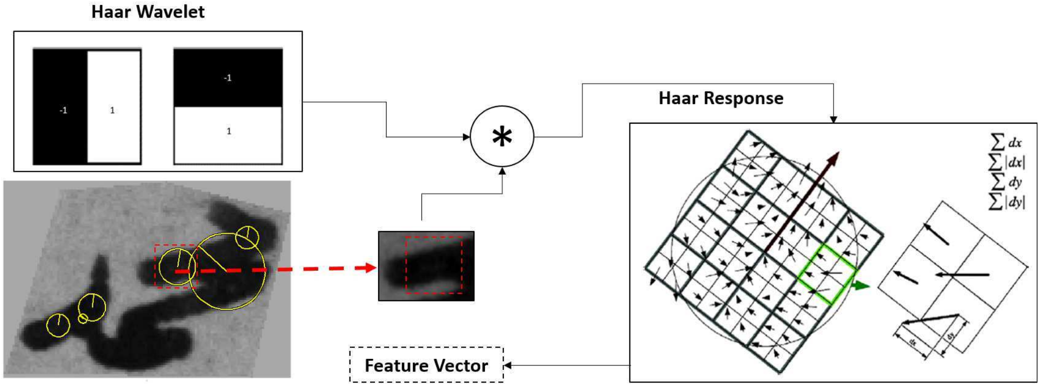


Figure 5.5: Overview of feature extraction using SURF

process is utilized in several spotting and recognition applications [220, 228]. There are two categorizes of skeletonization [229]; iterative and non-iterative. The iterative skeletonization performs two operations iteratively; examination and deletion of contour pixels, while non-iterative produces object skeleton in one pass without examining individual pixels such as [231].

Detecting interest points in handwritten text could be different from detecting them in natural scene objects. In the later case, the objects have less dynamic structures that could be captured in different situations. Therefore, instances of the same scene-object tend to preserve main geometric characteristics. Even though new instances might be captured from different angles, which could occlude that object partially, or the object being transformed, but still the main structure is preserved. On the other hand, instances of a handwritten object are recreated (i.e., re-written) which make their characteristics more dynamic. Therefore, using automatic detectors such as SIFT or SURF may select different interest points on each instance of a handwritten word-image. On the other hand, word skeleton can be used to extract a fixed number of interest points from a word locally.

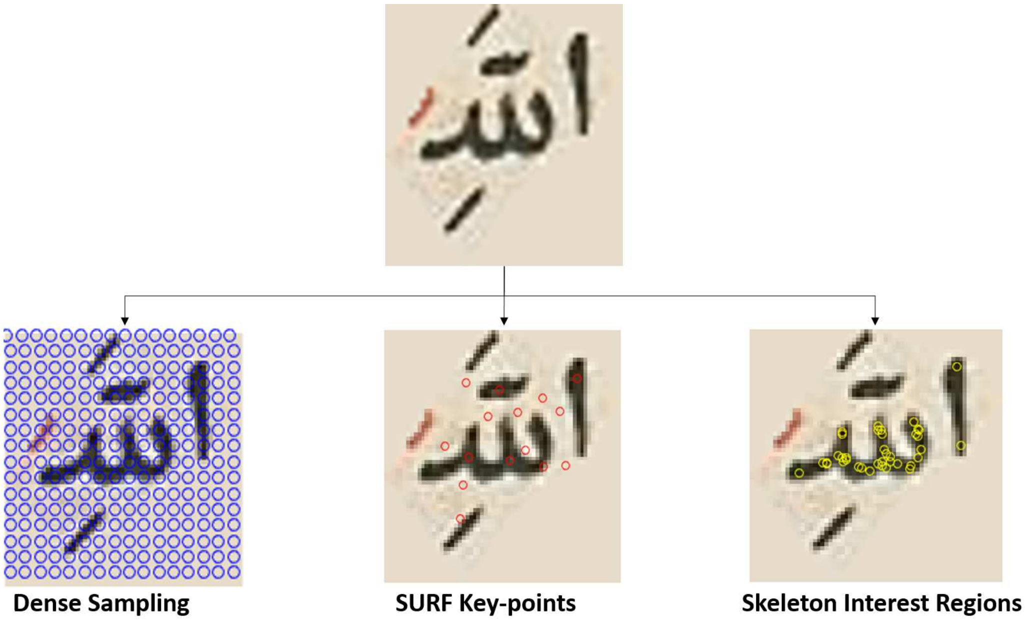


Figure 5.6: Handwritten key points sampling using different approaches

Skeleton-based key-point selection provides meaningful and word-related feature extraction. In addition, it reduces the number of key points per image in comparison to dense key-point sampling. Furthermore, while SURF detector may select off-writing key points due to background noise, all skeleton-based key points are located on the written object. Figure 5.6 depicts SURF, Dense, and skeleton-based key-point sampling. As can be observed from Figure 5.6 the SURF key-points may be detected off-writing areas because of background textures that could appear to be important. On the other hand, dense-sampling divides a keyword image into a grid of regular cells. Each cell is treated as key-point. Finally, skeleton-based method selects two groups of key points essential, and auxiliary points. The essential key-points are located on skeleton's joints and ends, while the auxiliary points are located on the path between skeleton end-points as shown in Figure 5.6.

5.3.3 Feature Representation Using Bag of Visual Words

Often, the number of detected key-points varies from word-instance to another of the same keyword query. This requires dynamic matching procedure. To address this issue and enhance the matching procedure, BoVW framework is adopted to represent features in a fixed size. The BoVW framework has two phases, namely, codebook generation, and BoVW encoding. The codebook is constructed by clustering the extracted features of each visual word in the training phase. Hence, the codebook is a set of cluster centers that represents different visual words i.e., a codeword. On the second phase, given a word image, a set of features are extracted from that image. Then, each extracted feature is quantized (i.e., encoded) to the closest codeword in the codebook. Figure 5.7 depicts the BoVW general framework.

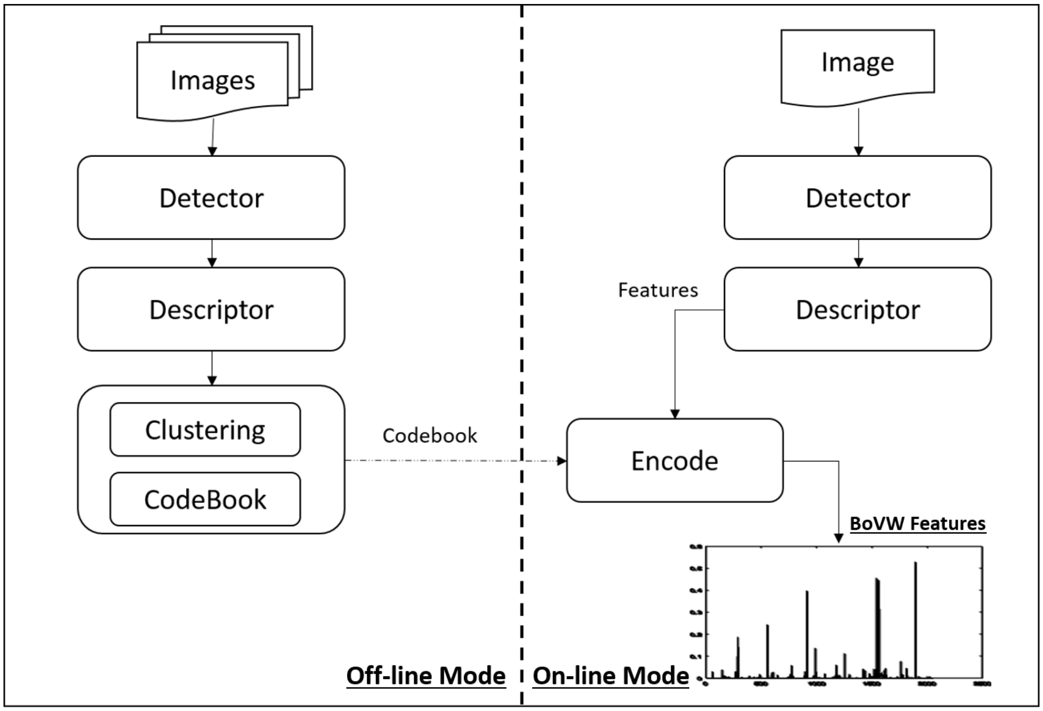


Figure 5.7: Bag of Visual Words framework

5.3.4 Support Vector Machines (SVM)

SVM is a hyperplane discriminative classifier. In this work, we use the algorithm in [233] to solve the SVM optimization problem:

$$minimize \left\{ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{m} \xi_i \right\}$$

subject to

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$$

where vector w and parameter C are controlling the decision boundary's width between the two classes by trade off wide margin between the two classes and small number of margin failures, $y_i \in \{-1, 1\}$, $i = 1, 2, \ldots, m$, and ξ_i are slack variables that are permitting a decision boundary failure, and $\phi(.)$ is a Gaussian kernel function (i.e., Radial Basis Function (RBF)) that has the equation [234]:

$$K(x_i, u) = e^{\left(-||x_i - u||^2\right)}$$

where the x_i is the template sample and u is the testing sample.

5.3.5 Recognition response analysis

In the configuration mode, the proposed algorithm trains multi-class SVMs to build proper models and analyze their matching, mismatching, and rejection behaviors using the keyword images dataset.

First, to select trained SVM models for KWS, the keywords dataset is divided into three sets training, validation, and testing sets. Then, using k-fold cross validation training in word recognition mode, the best three performed SVMs are selected for a spotting task. Figure 5.8 shows a set of SVM models that are fed with various BoVWs with codeword sizes, and the best three performed SVMs are selected (highlighted in bold paths in Figure 5.8).

The spotting approaches often depend on the selection of matching thresholds [210, 223, 11]. Generally, there are two methods to define a spotting threshold; brute-force, and/or objective. In brute-force, a priori rejection threshold is set and

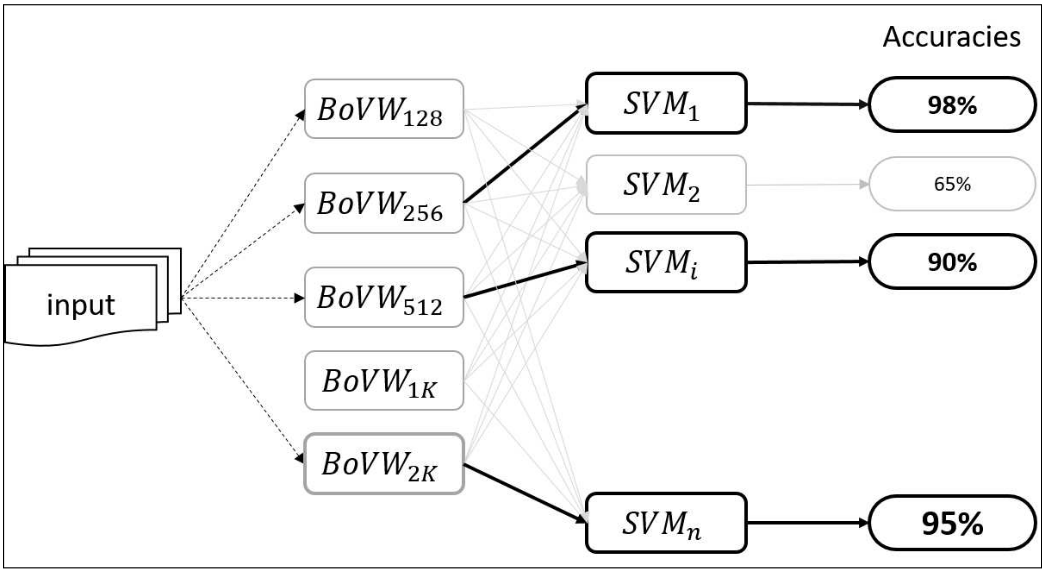


Figure 5.8: Identify the best three trained SVMs for spotting task

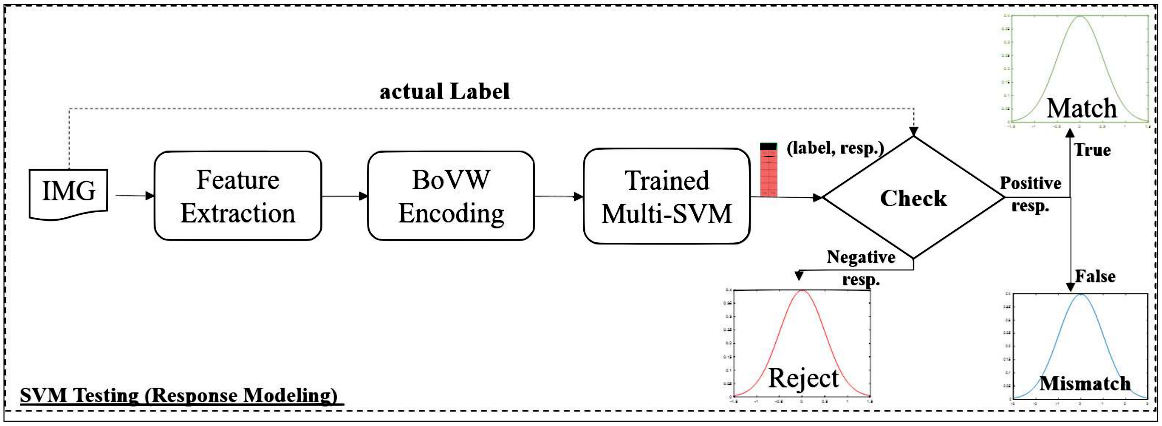


Figure 5.9: SVM recognition and classification behavior modeling

then investigate the classifier behavior on rejection and error rates. This process can be repeated to tune the threshold and set a proper threshold. On the other hand, the second method is objectively defining a threshold-independent measure of performance that possibly leads to a balance between absolute rejection and matching. One example of such threshold identification is based on an Equal Error Rate (EER). The EER is an objective evaluation criterion that defines a point of separation between correctly matching versus rejection scores. It is usually employed in biometric recognition and identification [235].

To learn the matching/rejection thresholds, the response of each wining SVM model are accumulated to form three normal distributions; Matching, Mismatching, and Rejection. Figure 5.9 illustrates the procedure of building these distributions.

The Matching distribution represents the responses that an SVM model produced correct matches of a given word with a template keyword, while Mismatching distribution is representing SVM responses of incorrect matching a word to a template keyword, and the Rejection distribution corresponds to SVM responses of rejected matches. Given these three distributions, three thresholds are computed t_1, t_2 and t_3

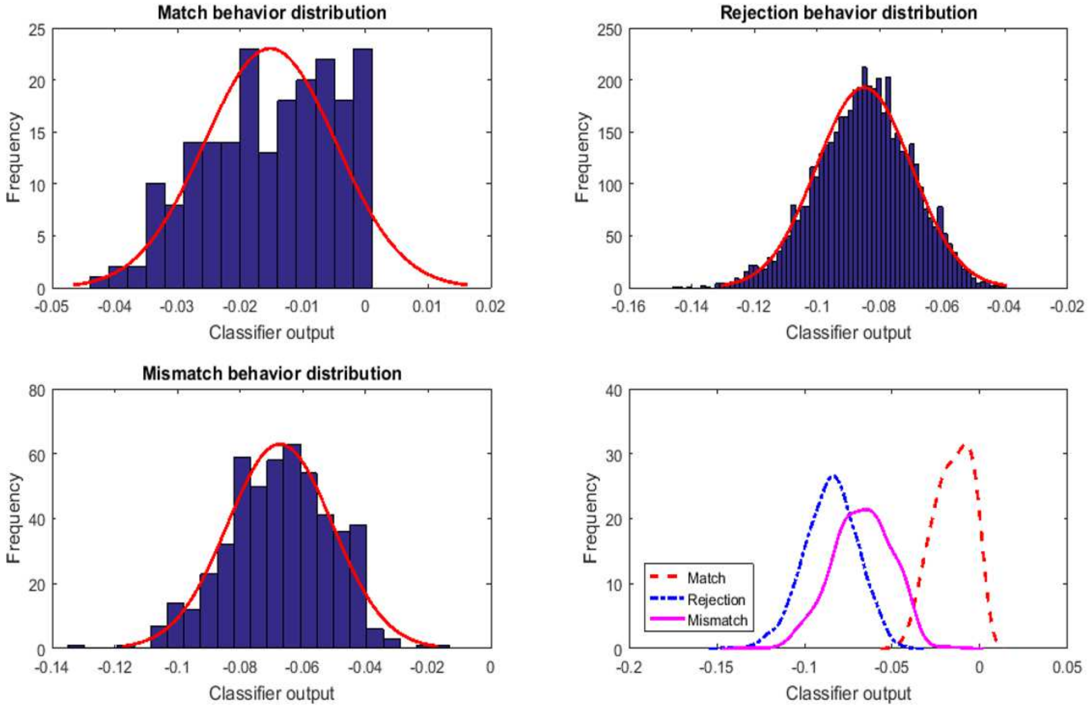


Figure 5.10: An example of SVMs response distributions

to perform keyword spotting. Thresholds t_1 and t_2 are the intersection points located between Mismatching/Matching, and Rejection/Matching distributions respectively. Threshold t_3 is the average value of the Matching distribution. Figure 5.11 shows the visualization of thresholds t_1, t_2 and t_3.

The Matching/Mismatching threshold is more important than Matching/Rejection because in the later, their distributions do not have large overlap. Figures 5.10 and 5.11 show the Matching, Mismatching, and Rejection distributions construction and estimated thresholds. The threshold t_1 is responsible for distinguishing keywords from sub-string words such as HKW04 and HKW19 as they have a common sub-string (see Table 6.1). This confusion is captured by t_1 and it is strictly rejected by t_3 threshold. On the other hand, t_2 can be utilized to conduct sub-string keyword spotting. It shows more tolerance to accept sub-string matches, however, it may incorporate more

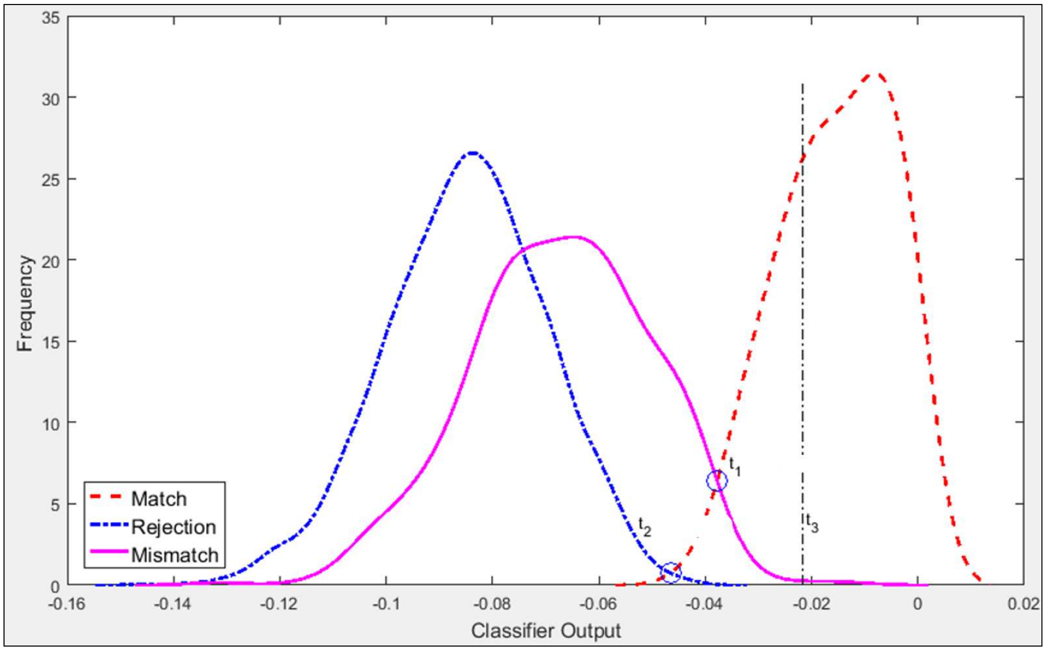


Figure 5.11: Spotting thresholds, t_1 estimated between matching/mismatching, t_2 computed between matching/rejection, t_3 is the matching mean

matching errors. Experiments and performance evaluation of the proposed keyword spotting are discussed in Chapter 6.

5.4 Conclusions

In this chapter, we propose a learning-based keyword spotting system(KWS) for Arabic historical manuscripts. The proposed approach follows the segmentation-based query-by-example (QBE) paradigm. In other words, it accepts a word query image as an input to retrieve word matches.

The proposed KWS has three main tasks; preprocessing, feature extraction and representation, and matching. In feature extraction, SURF descriptor is used to extract features from interest points that were selected using three methods; namely SURF, skeleton-based, and dense sampling. Each of these interest region selection methods has its strengths and weaknesses. For instance, SURF has the fastest selection procedure, but may select interest points that lie off-writing regions. The skeleton-based sampling selects interest points on writing, but requires additional preprocessing steps. Finally, dense sampling selects interest points on and off-writing, and hence results in the longest processing time.

The selection of interest points may vary from one word-image to another. Therefore, different lengths of feature-vectors may be produced. Therefore, BoVW has been adopted for feature representation. The BoVW encodes the extracted features of the three methods into a fixed-length histogram of visual words.

The matching has two main phases, recognition and spotting. In the recognition phase, a set of SVMs are trained, and validated on different instances of the possible keywords. During testing, the matching, rejection, and mismatching behaviors of the SVM classifiers are modeled to estimate possible spotting thresholds t_1, t_2, and t_3. Finally, these three thresholds are used to configure the spotting system at different modes. Threshold t_1 represents a decision boundary between rejection/matching distributions. It has large acceptance range, which means all keywords may be spotted by the system. However, its false positives are expected to be high due to accepting matches with moderate to low similarity. Threshold t_3 is expected to have an opposite behavior compared to t_1, because it is defined as the estimated mean of the match distribution. Lastly, threshold t_2 represents a decision boundary between match/mismatch distributions. Therefore, it is expected to balance between classifier's rejections and acceptances.

CHAPTER 6

EXPERIMENTAL RESULTS

In this chapter, we discuss experimental results of the proposed algorithms for document layout analysis and classification. It has two experimental setups for document analysis and keyword spotting approaches. The experimental setups are given in Section 6.1. The setup discusses adopted datasets and performance evaluation metrics. Performance results of the proposed DLA algorithm is presented in Section 6.2. In Section 6.3, the performance of the keyword spotting algorithm is presented. Further discussions and error-analysis of the algorithms' performance are detailed in Section 6.4. Finally, conclusions and closing remarks are stated in Section 6.5.

6.1 Experimental setup

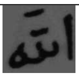
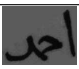
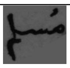
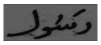
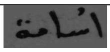
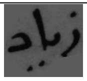
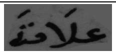
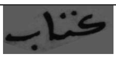
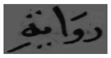
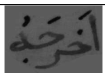
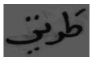
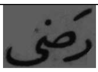
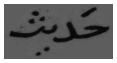
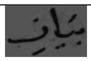
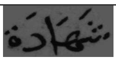
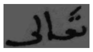
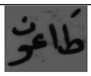
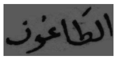
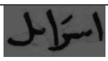
To evaluate the performance of the proposed algorithms, three datasets are used; two public datasets, and the developed AHHM dataset. Secondly, standard metrics are used to evaluate the proposed algorithms. Further details are given in the following subsections.

6.1.1 Datasets

Bukhari dataset contains 38 pages from seven Arabic manuscripts which were scanned at a private library in the old city of Jerusalem [90]. These manuscripts are written by several writers, and contain dense side-note text. Moreover, the manuscript images are binarized. The ground truth of Bukhari dataset is provided in [1]. The ground truth has two sets of images that contain either main-content text or side-notes. Therefore, Bukhari dataset was used to evaluate the proposed DLA approach which is described in Chapter 4. Moreover, the proposed DLA method is compared to [1] because both algorithms are learning-free and evaluated using the same dataset.

The second publicly available historical dataset is called HADARA80P [188]. The dataset has been developed by the Institute for Communications Technology, Technische Universitt Braunschweig. It contains 80 scanned pages of Arabic historical handwritten manuscript. The main source book of the dataset is ”بدل الماعون في فضل الطاعون” that can be transliterated as $ba\underline{d}lu \quad \bar{a}lm\bar{a}\bar{u}n \quad f\bar{i} \quad fadlu \quad \bar{a}lt\bar{a}\bar{u}n$, which can be translated to ”About the Advantage of the Pest”. The book was published in Jumada al-khirah, 833 AH (Islamic calender), which corresponds to Feb. 1430 AD. It was written by a single writer except for the last three pages of the book. The pages of the book are colored and contains main-content blocks with few separated side notes on some pages. The words in red color are used to structure the book's content such as chapter names. HADARA80P dataset is used to evaluate the proposed keyword spotting technique that was described in Chapter 5. The HADARA80P dataset provides 25 keywords that are selected to evaluate document retrieval. Table 6.1 shows

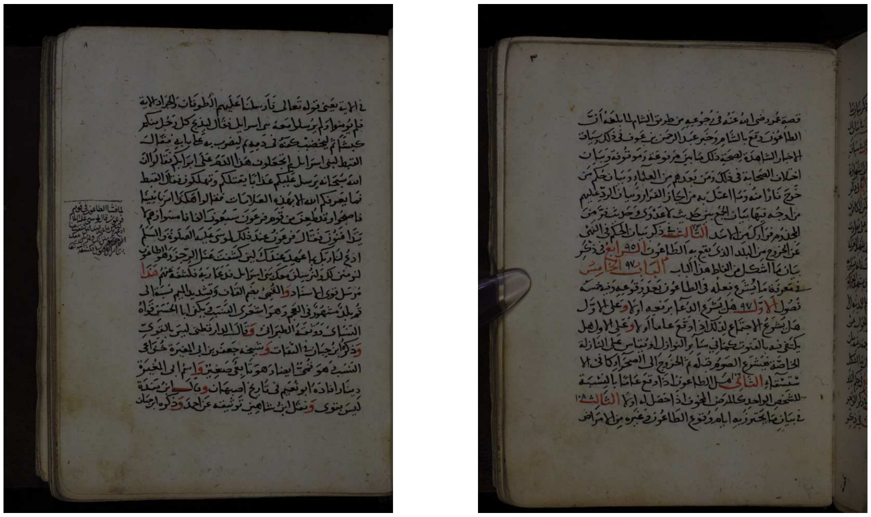
Table 6.1: Samples of HADARA80P dataset's keywords

Code	TRANS	Keyword	NS	Code	TRANS	Keyword	NS
HKW01	Allah		349	HKW02	Mohammad		41
HKW03	Ahmed		46	HKW04	Muslum		25
HKW05	Rasuol		45	HKW06	Osama		25
HKW07	Ziad		47	HKW08	Intaha		24
HKW09	A-la-gah		26	HKW10	Ketab		29
HKW11	Rwaiah		50	HKW12	Akhrajh		64
HKW13	Tarieq		34	HKW14	Radhi		48
HKW15	Hadieth		79	HKW16	Dhekr		58
HKW17	Ba-ian		26	HKW18	Shahada		23
HKW19	Slm		90	HKW20	Ta-ala		45
HKW21	Tauun		5	HKW22	At-Tauun		147
HKW23	Aadaakum		24	HKW24	Al-Jen		65
				HKW25	Israil		22

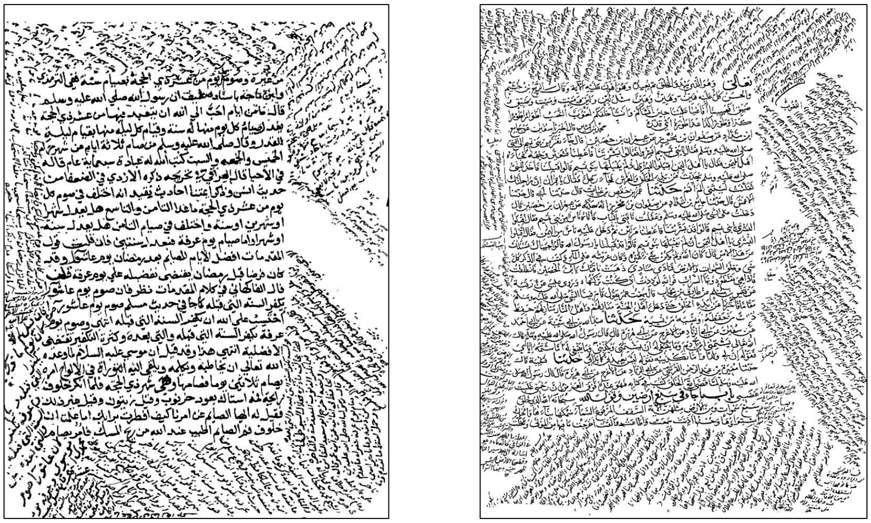
- TRANS: Transliteration, NS: Number of Samples

examples of each keyword.

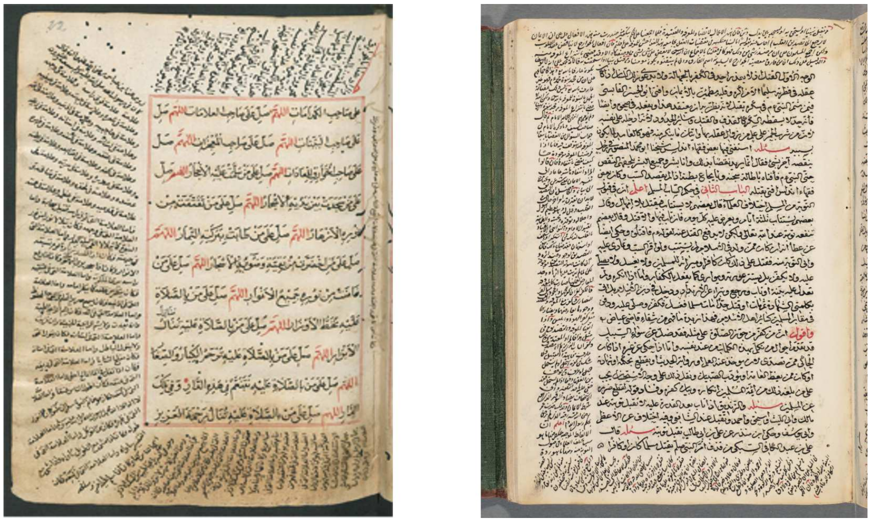
Thirdly, the AHHM dataset contains 108 manuscript pages collected from different sources. Each page has main content and some side-notes. The complete description of the AHHM dataset is presented in Chapter 3. Figure 6.1 shows some examples of manuscript pages of these datasets.



(a) HADARA80P dataset



(b) Bukhari dataset



(c) AHHM dataset

Figure 6.1: Two page examples from each dataset

6.1.2 Evaluation metrics

DLA evaluation

In document layout analysis algorithm performance evaluation, there are two main issues; correspondence, and type of comparison. Since the objective is to segment the main content from handwritten historical manuscript pages, the correspondence becomes simple. That is the main-content region of a manuscript page is used for evaluating the results. Secondly, we adopted evaluation framework 3 that was described in Chapter 2, Section 2.4.

In framework 3, a segmentation result is considered successful if it completely overlaps only one ground truth region. Therefore, errors are characterized based on PRImA framework [195] to five types (see Figure 6.2):

- Merge: A segmented region that overlaps more than one template region
- Split: A template region is overlapped by more than one segmentation region.
- Miss: A template region that does not overlap any segmented region.
- Partial Miss (PMiss): A template region that does not completely overlap a segmented region.
- False detection: A segmented region that overlaps no template region.

Since the page segmentation has binary classes (main-content or side-notes), the errors are relatively computed to both regions. Let D_1 be a segmentation result, and G_1 and G_2 are two template regions. A segmented region D_1 is considered a merge error

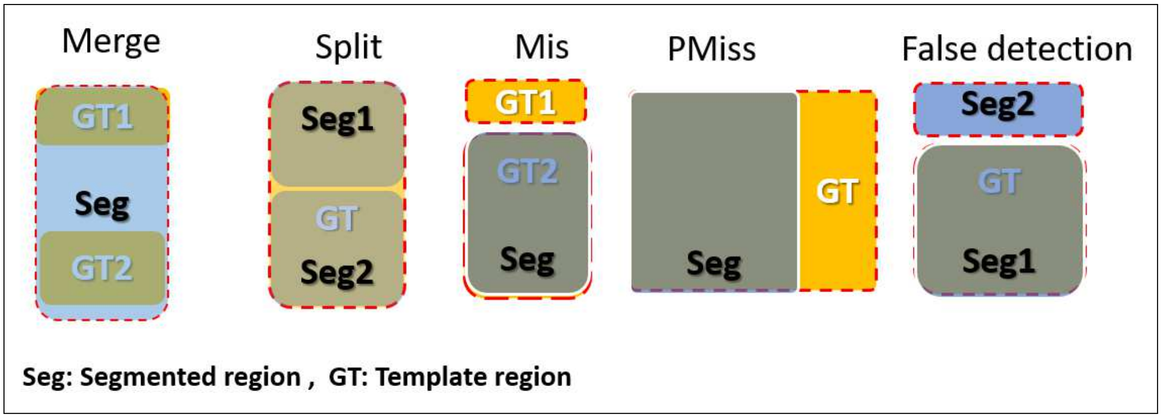


Figure 6.2: PRImA Framework errors

if it overlaps G_1 completely and at least 60% of G_2. A split error occurs when a segmented region D_1 is divided to D_{11} and D_{12} that both correspond to G_1. A miss error is encountered when a segmented region D_1 overlaps at most 60% of G_1. It is a partial-miss (PMiss) if a segmented region D_1 overlaps at most 90% of G_1. Finally, the false detection occurs when D_1 does not overlap any template region.

In order to calculate the success rate of the segmentation SR, each error is first multiplied by the affected foreground pixels of the merged, missed, partially missed, split, or falsely detected regions. Then, these error rates ER are used to calculate the success rate [193]:

$$SR = \frac{\sum_{i=1}^{N} \omega_i}{\sum_{i=1}^{N} \frac{\omega_i}{1-ER_i}} \tag{6.1}$$

where N is the number of error types, and ω_i are the final weights that are calculated of each error type as:

$$\omega_i = \frac{(N-1)ER_i + 1}{N}$$

In order to compare the segmentation results of the proposed algorithm to the method described in [1], we also adopted their evaluation metrics. Therefore, Precision, Recall, and $F_{measure}$ metrics have been used to evaluate the performance of the DLA method. The Precision (P) and Recall (R) are estimated per equations (6.2) and (6.3). True-Positive (T_P) is the rate of main-content PAWs labeled as main text, False-Positive (F_P) is defined as the rate of side-note PAWs labeled as main text, and False-Negative (F_N) is the rate of main-content components classified as side-notes.

$$P = \frac{T_P}{T_P + F_P} \tag{6.2}$$

$$R = \frac{T_P}{T_P + F_N} \tag{6.3}$$

The $F_{measure}$ is a single value that combines both the precision and recall. It shows how precise is the segmentation result to recall correct elements out of all segmented components. The $F_{measure}$ is computed according to the following equation:

$$F_{measure} = 2 \times \left(\frac{P \times R}{P + R} \right) \tag{6.4}$$

Keyword spotting evaluation

The experiments on KWS includes two main phases; keyword training and keyword spotting. The used datasets are divided into two parts; 1) set of keywords, 2) set of all words. In the training mode, the main purpose is to train a set of SVMs to recognize keywords. Therefore, a set of keywords is used in this mode. Moreover, the recognition behavior of these SVMs are analyzed to define spotting thresholds. The performance of the word recognition, in the training phase, is reported using the recognition-rate as follows:

$$RR = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

where T_P is true positive; a word w_i is correctly recognized as keyword KW_i, T_N is true negative; a word w_i is correctly not-matched with keyword KW_j, F_P is a false positive; a word w_j is falsely recognized as keyword KW_i, and F_N is a false negative where a keyword w_i is falsely recognized as word KW_j.

On the second experiments, we adopted Average Precision (P_{avg}) metric which is a standard evaluation metric for information retrieval performance evaluation. For each keyword, Precision metric reflects the percentage of true positives as compared to the total number of retrieved image words by KWS algorithm. Then, P_{avg} is the average value of all precisions. It provides a single value measure of precision for all spotted keyword images and computed as:

$$P_{avg} = \frac{1}{K} \sum_{i=1}^{K} P_i^w$$

where K is the total number of queries, P_i^w is computed precision per query.

Moreover, to have a complete overview of the retrieval results, Average Recall (R_{avg})

can be utilized. The R_{avg} is computed as follows:

$$R_{avg} = \frac{1}{K} \sum_{i=1}^{K} R_i^w \tag{6.5}$$

where R_i^w is computed recall per query. Finally, these two evaluation metrics are combined to indicate the overall spotting system performance using an F-measure as:

$$F_{measure_{avg}} = 2 \times \left(\frac{P_{avg} * R_{avg}}{P_{avg} + R_{avg}} \right)$$

6.2 Document Layout Analysis Results

The proposed algorithm is compared to [1] because of three reasons; 1) both methods are learning-free approaches, 2) Exact datasets are used in the experiments, 3) the analysis code of [1] is publicly available in [236].

First, we reproduced the segmentation results of [1] using their provided Matlab code [236]. Then, we evaluated the results of both methods, using [1] evaluation metrics, and using PRImA framework. Table 6.2 tabulates the performance evaluation using $F_{measure}$ metric (Equation (6.4)), and Table 6.3 shows the performance in terms of segmentation success rate (PRImA framework).

The results in Table 6.2 shows the superiority of the proposed method in terms of $Fmeasure$ using Bukhari dataset. It has high segmentation recall 98.55% at 96.93% segmentation precision. Moreover, the proposed method shows promising results in comparison to other method in terms of quality of main-content segmentation in gen-

Table 6.2: Performance evaluation using [1] metric

Data	Method	P	R	$F_{measure}$
Bukhari	[1]	97.94	84.30	90.11
	Proposed	96.93	98.55	97.70
AHHM	[1]	98.67	95.27	96.78
	Proposed	97.49	97.14	97.08

Table 6.3: Segmentation accuracy based on success rate

Data	Method	$SR(\%)$
Bukhari	[1]	70.41
	Proposed	98.83
AHHM	[1]	92.97
	Proposed	98.23

eral as indicated in Table 6.3 (98.83% and 98.23% SR). In addition, we can notice from Table 6.3 that the computed success rate of [1] on Bukhari dataset is penalized because of its Merge and Miss errors. Also, the same performance degradation is captured by the difference between Precision and Recall of method [1] on Bukhari dataset that is shown in Table 6.2. Figures 6.3 and 6.4 show examples of the segmentation results on both datasets. On the other hand, our method is slower than the other method. It requires 159.5 seconds on the average for the analysis per page, while the other method requires 73 seconds on the average per page. Comparing both techniques, our layout analysis needs more time to perform moving-window analysis while the other method uses fast energy minimization cuts to extract main content [64].

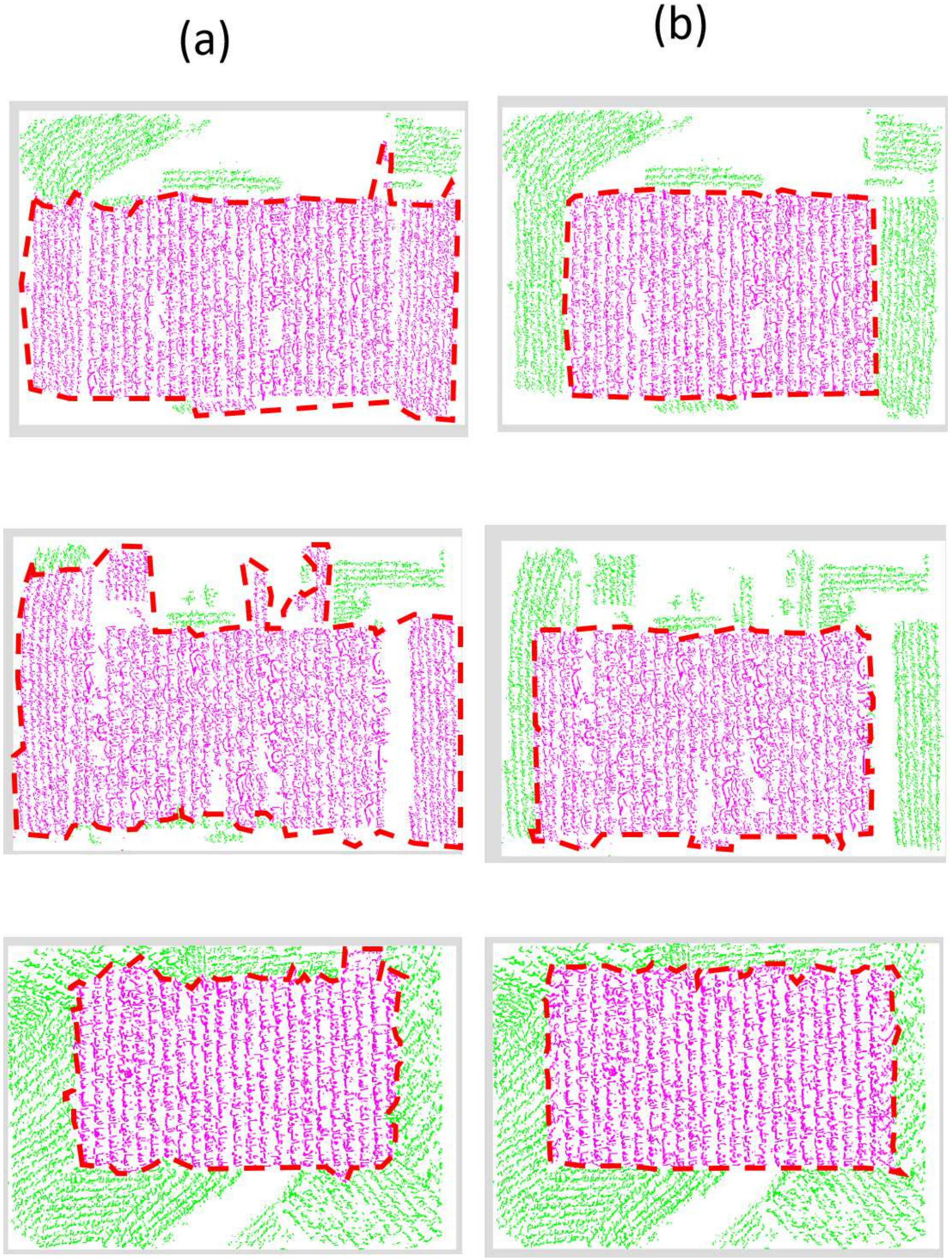


Figure 6.3: Sample results on Bukhari dataset. a) Segmentation results of [1], b) The proposed method results

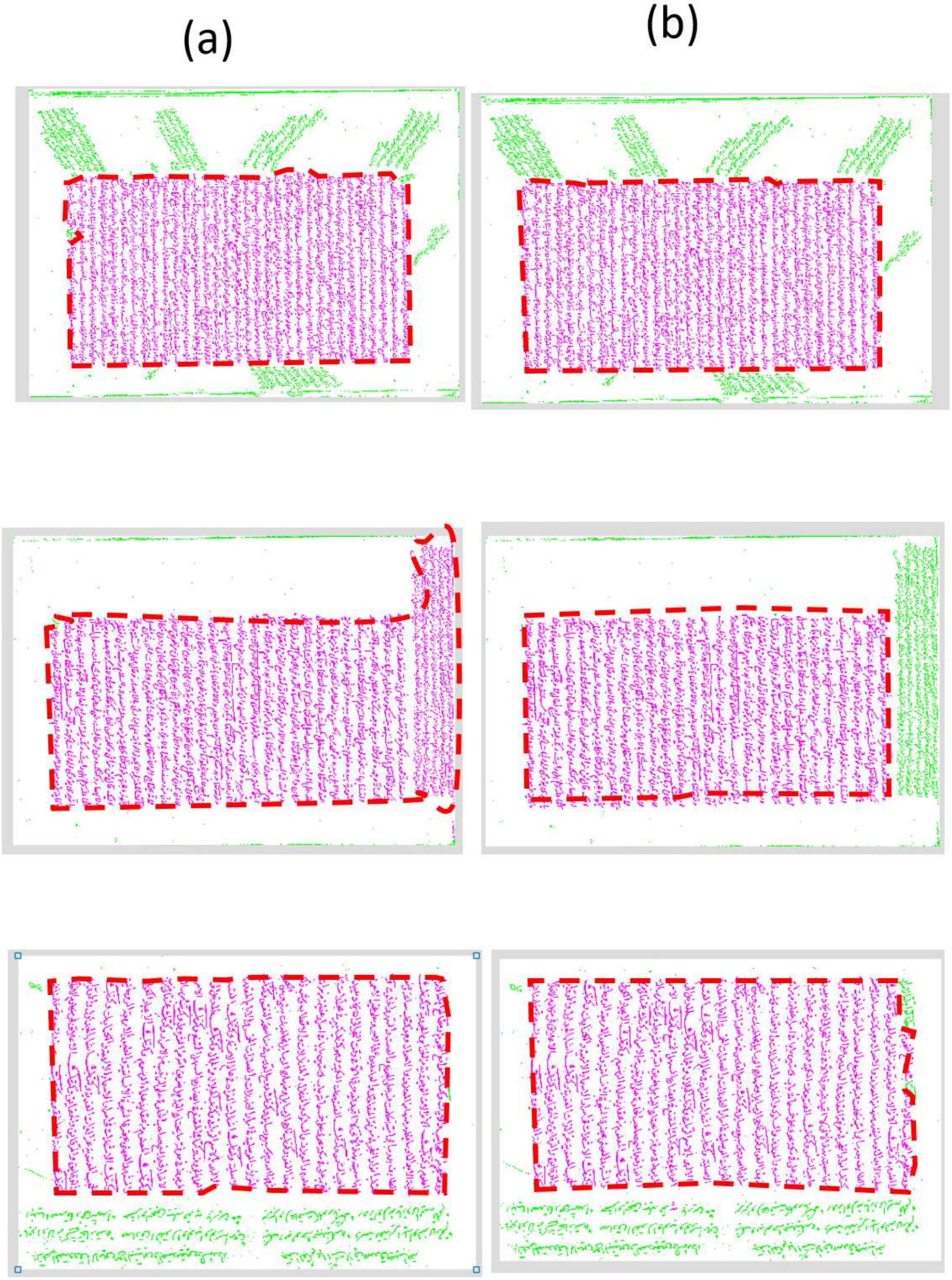


Figure 6.4: Sample results on AHHM dataset. a) Segmentation results of [1], b) The proposed method results

6.2.1 Discussions

Two reasons have affected the performance of [1]. One reason is that Gabor filter may produce similar responses to the main-content and side-notes. Thus, the energy minimization step of [1] does not produce correct segmentation results (see Figure 6.4 middle example). On the other hand, the proposed method addresses the main-content segmentation by analyzing the whitespace and connected components on the boundary regions. It considers the analysis locally to address touching components and unclear whitespaces. Moreover, it shows good analysis for several examples of manuscript layouts, writing styles, fonts, or writers.

Sometimes, it is difficult for the proposed method to detect a separation boundary between main-body and side-notes as discussed in Chapter 4. This issue is reflected in the results shown in Figure 6.3.

In the second experiment, the methods are evaluated on AHHM. The proposed method has slightly degraded performance due to the Recall factor. On contrast, method [1] have enhanced performance in the second experiment due to the response of Gabor filter on different text. Further error analysis discussion is presented in Section 6.4.

6.3 Document Classification Results

To evaluate the proposed keyword spotting method, we used a set of keywords provided by HADARA80P dataset (25 keywords) and AHHM dataset (25 keywords). A sample of each keyword is shown in Tables 6.1, and 3.2 respectively.

To prepare SVM classifiers for spotting, we collected all keyword instances from HADARAH80P and AHHM datasets. The number of keyword instances from HADARA80P dataset are 1432 word images, while 1062 word instances are in AHHM dataset. These datasets of keywords are used in separate experiments to prepare the SVM classifiers for keyword spotting tasks.

Each dataset of keywords is divided into three sets for training, validation and testing. The training and validation sets are used to build SVM models using BoVW of SURF features. These features are extracted from different image locations based on three methods; dense, SURF, and word-skeleton key-point sampling. The best validated SVM models are selected to estimate keyword-spotting thresholds t_1, t_2, and t_3 as discussed in chapter 5. Finally, keyword spotting is carried-out on the complete datasets.

6.3.1 Training Results

Since the training consists of word recognition, we report in this subsection the recognition performance of SVMs on both datasets. Table 6.4 presents the recognition rates of the three methods and their integrated system. The integration of the proposed methods is carried-out using weighted majority voting. The weights are set automatically using the recognition rates of the SVM models on the validation sets. Therefore, each method is assigned a weight ω_j using the following equation:

$$\omega_j = \frac{RR_{v_j}}{\sum_{i=1}^{i<3} RR_{v_i}} \tag{6.6}$$

where RR_v is the recognition-rate on the validation set. To explain how the integration is carried out; suppose we have a keyword set of four words$\{K_1, K_2, K_3, K_4\}$, and three expert systems S_1, S_2, and S_3 with recognition validations 95%, 80%, and 85% respectively. The weights of the expert systems are computed as equation (6.6) which yields weights as $\omega_1 = 0.365$, $\omega_2 = 0.307$ and $\omega_3 = 0.326$ respectively.

Individual decisions of expert systems is given as :

$$\begin{bmatrix} \underline{S_1} & \underline{S_2} & \underline{S_3} \\ K_1 & K_2 & K_2 \\ K_3 & K_1 & K_1 \\ K_4 & K_2 & K_3 \\ K_1 & K_1 & K_3 \end{bmatrix}$$

The integration decisions are computed as:

$$\begin{bmatrix} \underline{K_1} & \underline{K_2} & \underline{K_3} & \underline{K_4} & \textbf{decision} \\ 0.365 & \mathbf{0.633} & 0 & 0 & : K_2 \\ \mathbf{0.633} & 0 & 0.365 & 0 & : K_1 \\ 0 & 0.307 & 0.326 & \mathbf{0.365} & : K_4 \\ \mathbf{0.672} & 0 & 0.326 & 0 & : K_1 \end{bmatrix}$$

Therefore, we hope by the integration to capture more correct word recognition.

The results in Table 6.4 indicate that the skeleton-based keypoint sampling results

Table 6.4: Recognition results using HADARAH80P dataset

Codebook size	Average Recognition Rate (%)			
	Skeleton	SURF	Dense	Integration
128	**87.60**	84.03	61.36	87.33
256	**90.19**	85.13	80.16	89.50
512	**92.37**	90.44	85.07	90.31
1024	**94.14**	93.73	88.79	90.99
2048	**95.51**	94.28	90.48	91.95

have better recognition rates in comparison to the other methods. It is also observed that by increasing the size of the codebook the recognition rate becomes better. This behavior seems to be a consequence of strengthening BoVW representation by having better resolution and more features of each keyword. This extension of BoVW codebook supports the SVM classifiers to distinguish between keywords and their substrings or similar words. On the other hand, SURF-based method runs faster than the other methods. It performs the recognition of a keyword in (≈ 4) seconds on average, while (≈ 20) seconds are needed for the skeleton-based method. This is because the skeleton-based method requires additional preprocessing steps. Dense-based method suffers from long computation time with (≈ 200) seconds per keyword on the average. Hence, the integration performance of the three methods is affected by this drawback. Figure 6.5 shows the recognition time in relation to the codebook sizes. The SURF and word-skeleton have small increase in running-time by increasing the codebook sizes, while dense-based SURF recognition time increases linearly by increasing the codebook size. The integration performance could be affected by the performance of dense-based method. This is reflected in the reported performance of the integration approach in Table 6.4. Comparing, the results of the integration approach using

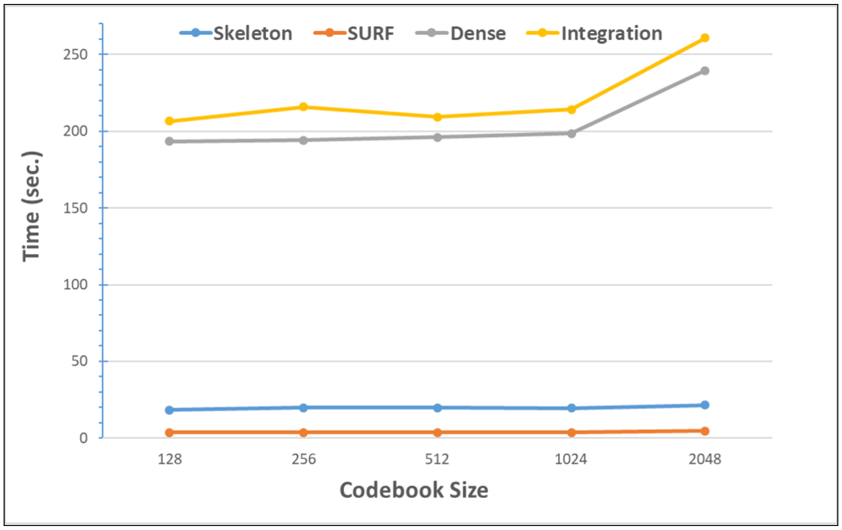


Figure 6.5: Average recognition time of each method

different codebooks in Table 6.4, we notice that by increasing the codebook size, the performance of the integration increases slowly. It could be due to the similar recognition performance of the three systems. Therefore, their classification decisions may be the same.

The results of the methods on AHHM dataset are tabulated in Table 6.5. The AHHM dataset has been written by several writers using different font types. This causes variations on the keyword instances that impacted the results. Sample instances of two keywords are shown in Figure 6.6.

By studying the results in Table 6.5, the BoVW codebook size has a positive impact

Table 6.5: Recognition results using AHHM dataset

Codebook size	Average Recognition Rate (%)			
	Skeleton	SURF	Dense	Integration
128	38.96	25.79	8.39	**41.55**
256	54.85	46.41	32.11	**60.98**
512	61.37	57.12	55.58	**69.30**
1024	76.19	66.17	75.07	**82.03**
2048	81.44	69.83	78.38	**84.75**

on the performance of the methods. By increasing the size of the BoVW codebook, better results have been achieved. Starting by 128 codebook-size, the methods were not performing comparable to using 2048 codebook size. This could be due to low resolution of extracted keywords from AHHM dataset. So, 128 visual words were not enough to capture differences between keyword image. This issue is evident in the case of dense sampling. It achieved 8.39% recognition rate at 128 codebook size, but 78.38 % recognition rate at 2048 codebook size. The Skeleton-based method outperformed SURF and dense sampling. It could be attributed to the localized features on the handwriting word-skeleton. This means the word-skeleton is less affected by the low image-resolution in comparison to other methods.

The performance of all methods have improved by increasing the codebook size. It is observed in Table (6.5) that the performance is linearly enhanced. However, at 1024 codebook size, the enhancement has started to slow down. To reflect on this observation from both experiments, the integration may produce higher performance results than individual methods if the size of a codebook was not enough to capture discriminant features. Otherwise, the integration performance may have some minor improvements.

6.3.2 Spotting Results

Based on the performance of each SVM in the training phase, three best performing SVM models are selected to carry-out the spotting task on the complete datasets. In spotting, the case of sub-string and similar word instance matching becomes a chal-

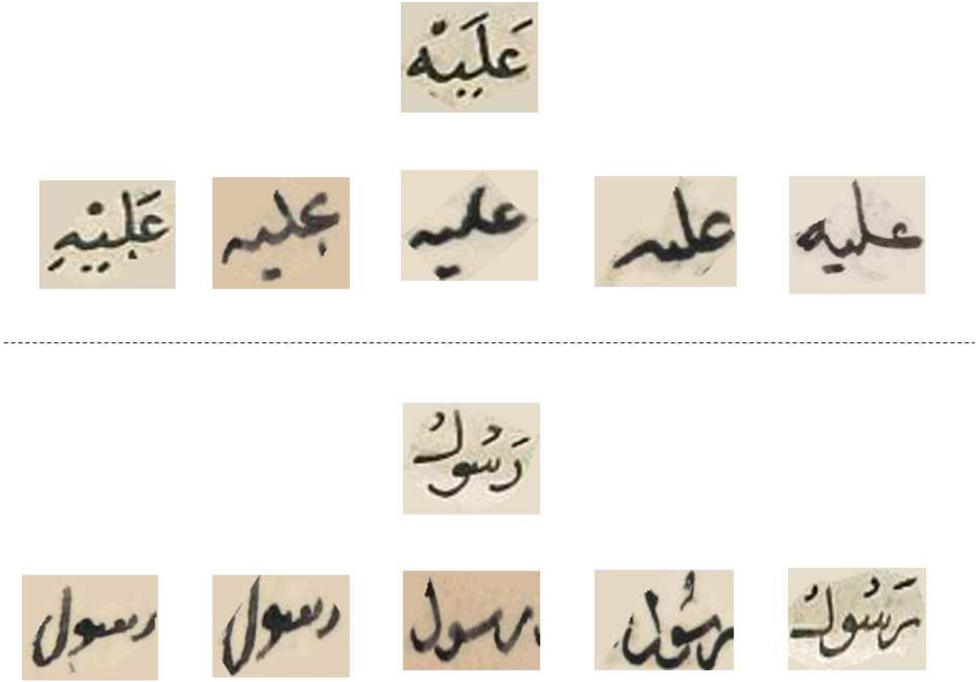


Figure 6.6: Two keyword examples and their corresponding word instances

lenge. According to the handwriting in HADARA80P dataset, completely different words may become similar if their diacritics are removed as shown in Figure 6.7. In that example, Keyword KW14 which can be translated as ”To satisfy” is compared to another word that can be translated as ”To advise”.

By analyzing the SVMs response distributions, we found that t_1 is approximately ranged in $[-0.037, -0.025]$, and t_2 is in the range of $[-0.048, -0.036]$, and $t_3 \approx -0.021, \pm 0.002$. These parameters have different impacts on the proposed KWS. Tables 6.6 and 6.7 show the performance results of keyword spotting on HADARA80P, and AHHM datasets respectively.

Table 6.6 is divided into three parts corresponding to the used threshold. Threshold t_2 has a balanced impact on the skeleton-based and SURF systems as indicated by their P_{avg} and R_{avg} results. Their selected interest points are mostly located on the writing in comparison to dense-based sampling that have interest points located off-writing. This performance behavior is noticed in the training phase as well (see Tables 6.4 and 6.5). The negative impact of dense-based method in the performance of the integration is indicated in Table 6.6 using t_2 and t_3 thresholds.

As indicated in Table 6.6 (First part), the performance of all KWS using the computed threshold t_1 has low precision because threshold t_1 allows more false positives due to sub-string and similar word instances. This issue is clearly noted by studying the systems' Recall results. In other words, these systems have accepted most of the similar word instances compared to a given keyword query. Consequently, the number of retrieved true instances per keyword query is nearly perfect with large false retrievals. In general, the $F_{measure_{avg}}$ indicates that skeleton-based method performed better spotting than other methods.

The integrated KWS in spotting using threshold t_1 performed better than skeleton method. By investigating the performance behavior of the integrated method in train-

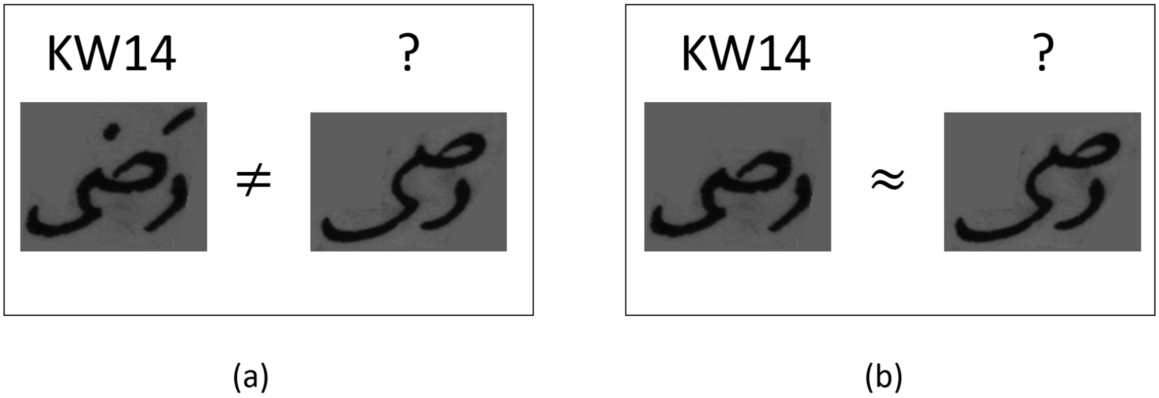


Figure 6.7: Confusion due to similar shape structures; a) Comparison between keyword HKW14 (Radhi) and another word "To advise", b) Comparison after removing written diacritics from keyword HKW14 (Radhi)

Table 6.6: Keyword spotting performance on HADARA80P dataset

Methods	Spotting Using t_1		
	P_{avg}	R_{avg}	$F_{measure_{avg}}$
Dense	25.73	83.43	39.33
SURF	25.49	95.00	40.20
Skeleton	40.98	97.79	57.75
Integration	44.82	96.27	**61.17**
	Spotting Using t_2		
	P_{avg}	R_{avg}	$F_{measure_{avg}}$
Dense	34.19	80.57	48.01
SURF	83.77	78.36	80.98
Skeleton	82.19	85.09	**83.61**
Integration	97.24	70.58	81.79
	Spotting Using t_3		
	P_{avg}	R_{avg}	$F_{measure_{avg}}$
Dense	90.17	52.99	66.75
SURF	95.54	58.74	72.75
Skeleton	93.65	64.73	**76.55**
Integration	99.69	50.93	67.42

ing and spotting phases, we can conclude that the integration tend to perform better when individual methods have low performance. In other words, the majority voting integration may suite combing weak classifiers.

Finally, an aggressive spotting using t_3 is performed in the last experiment. The system in this case rejects mostly all false similar instances of a keyword query. Therefore, the false positives are low in the performance of these systems. However, these systems have increased their rejection behavior at the same rate. This issue affects the Recall measure and results in lower $F_{measure_{avg}}$ value in comparison to t_2.

Table 6.7: Keyword spotting performance on AHHM dataset

Methods	Spotting Using t_1		
	P_{avg}	R_{avg}	$F_{measure_{avg}}$
Dense	63.3	96.04	76.31
SURF	63.21	93.61	75.46
Skeleton	55.04	96.93	70.21
Integration	75.45	94.24	**83.80**
	Spotting Using t_2		
	P_{avg}	R_{avg}	$F_{measure_{avg}}$
Dense	96.95	86.03	91.11
SURF	97.79	84.86	90.87
Skeleton	92.21	90.60	91.40
Integration	97.87	85.83	**91.46**
	Spotting Using t_3		
	P_{avg}	R_{avg}	$F_{measure_{avg}}$
Dense	80.32	61.71	70.87
SURF	84.00	56.16	67.31
Skeleton	99.74	73.64	**84.73**
Integration	86.30	59.16	70.20

6.4 Error Analysis

6.4.1 Document Layout analysis

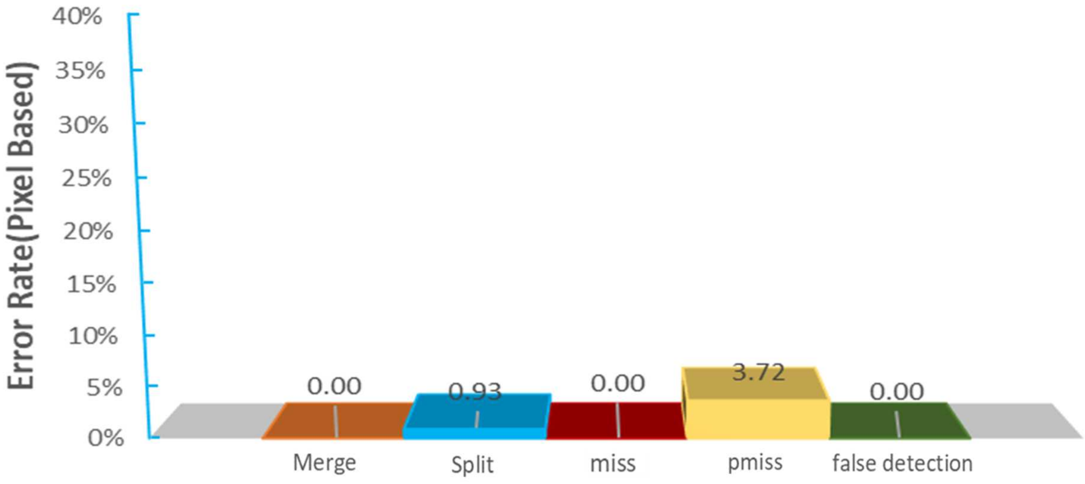
The error rates breakdown, in Figures (6.8, and 6.9), show that most of the segmentation errors of the proposed method, on both datasets, are due to Partial-Miss error type. Figures 6.8 and 6.9.(a) show all segmentation error rates of the proposed method on both datasets. On the other hand, method [1] suffers from high Merge error on both datasets. This error reflects the weakness of Gabor filter in differentiating main-content elements from side-notes elements. Figures 6.8 and 6.9.(b) illustrate the error-rates breakdown for method [1] on Bukhari and AHHM respectively.

Although the experiments show that the proposed algorithm performs well to extract main content from historical Arabic manuscripts, a number of limitations have been identified. This method computes initial main-content characteristics that could be falsely extracted from the transition zones which could be mixed with side-notes components. In other words, a manuscript with dense text on both main-content and side-notes may lead to incorrect characterization of main-region. Secondly, the proposed method uses K-means clustering to define a local separation path between main-content and side-note elements of the transition regions which is the main reason for segmentation Partial-Misses errors. To overcome this issue, one solution could be including context information while clustering to avoid blind separation.

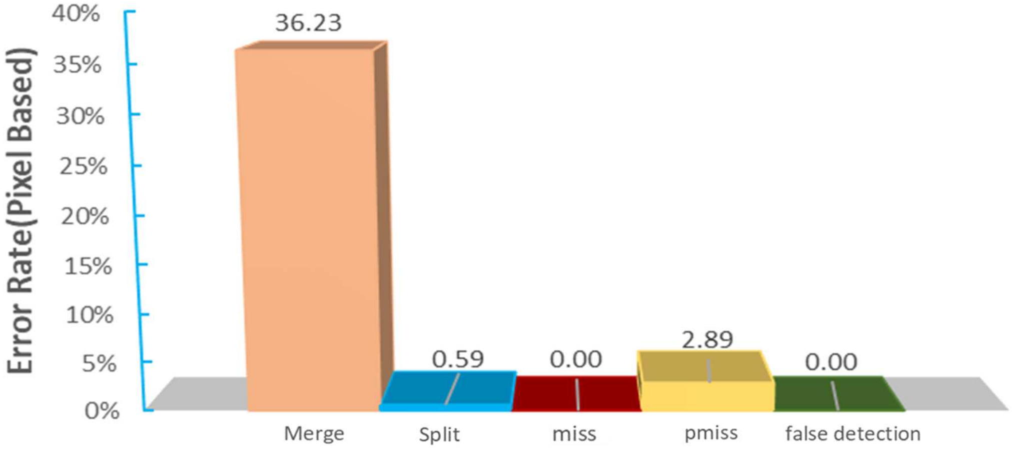
6.4.2 Document Classification

In the proposed KWS system, the objective thresholding is heavily depending on the trained SVM classifier behavior. Since the data is highly unbalanced; the number of word instances per keyword is totally different from one class to another. For example, in HADARA80P dataset, keyword HKW01 has 349 instances while HKW25 has only 22 instances which makes the classifier training difficult to generalize. Therefore, k-fold cross validation were used to ensure stability and effectiveness of the SVM training. The trained SVMs are used to compute the spotting threshold. In this work, we tested three thresholds t_1, t_2, and t_3, which may yield three different spotting configurations.

For threshold t_1, the errors are most likely drawn from false positives because the behavior of t_1 in spotting tends to accept the matching of similar words. Figure 6.10 shows some examples of correct and false matches of HKW01 and HKW09. The

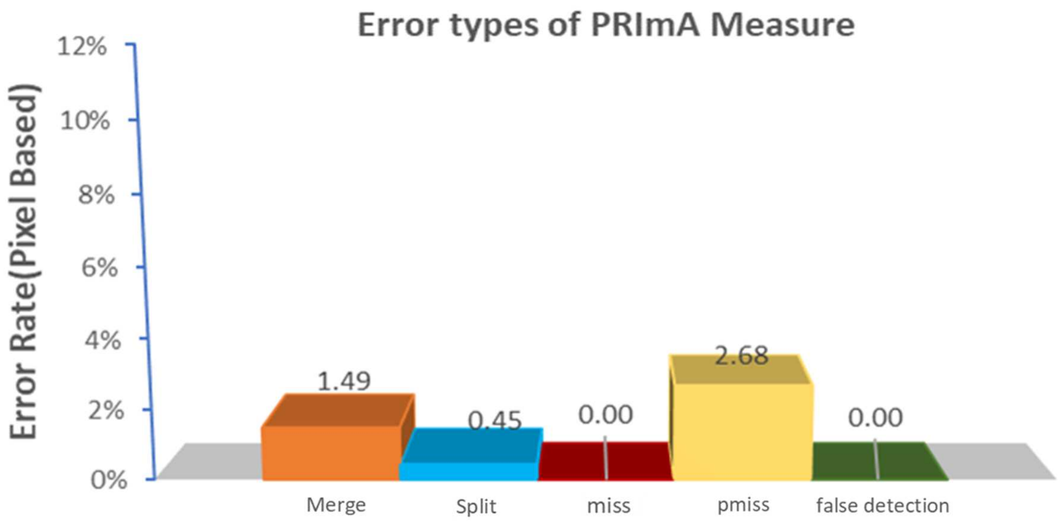


(a) Error rates of the proposed method

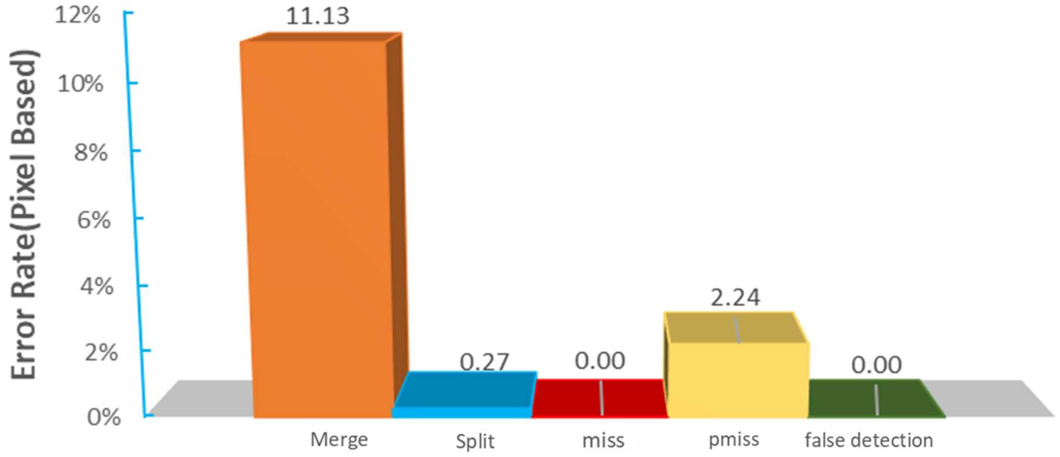


(b) Error rates of [1] method

Figure 6.8: Analysis of the error rates breakdown using Bukhari dataset

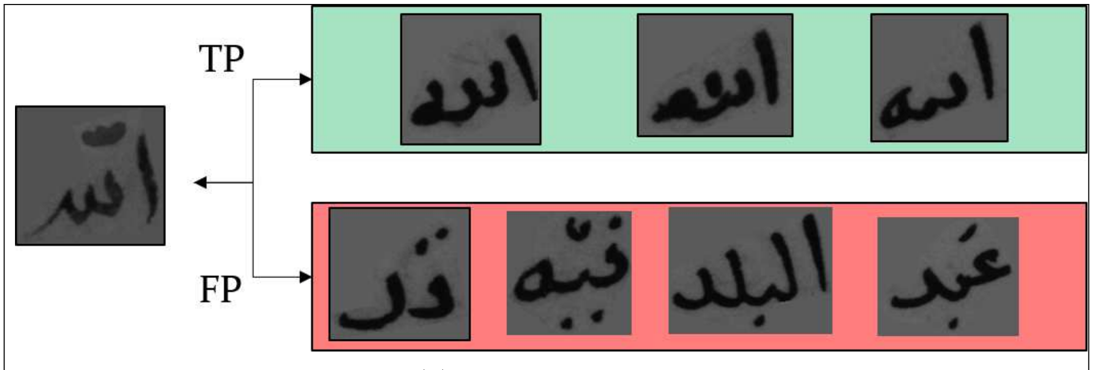


(a) Error rates of the proposed method

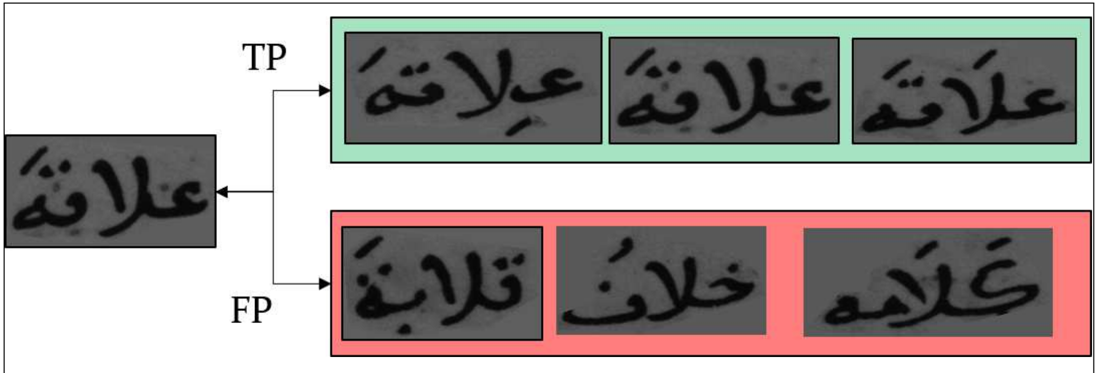


(b) Error rates of [1] method

Figure 6.9: Analysis of the error rates breakdown using AHHM dataset



(a) Keyword HKW01



(b) keyword HKW09

Figure 6.10: Examples of retrieved word errors

limitation of the skeleton-based keypoint sampling is ignoring keywords' diacritics. So, by extracting features from the main strokes only, the skeleton-based KWS method loses part of its accuracy. This issue can be observed from the sample results shown in Figure 6.10. Removing diacritics of some words make them look like another keyword query (e.g. HKW09). The same errors are observed from the performance of KWS using thresholds t_3, but with increased number of false negatives per keyword query. This indicates that t_2 is a proper objective threshold which balances the rejection and matching behavior of KWS.

6.5 Conclusions

In this chapter, the performance evaluation of document layout analysis and classification algorithms is discussed. For document layout analysis, the performance is evaluated on two datasets (viz. Bukhari and AHHM). The first dataset is used to compare the performance of the proposed algorithm against a state-of-the-art method that have used the same dataset. The second experiment is conducted on our developed AHHM dataset to benchmark its analysis results.

The analysis results shows that our method outperformed the other algorithm in both experiments. By investigating the results, we observed that our method has two levels of local analysis, while the other method has only one level of local analysis.

Based on the success rate metric, the quality of the segmentation results can be computed by considering the segmentation errors. The proposed DLA method has achieved 98.83%, and 98.23% success rate using both datasets respectively. The segmentation errors are mostly caused by Partial-Miss errors. This issue is due to the limitation in correcting the final stop windows using K-means clustering.

Secondly, a learning-based keyword spotting algorithm is proposed to address information retrieval. The proposed algorithm has two phases; training, and spotting. In training, the algorithm estimates the spotting thresholds based on the behavior of SVM in training phase. Three thresholds are estimated t_1, t_2 and t_3. Each of these thresholds can configure the spotting system at different operation level. Threshold t_1 can be used to spot all keywords with their similar words. In other words, it has wide acceptance range. On the other hand, threshold t_3 tends to reject matches that have weak results. Threshold t_2 has a balance behavior among the three operating thresholds.

Finally, comparing the performance of the KWS using different keypoint sampling methods indicates that the skeleton-based keypoint sampling is suitable for handwriting. The features extracted from word-skeleton keypoints have good impact on the KWS system performance. Furthermore, increasing the BoVW codebook size has enhanced the overall KWS performance. However, this improvement is not linear, it slows at some points.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, we summarize our contributions and highlight the limitations to pinpoint some directions for future research.

7.1 Concluding Remarks

Due to the lack of benchmark Arabic database and challenging issues in Arabic historical manuscripts, few studies have been proposed to address Arabic historical manuscript layout analysis and classification. In comparison to other languages, the research on Arabic documents forms 16% per our reviewed research population and most of them are conducted on contemporary documents. Consequently, large number of research on word spotting is dedicated to non-Arabic document classification.

In this thesis, we have conducted research on document layout analysis and classification of Arabic historical manuscripts. Motivated by the strengths of hybrid analysis techniques for complex layout analysis, we developed a novel DLA approach that boosts the whitespace separation between main-content and side-notes regions, and eases its final segmentation. Furthermore, we developed a novel learning-based keyword spotting approach that models the behavior of the learning classifier and improves the performance of word spotting. In general, our contribution includes developing successful manuscript layout analysis and classification system for Arabic historical manuscripts. The following are the contributions of this thesis:

- **Comprehensive Literature Survey**

  When we reviewed the literature, we did not find any survey since the last decade. In 2017 and after we wrote our survey paper two new literature surveys are published [232, 11]. Our comprehensive literature survey is presented in Chapter 2. It addresses the preprocessing, analysis strategies, databases and evaluation metrics, with emphasizes on Arabic manuscripts.

- **AHHM database**

  The second contribution is the development of an Arabic historical handwritten manuscript database that is presented in Chapter 3. AHHM database is designed to support segmentation and segmentation-free document retrieval. The database consists of 108 historical manuscript pages. Using this database, we segmented 2135 words from which we selected 25 keywords. The database was used in our research and experiments, and will be made freely available to researchers.

- **Document layout Analysis**

  The third contribution is achieved by designing and developing a novel learning-free hybrid document layout analysis approach. The approach is outlined in Chapter 4. The proposed approach has three novel outcomes; **1)** A fast localization of the main-content region of historical manuscripts by using an anisotropic diffusion filtering (ADF) that allows automatic document characterization; **2)** developing whitespace analysis for handwritten documents. To-our-knowledge, whitespace analysis is used in printed document analysis only; **3)** A hybrid technique that integrates global and local analysis to identify the main-content boundary. Experiments of the proposed approach using Bukhari and AHHM databases have yielded promising results achieving precision rate of 96.93% at 98.55% recall, and precision rate of 97.49% at recall 97.14% respectively. Furthermore, the performance is analyzed using PRImA evaluation metric that reveals the approach's segmentation quality. The proposed method have achieved 98.83% and 98.23% PRImA success rate using Bukhari and AHHM databases respectively.

- **Document Classification**

  The fourth contribution is the design and implementation of a learning-based keyword spotting system (KWS) for Arabic historical manuscripts. The method is outlined in Chapter 5. In this contribution, we suggested three novel outcomes; **1)** Skeleton-based interest region sampling. Unlike automatic detectors that may select keypoints off-writing regions of a word-image. By using word-skeleton method, we guaranteed feature extraction from writing regions of a word-image. **2)** The proposed KWS estimates an operational threshold objectively by modeling the classifier's matching behavior. **3)** A configurable keyword spotting system. By analyzing and modeling the behavior of the SVM classifiers in the training mode, the spotting task can be configured into three types, soft, balanced and aggressive. The experiments of the proposed KWS approach was applied using two databases; HADARA80P and AHHM. HADARA80P database includes 80 manuscript pages with 16945 segmented words whereas AHHM database includes 108 manuscript pages and 2135 segmented words. The proposed approach yields good results using both datasets. Our proposed KWS system using threshold t_2, which is computed between the rejection and matching SVM-responses distributions, was able to achieve 83.61% $F_{measure}$ on HADARA80P database, and 91.40% $F_{measure}$ on AHHM.

7.2 Future Research Directions

Even though the research has several contributions and achieved its aim, we identified some limitations. First, the proposed document layout analysis have adopted K-NN clustering to find the separation boundary between main-content components and side-notes. We found that K-means clustering may lack context information. Therefore, the segmentation in the main-content against side-notes at the transition condition was not optimum.

Secondly, the ADF main-content estimation may fail to find the main content region due to the impact of scale estimation of the filter when manuscript pages have low resolution. In this case, the approach estimates the characteristics using integration method of SWL and ADF that may allow extracting features from side-note regions. Third, skeleton-based keypoint selection has been designed to extract features from the main component of a given word. This design may have issues in matching different words that have similar structures as dissected in Chapter 6.

A number of improvements that researchers of Arabic historical manuscript analysis and classification need to address. It would be interesting to study the performance of the proposed analysis technique on colored manuscripts. Consequently, a set of texture feature may be required to characterize manuscripts main content region instead of the geometric features used in this research. Moreover, by applying ADF filtering on manuscripts, whitespaces are boosted and hence simplifies regional-based analysis. Thus, finer level analysis can be conducted to extract text lines or smaller regions using ADF filtering. Furthermore, the proposed method showed segmentation errors due to Partial-Miss that indicates that local window separation using K-means clustering was not successful for some cases. Therefore, an improvement is required to address this limitation in the future. Usually, Arabic historical manuscript layouts are complex and include several issues such as text-touching, variation in writing-style and fonts, page degradation etc. that are required to be addressed. One possible generic solution may be using deep learning to differentiate between the document elements at the pixel level.

In the proposed keyword spotting, the integration of the three methods was not successful in some cases. This could be due to high variations in the integrated systems. In other words, the three methods integration may not be the optimum solution to improve the performance of the spotting system. Therefore, it would be interesting to investigate the performance using other combinations of these three KWS approaches. Finally, word-skeleton keypoint selection may be improved by including spatial features.

REFERENCES

[1] A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A Coarse-to-Fine Approach for Layout Analysis of Ancient Manuscripts," *14th International Conference on Frontiers in Handwriting Recognition, Greece*, pp. 140–145, 2014.

[2] M. Younki, C. Sung-Bae, and L. Yillbyung, "A data reduction method for efficient document skew estimation based on hough transformation," *in the 13th International Conference on Pattern Recognition, Vienna*, pp. 732–736, 1996.

[3] S. Lowther, V. Chandran, and S. Sridharan, "An Accurate Method for Skew Determination in Document Images," *in Digital Image Computing Techniques and Applications*, vol. 1, pp. 25–29, 2002.

[4] M. Diem, F. Kleber, and R. Sablatnig, "Skew Estimation of Sparsely Inscribed Document Fragments," in *10th International Workshop on Document Analysis Systems, Australia*, pp. 292–296, 2012.

[5] N. Journet, V. Eglin, J. Ramel, and R. Mullot, "Text/Graphic Labelling of Ancient Printed Documents," in *the 8th International Conference on Document*

Analysis and Recognition, Seoul, pp. 1010–1014 , 2005.

[6] R. Saabni, A. Asi, and J. El-Sana, “Text line extraction for historical document images,” *Pattern Recognition Letters*, vol. 35, no. 1, pp. 23–33, 2014.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[8] S. M. Harding, W. B. Croft, and C. Weir, “Probabilistic retrieval of ocr degraded text using n-grams,” in *International Conference on Theory and Practice of Digital Libraries, Berlin, Heidelberg.* Springer, pp. 345–359, 1997.

[9] Y. Leydier, F. Lebourgeois, and H. Emptoz, “Text search for medieval manuscript images,” *Pattern Recognition*, vol. 40, no. 12, pp. 3552–3567, 2007.

[10] D. Doermann, “The indexing and retrieval of document images: A survey,” *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 287–298, 1998.

[11] A. P. Giotis, G. Sfikas, B. Gatos, and C. Nikou, “A survey of document image word spotting techniques,” *Pattern Recognition*, vol. 68, pp. 310–332, 2017.

[12] Harvard Library. Islamic heritage project. [Online]. Available: http://ocp.hul.harvard.edu/ihp/scope.html#se

[13] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, “Syntactic segmentation and labeling of digitized pages from technical journals,” *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, vol. 15, no. 7, pp. 737–747, 1993.

[14] A. Antonacopoulos and R. Ritchings, "Representation and classification of complex-shaped printed regions using white tiles," in *the 3rd International Conference on Document Analysis and Recognition,Montreal, Que., Canada*, vol. 2. IEEE, pp. 1132–1135, 1995.

[15] A. Simon, J.-C. Pret, and A. Johnson, "A fast algorithm for bottom-up document layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 273–277, 1997.

[16] J. Y. Ramel, S. Leriche, M. L. Demonet, and S. Busson, "User-driven page layout analysis of historical printed books," *International Journal of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 243–261, 2007.

[17] T. A. Tran, I.-S. Na, and S.-H. Kim, "Hybrid page segmentation using multilevel homogeneity structure," in *the 9th International Conference on Ubiquitous Information Management and Communication*. ACM Press, pp. 1–6, 2015.

[18] J. Van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel, "Distance measures for layout-based document image retrieval," in *in the 2nd International Conference on Document Image Analysis for Libraries, .* IEEE, pp. 11–pp, 2006.

[19] S.-S. Kuo and O. E. Agazzi, “Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 842–848, 1994.

[20] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, “Word spotting and recognition with embedded attributes,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[21] A. Fischer, A. Keller, V. Frinken, and H. Bunke, “Lexicon-free handwritten word spotting using character hmms,” *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934–942, 2012.

[22] T. Konidaris, A. L. Kesidis, and B. Gatos, “A segmentation-free word spotting method for historical printed documents,” *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 963–976, 2016.

[23] G. A. Fink, L. Rothacker, and R. Grzeszick, “Grouping historical postcards using query-by-example word spotting,” in *the 14th International Conference on Frontiers in Handwriting Recognition,Heraklion, Greece.* IEEE, pp. 470–475, 2014.

[24] R. Pintus, Y. Yang, E. Gobbetti, and H. Rushmeier, “An automatic word-spotting framework for medieval manuscripts,” in *Digital Heritage, Granada, Spain,* IEEE, vol. 2, pp. 5–12, 2015.

[25] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos, “Segmentation of historical machine-printed documents using adaptive run

length smoothing and skeleton segmentation paths," *Image and Vision Computing*, vol. 28, no. 4, pp. 590–604, 2010.

[26] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents : a survey," *International Journal of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 123–138, 2007.

[27] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[28] W. Niblack, *An Introduction to Digital Image Processing.* Prentice-Hall, Englewood Cliffs NJ, 1986.

[29] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[30] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.

[31] J. Bernsen, "Dynamic thresholding of gray level images," in *International Conference on Pattern Recognition*, pp. 1251 –1255, 1986.

[32] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[33] M. a. Ramírez-Ortegón, L. L. Ramírez-Ramírez, I. B. Messaoud, V. Märgner, E. Cuevas, and R. Rojas, "A model for the gray-intensity distribution of histor-

ical handwritten documents and its application for binarization," *International Journal on Document Analysis and Recognition*, vol. 17, no. 2, pp. 139–160, 2014.

[34] K. M. Amin, M. Abd Elfattah, A. E. Hassanien, and G. Schaefer, "A binarization algorithm for historical Arabic manuscript images using a neutrosophic approach," in *the 9th International Conference on Computer Engineering & Systems, Cairo, Egypt.* IEEE, pp. 266–270, 2014.

[35] M.-L. Feng and Y.-P. Tan, "Contrast Adaptive Binarization of Low Quality Document Images," *IEICE Electronics Express*, vol. 1, no. 16, pp. 501–506, 2004.

[36] B. Gatos, P. Ioannis, and S. J. Perantonis., "An Adaptive Binarization Technique for Low Quality Historical Documents," in *Document Analysis Systems VI*, Springer Berlin Heidelberg, pp. 102–113, 2004.

[37] B. M. Singh, R. Sharma, D. Ghosh, and A. Mittal, "Adaptive binarization of severely degraded and non-uniformly illuminated documents," *International Journal on Document Analysis and Recognition*, vol. 17, no. 4, pp. 393–412, 2014.

[38] S. Bolan, L. Shijian, and T. Chew Lim, "Binarization of historical document images using the local maximum and minimum," in *the 8th International Workshop on Document Analysis Systems.* ACM Press, pp. 159–166, 2010.

[39] N. Chaki, S. H. Shaikh, and K. Saeed, "A comprehensive survey on image binarization techniques." Springer India, pp. 5–15, 2014.

[40] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "ICFHR2014 Competition on Handwritten Document Image Binarization," in *International Conference on Frontiers in Handwriting Recognition,Heraklion, Greece.* IEEE, pp. 809–813, 2014.

[41] M. Shafii and M. Sid-Ahmed, "Skew Detection and Correction Based on an Axes-parallel Bounding Box," *International Journal on Document Analysis and Recognition*, vol. 18, no. 1, pp. 59–71, 2015.

[42] B. T. Ávila and R. D. Lins, "A fast orientation and skew detection algorithm for monochromatic document images," in *ACM symposium on Document engineering.* New York, USA: ACM Press, pp. 118, 2005.

[43] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to line segmentation in handwritten documents," X. Lin and B. A. Yanikoglu, Eds. International Society for Optics and Photonics, pp. 65000T, 2007.

[44] I. Bar-Yosef, N. Hagbi, K. Kedem, and I. Dinstein, "Line segmentation for degraded handwritten historical documents," in *the 10th International Conference on Document Analysis and Recognition,Barcelona, Spain.* IEEE, pp. 1161–1165, 2009.

[45] A. Bagdanov and J. Kanai, “Projection profile based skew estimation algorithm for jbig compressed images,” in *the 4th International Conference on Document Analysis and Recognition,Ulm, Germany,* IEEE, vol. 1, pp. 401–405, 1997.

[46] W. Postl, “Detection of linear oblique structures and skew scan in digitized documents,” in *the 8th International Conference on Pattern Recognition*, pp. 687–689, 1986.

[47] H. S. Baird, “The Skew Angle of Printed Documents,” in *the 40th annual Conference and Symposium on Hybrid Imaging Systems*, Rochester, NY, pp. 21–24, 1987.

[48] Y. Nakano, Y. Shima, H. Fujisawa, J. Higashino, and M. Fujinawa, “An Algorithm for the Skew Normalization of Document Image,” in *the 10th International Conference on Pattern Recognition, Atlantic City, USA,* IEEE, vol. 2, pp. 8–13, 1990.

[49] D. Bloomberg and G. Kopec, “Method and apparatus for identification and correction of document skew, U.S. patent no. 5,187,753.” 1989.

[50] M. Sarfraz, S. A. Mahmoud, and Z. Rasheed, “On skew estimation and correction of text,” in *Computer Graphics, Imaging and Visualisation* . IEEE, pp. 308–313, 2007.

[51] S. Li, Q. Shen, and J. Sun, “Skew detection using wavelet decomposition and projection profile analysis,” *Pattern Recognition Letters*, vol. 28, no. 5, pp. 555–562, 2007.

[52] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, “Arabic hand-written text-line extraction,” in *the 6th International Conference on Document Analysis and Recognition, Seattle, WA, USA.* IEEE, pp. 281–285, 2001.

[53] P. V. Hough, “Method and Means for Recognizing Complex Patterns“ U.S. patent no. 3,069,654. 1962

[54] N. Nandini, K. Srikanta Murthy, and G. Hemantha Kumar, “Estimation of skew angle in binary document images using hough transform,” *World Academy of Science, Engineering and Technology*, vol. 18, pp. 44–49, 2008.

[55] B. Yu and A. K. Jain, “A robust and fast skew detection algorithm for generic documents,” *Pattern Recognition*, vol. 29, no. 10, pp. 1599–1629, 1996.

[56] U. Pal and B. Chaudhuri, “An improved document skew angle estimation technique,” *Pattern Recognition Letters*, vol. 17, no. 8, pp. 899–904, 1996.

[57] C. Singh, N. Bhatia, and A. Kaur, “Hough Transform Based Fast Skew Detection and Accurate Skew Correction Methods,” *Pattern Recognition*, vol. 41, no. 12, pp. 3528–3546, 2008.

[58] S. Hinds, J. Fisher, and D. D’Amato, “A document skew detection method using run-length encoding and the hough transform,” in *the 10th International Conference on Pattern Recognition,Atlantic City, USA*, IEEE, vol. 1, pp. 464–468, 1990.

[59] D. S. Le, G. R. Thoma, and H. Wechsler, "Automated page orientation and skew angle detection for binary document images," *Pattern Recognition*, vol. 27, no. 10, pp. 1325–1344, 1994.

[60] L. Likforman-Sulem, A. Hanimyan, and C. Faure, "A hough based algorithm for extracting text lines in handwritten documents," in *the 3rd International Conference on Document Analysis and Recognition,Montreal, Canada*, IEEE, vol. 2, pp. 774–777, 1995.

[61] G. Louloudis, B. Gatos, and C. Halatsis, "Text line detection in unconstrained handwritten documents using a block-based hough transform approach," in *the 9th International Conference on Document Analysis and Recognition, Parana, Brazil* . IEEE, vol. 2, pp. 599–603, 2007.

[62] A. Hashizume, P.-S. Yeh, and A. Rosenfeld, "A Method of Detecting the Orientation of Aligned Components," *Pattern Recognition Letters*, vol. 4, no. 2, pp. 125–132, 1986.

[63] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.

[64] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[65] N. Liolios, N. Fakotakis, and G. Kokkinakis, “Improved document skew detection based on text line connected-component clustering,” in *International Conference on Image Processing, Thessaloniki, Greece*, IEEE, vol. 1, pp. 1098–1101, 2001.

[66] Y. Lu and C. Lim Tan, “A Nearest-Neighbor Chain based Approach to Skew Estimation in Document Images,” *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2315–2323, 2003.

[67] A. Amin and S. WU, “A robust system for thresholding and skew detection in mixed text/graphics documents,” *International Journal of Image and Graphics*, vol. 05, no. 02, pp. 247–265, 2005.

[68] P. Saragiotis and N. Papamarkos, “Local skew correction in documents,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 04, pp. 691–710, 2008.

[69] J. Fabrizio, “A precise skew estimation algorithm for document images using knn clustering and fourier transform,” in *IEEE International Conference on Image Processing, Paris, France*, IEEE, pp. 2585–2588, 2014.

[70] H. Yan, “Skew correction of document images using interline cross-correlation,” *Graphical Models and Image Processing*, vol. 55, no. 6, pp. 538–543, 1993.

[71] B. Gatos, N. Papamarkos, and C. Chamzas, “Skew detection and text line position determination in digitized documents,” *Pattern Recognition*, vol. 30, no. 9, pp. 1505–1519, 1997.

[72] Y. Cao, S. Wang, and H. Li, "Skew Detection and Correction in Document Images Based on Straight-Line Fitting," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1871–1879, 2003.

[73] P. Shivakumara, G. H. Kumar, D. S. Guru, and P. Nagabhushan, "A novel technique for estimation of skew in binary text document images based on linear regression analysis," *Sadhana*, vol. 30, no. 1, pp. 69–85, 2005.

[74] G. Peake and T. Tan, "A general Algorithm for Document Skew Angle Estimation," in *the International Conference on Image Processing.* IEEE, pp. 230–233, 1997.

[75] Gaofeng Meng, Chunhong Pan, Nanning Zheng, and Chen Sun, "Skew estimation of document images using bagging," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1837–1846, 2010.

[76] P. K. Aithal, G. Rajesh, D. U. Acharya, and P. Siddalingaswamy, "A fast and novel skew estimation approach using radon transform," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 5, pp. 337–344, 2013.

[77] J. Sauvola and M. Pietikainen, "Skew Angle Detection Using Texture Direction Analysis," in *the 9th Scandinvian Conference on Image Analysis*, pp. 1099–1106, 1995.

[78] S. Changming and S. Deyi, "Skew and slant correction for document images using gradient direction," in *the 4th International Conference on Document Analysis and Recognition,* IEEE, vol. 1, pp. 142–146, 1997.

[79] B. Epshtein, "Determining Document Skew Using Inter-line Spaces," in *International Conference on Document Analysis and Recognition, Beijing, China.* IEEE, pp. 27–31, 2011.

[80] N. Khorissi, A. Namane, A. Mellit, F. Abdati, Z. Bensalama, and A. Guessoum, "Application of the Wavelet and the Hough Transform for Detecting the Skew Angle in Arabic Printed Documents," in *the 9th International Symposium on Signal Processing and Its Applications, Sharjah, UAE.* IEEE, pp. 1–4, 2007.

[81] A. K. Jain and Y. Zhong, "Page segmentation using texture analysis," *Pattern Recognition*, vol. 29, no. 5, pp. 743–770, 1996.

[82] S.-W. Seong-Whan Lee and D.-S. Dae-Seok Ryu, "Parameter-free geometric document layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1240–1256, 2001.

[83] C.-C. Wu, C.-H. Chou, and F. Chang, "A Machine-learning Approach for Analyzing Document Layout Structures with Two Reading Orders," *Pattern Recognition*, vol. 41, no. 10, pp. 3200–3213, 2008.

[84] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 375–390, 1982.

[85] G. Nagy and S. Seth, “Hierarchical Representation of Optically Scanned Documents,” in *International Conference on Pattern Recognition.* IEEE, pp. 347–349, 1984.

[86] T. Saitoh, M. Tachikawa, and T. Yamaai, “Document Image Segmentation and Text Area Ordering,” in *the 2nd International Conference on Document Analysis and Recognition,Tsukuba, Japan.* IEEE, pp. 323–329, 1993.

[87] J. B. Kruskal, “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem,” *Proceedings of the American Mathematical Society*, vol. 7, no. 1, p. 48, feb 1956.

[88] I. Rabaev, O. Biller, J. El-Sana, K. Kedem, and I. Dinstein, “Text line detection in corrupted and damaged historical manuscripts,” in *the 12th International Conference on Document Analysis and Recognition, Washington DC, USA.* IEEE, pp. 812–816, 2013.

[89] S. S. Bukhari, M. I. A. Al Azawi, F. Shafait, and T. M. Breuel, “Document image segmentation using discriminative learning over connected components,” in *the 8th International Workshop on Document Analysis Systems.* ACM Press, pp. 183–190, 2010.

[90] S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana, “Layout analysis for Arabic historical document images using machine learning,” in *International Conference on Frontiers in Handwriting Recognition, Bari, Italy.* IEEE, pp. 639–644, 2012.

[91] B. Thomas and F. Shafait, “AutoMLP: Simple, Effective, Fully Automated Learning Rate and Size Adjustment,” in *The Learning Workshop, Utah*, pp. 51, 2010.

[92] N. Journet, J.-Y. Ramel, R. Mullot, and V. Eglin, “Document image characterization using a multiresolution analysis of the texture: Application to old documents,” *International Journal of Document Analysis and Recognition*, vol. 11, no. 1, pp. 9–18, jun 2008.

[93] A. Garz and R. Sablatnig, “Multi-scale Texture-based Text Recognition in Ancient Manuscripts,” in *the 16th International Conference on Virtual Systems and Multimedia, Seoul, South Korea.* IEEE, pp. 336–339, 2010.

[94] M. Mehri, P. Gomez-Krämer, P. Héroux, A. Boucher, and R. Mullot, “Texture Feature Evaluation for Segmentation of Historical Document Images,” in *the 2nd International Workshop on Historical Document Imaging and Processing, Washington DC, USA.* ACM Press, pp. 102, 2013.

[95] M. Mehri, N. Nayef, P. Héroux, P. Gomez-Krämer, and R. Mullot, “Learning Texture Features for Enhancement and Segmentation of Historical Document Images,” in *the 3rd International Workshop on Historical Document Imaging and Processing, Washington DC, USA.* ACM Press, pp. 47–54, 2015.

[96] K. Hadjar and R. Ingold, “Physical Layout Analysis of Complex Structured Arabic Documents Using Artificial Neural Nets,” in *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, pp. 170–178, 2004.

[97] M. Baechler, M. Liwicki, and R. Ingold, “Text Line Extraction Using DMLP Classifiers for Historical Manuscripts,” in *the 12th International Conference on Document Analysis and Recognition, Washington DC, USA.* IEEE, pp. 1029–1033, 2013.

[98] A. Fischer, M. Baechler, A. Garz, M. Liwicki, and R. Ingold, “A Combined System for Text Line Extraction and Handwriting Recognition in Historical Documents,” in *the 11th International Workshop on Document Analysis Systems, Tours, France*, pp. 71–75, 2014.

[99] M. Diem, F. Kleber, and R. Sablatnig, “Text classification and document layout analysis of paper fragments,” in *International Conference on Document Analysis and Recognition, Beijing, China.* IEEE, pp. 854–858, 2011.

[100] A. Garz, M. Diem, and R. Sablatnig, “Detecting Text Areas and Decorative Elements in Ancient Manuscripts,” in *the 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India.* IEEE, pp. 176–181, 2010.

[101] A. Garz, R. Sablatnig, and M. Diem, “Layout Analysis for Historical Manuscripts Using SIFT Features,” in *the International Conference on Document Analysis and Recognition, Beijing, China*, pp. 508–512, 2011.

[102] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, “Binarization-Free Text Line Segmentation for Historical Documents Based on Interest Point Clustering,” in

the 10th IAPR International Workshop on Document Analysis Systems, Gold Cost, Australia. IEEE, pp. 95–99, 2012.

[103] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Kdd*, vol. 96, no. 34, pp. 226–231, 1996.

[104] H. Wei, K. Chen, A. Nicolaou, M. Liwicki, and R. Ingold, “Investigation of Feature Selection for Historical Document Layout Analysis,” in *the 4th International Conference on Image Processing Theory, Tools and Applications, Paris, France*, pp. 1–6, 2014.

[105] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, “Page Segmentation for Historical Handwritten Document Images Using Color and Texture Features,” in *the 14th International Conference on Frontiers in Handwriting Recognition, Heraklion, Greece*, pp. 488–493, 2014.

[106] H. Wei, K. Chen, R. Ingold, and M. Liwicki, “Hybrid Feature Selection for Historical Document Layout Analysis,” in *the 14th International Conference on Frontiers in Handwriting Recognition, Heraklion, Greece*, pp. 87–92, 2014.

[107] K. Kise, A. Sato, and M. Iwata, “Segmentation of page images using the area voronoi diagram,” *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 370–382, 1998.

[108] M. J. Burge and G. Monagan, “Using the Voronoi Tessellation for Grouping Words and Multipart Symbols in Documents,” in *SPIE International Sympo-*

sium on Optics, Imaging and Instrumentation, R. A. Melter, A. Y. Wu, F. L. Bookstein, and W. D. K. Green, Eds. International Society for Optics and Photonics, pp. 116–124, 1995.

[109] Y. Lu, Z. Wang, and C. L. Tan, "Word grouping in Document Images based on Voronoi Tessellation," in *International Workshop on Document Analysis Systems*. Springer, pp. 147–157, 2004.

[110] M. Agrawal and D. Doermann, "Voronoi++: A Dynamic Page Segmentation Approach based on Voronoi and Docstrum Features," in *International Conference on Document Analysis and Recognition*. IEEE, pp. 1011–1015, 2009.

[111] Yue Lu and C. Tan, "Constructing Area Voronoi Diagram in Document Images," in *the 8th International Conference on Document Analysis and Recognition*. IEEE, vol. 1, pp. 342–346, 2005.

[112] Y. Xiao and H. Yan, "Text region extraction in a document image based on the delaunay tessellation," *Pattern Recognition*, vol. 36, no. 3, pp. 799–809, 2003.

[113] D. S. Bloomberg, "Multiresolution Morphological Approach to Document Image Analysis," in *the International Conference on Document Analysis and Recognition*, Saint-Malo, France, 1991.

[114] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Improved document image segmentation algorithm using multiresolution morphology," in *International Society for Optics and Photonics*, G. Agam and C. Viard-Gaudin, Eds. International Society for Optics and Photonics, pp. 78740D, 2011.

[115] R. Saabni and J. El-Sana, “Language-Independent Text Lines Extraction Using Seam Carving,” in *International Conference on Document Analysis and Recognition, Beijing, China.* IEEE, pp. 563–568, 2011.

[116] A. Asi, R. Saabni, and J. El-Sana, “Text Line Segmentation for Gray Scale Historical Document Images,” in *the Workshop on Historical Document Imaging and Processing.* ACM Press, pp. 120, 2011.

[117] N. Arvanitopoulos and S. Susstrunk, “Seam Carving for Text Line Extraction on Color and Grayscale Historical Manuscripts,” in *the 14th International Conference on Frontiers in Handwriting Recognition.* IEEE, pp. 726–731, 2014.

[118] R. Cohen, A. Asi, K. Kedem, J. El-Sana, and I. Dinstein, “Robust Text and Drawing Segmentation Algorithm for Historical Documents,” in *the 2nd International Workshop on Historical Document Imaging and Processing* , pp. 110, 2013.

[119] R. Cohen, I. Dinstein, J. El-Sana, and K. Kedem,“Using Scale-Space Anisotropic Smoothing for Text Line Extraction in Historical Documents“ in *International Conference Image Analysis and Recognition.* Springer, pp. 349–358, 2014.

[120] A. Asi, R. Cohen, K. Kedem, and J. El-Sana, “Simplifying the reading of historical manuscripts,” in *the 13th International Conference on Document Analysis and Recognition, Tunis, Tunisia* . IEEE, pp. 826–830, 2015.

[121] Z. Shi and V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Runlength," in *the 1st International Workshop on Document Image Analysis for Libraries, Palo Alto, USA*, pp. 306–312, 2004.

[122] W. Swaileh, K. A. Mohand, and T. Paquet, "Multi-script Iterative Steerable Directional Filtering for Handwritten Text Line Extraction," in *the 13th International Conference on Document Analysis and Recognition, Tunis, Tunisia.* IEEE, pp. 1241–1245, 2015.

[123] A. Alaei, U. Pal, and P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," *Pattern Recognition*, vol. 44, no. 4, pp. 917–928, 2011.

[124] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel, "Background variability modeling for statistical layout analysis," in *the 19th International Conference on Pattern Recognition, Tampa, FL, USA*. IEEE, pp. 1–4, 2008.

[125] Jaekyu Ha, R. Haralick, and I. Phillips, "Document page decomposition by the bounding-box project," in *the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada,* IEEE, vol. 2, pp. 1119–1122, 1995.

[126] D. Sylwester and S. Seth, "A Trainable, Single-pass Algorithm for Column Segmentation," in *the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada,* IEEE, vol. 2, pp. 615–618, 1995.

[127] F. Shafait and T. M. Breuel, “The effect of border noise on the performance of projection-based page segmentation methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 846–851, 2011.

[128] T. M. Breuel, “High Performance Document Layout Analysis,” *Symposium on Document Image Understanding Technology*, vol. 03, pp. 209–218, 2003.

[129] P. Barlas, S. Adam, C. Chatelain, and T. Paquet, “A typed and handwritten text block segmentation system for heterogeneous and complex documents,” in *the 11th International Workshop on Document Analysis Systems.* IEEE, pp. 46–50, 2014.

[130] F. Kleber, R. Sablatnig, M. Gau, and H. Miklas, “Ancient Document Analysis based on Text Line Extraction,” in *the 19th International Conference on Pattern Recognition, ampa, FL, USA.* IEEE, pp. 1–4, 2008.

[131] T. A. Tran, I. S. Na, and S. H. Kim, “Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology,” *International Journal on Document Analysis and Recognition* , vol. 19, no. 3, pp. 191–209, 2016.

[132] H. Wei, M. Baechler, F. Slimane, and R. Ingold, “Evaluation of SVM, MLP and GMM Classifiers for Layout Analysis of Historical Documents,” in *the 12th International Conference on Document Analysis and Recognition, Washington, DC, USA.* IEEE, pp. 1220–1224, 2013.

[133] M. Agrawal and D. Doermann, “Context-aware and Content-based Dynamic Voronoi Page Segmentation,” in *the 8th International Workshop on Document Analysis Systems - DAS ’10.* New York, USA, ACM Press, pp. 73–80, 2010.

[134] K. Kise, A. Sato, and K. Matsumoto, “Document Image Segmentation as Selection of Voronoi Edges,” in *the Workshop on Document Image Analysis* . IEEE, pp. 32–39, 1997.

[135] Z. Shi, S. Setlur, and V. Govindaraju, “A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines,” in *the 10th International Conference on Document Analysis and Recognition.* IEEE, pp. 176–180, 2009.

[136] S. S. Bukhari, F. Shafait, and T. M. Breuel, “Script-independent Handwritten Textlines Segmentation Using Active Contours,” in *the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain.* IEEE, pp. 446–450, 2009.

[137] R. Manmatha, C. Han, and E. M. Riseman, “Word Spotting: A New Approach to Indexing Handwriting,” in *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, USA,* IEEE, pp. 631–637, 1996.

[138] S. Lu, L. Li, and C. L. Tan, “Document image retrieval through word shape coding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1913–1918, 2008.

[139] Q. A. Bui, M. Visani, and R. Mullot, “Unsupervised Word Spotting Using a Graph Representation based on Invariants,” in *the 13th International Conference on Document Analysis and Recognition, Tunis, Tunisia* , IEEE, pp. 616–620, 2015.

[140] S. Sudholt, L. Rothacker, and G. A. Fink, “Learning Local Image Descriptors for Word Spotting,” in *the 13th International Conference on Document Analysis and Recognition*. IEEE, pp. 651–655, 2015.

[141] A. Ghorbel, J.-M. Ogier, and N. Vincent, “A Segmentation Free Word Spotting for Handwritten Documents,” in *the 13th International Conference on Document Analysis and Recognition* . IEEE, pp. 346–350, 2015.

[142] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis, “Keyword-guided word spotting in historical printed documents using synthetic data and user feedback,” *International Journal of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 167–177, 2007.

[143] E. Vidal, A. H. Toselli, and J. Puigcerver, “High performance query-by-example keyword spotting using query-by-string techniques,” in *the 13th International Conference on Document Analysis and Recognition, Tunis, Tunisia.* IEEE, pp. 741–745, 2015.

[144] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, “Efficient segmentation-free keyword spotting in historical document collections,” *Pattern Recognition*, vol. 48, no. 2, pp. 545–555, 2015.

[145] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Lladós, and A. Fornés, "A Novel Learning-free Word Spotting Approach based on Graph Representation," in *the 11th International Workshop on Document Analysis Systems.* IEEE, pp. 207–211, 2014.

[146] R. Ahmed, W. G. Al-Khatib, and S. Mahmoud, "A survey on handwritten documents word spotting," *International Journal of Multimedia Information Retrieval*, vol. 6, no. 1, pp. 31–47, 2017.

[147] T. M. Rath and R. Manmatha, "Word Image Matching Using Dynamic Time Warping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 2, pp. 2–2, 2003.

[148] S. Srihari, H. Srinivasan, P. Babu, and C. Bhole, "Spotting Words in Handwritten Arabic Documents," in *Document Recognition and Retrieval XIII*, vol. 6067. International Society for Optics and Photonics, pp. 606702, 2006.

[149] T. Adamek and N. E. O'Connor, "A multiscale representation method for nonrigid shapes with a single closed contour," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 742–753, 2004.

[150] C. Wieprecht, L. Rothacker, and G. A. Fink, "Word spotting in historical document collections with online-handwritten queries," in *in the 12th Workshop on Document Analysis Systems, Santorini, Greece* , IEEE, pp. 162–167, 2016.

[151] S. Bai, L. Li, and C. L. Tan, "Keyword Spotting in Document Images through Word Shape Coding," in *the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain* IEEE, pp. 331–335, 2009.

[152] A. K. Bhunia, P. P. Roy, and U. Pal, "Zone-based keyword spotting in bangla and devanagari documents," *arXiv preprint arXiv:1712.01434*, 2017.

[153] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence* vol. PP, no. 99, pp. 1–1, 2017.

[154] Y. Elfakir, G. Khaissidi, M. Mrabti, and D. Chenouni, "Handwritten Arabic documents indexation using hog feature," *International Journal of Computer Applications*, vol. 126, no. 9, 2015.

[155] A. Hast and A. Fornés, "A Segmentation-free Handwritten Word Spotting Approach by Relaxed Feature Matching," in *the 12th Workshop on Document Analysis Systems.* IEEE, pp. 150–155, 2016.

[156] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Segmentation-free word spotting with exemplar svms," *Pattern Recognition*, vol. 47, no. 12, pp. 3967–3978, 2014.

[157] D. Aldavert, M. Rusiñol, R. Toledo, and J. Lladós, "A study of bag-of-visual-words representations for handwritten keyword spotting," *International Journal on Document Analysis and Recognition* , vol. 18, no. 3, pp. 223–234, 2015.

[158] M. Stauffer, A. Fischer, and K. Riesen, "Graph-based keyword spotting in historical handwritten documents," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR).* Springer, pp. 564–573, 2016.

[159] St. Michael, F. Andreas and R. Kaspar, "Speeding-up Graph-based Keyword Spotting in Historical Handwritten Documents," in *International Workshop on Graph-Based Representations in Pattern Recognition.* Springer, pp. 83–93, 2017.

[160] J. Li, Z.-G. Fan, Y. Wu, and N. Le, "Document Image Retrieval with Local Feature Sequences," in *the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain,* IEEE, pp. 346–350, 2009.

[161] D. Aldavert, M. Rusinol, R. Toledo, and J. Lladós, "Integrating Visual and Textual Cues for Query-by-string Word Spotting," in *the 12th International Conference on Document Analysis and Recognition, Washington, DC, USA*, IEEE, pp. 511–515, 2013.

[162] D.-R. Lee, W. Hong, and I.-S. Oh, "Segmentation-free Word Spotting Using SIFT," in *the IEEE Southwest Symposium on Image Analysis and Interpretation, Santa Fe, USA*, IEEE, pp. 65–68, 2012.

[163] S. Marinai, E. Marino, and G. Soda, "Font adaptive word indexing of modern printed documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1187–1199, 2006.

[164] S. Ortmanns, H. Ney, and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," *Computer Speech & Language*, vol. 11, no. 1, pp. 43–72, 1997.

[165] S. En, C. Petitjean, S. Nicolas, and L. Heutte, "A scalable pattern spotting system for historical documents," *Pattern Recognition*, vol. 54, pp. 149–161, 2016.

[166] P. Riba, J. Lladãs, and A. Fornés, "Handwritten Word Spotting by Inexact Matching of Grapheme Graphs," in *the 13th International Conference on Document Analysis and Recognition.* IEEE, pp. 781–785, 2015.

[167] S. Wshah, G. Kumar, and V. Govindaraju, "Statistical script independent word spotting in offline handwritten documents," *Pattern Recognition*, vol. 47, no. 3, pp. 1039–1050, 2014.

[168] M. Khayyat, L. Lam, and C. Y. Suen, "Learning-based word spotting system for Arabic handwritten documents," *Pattern Recognition*, vol. 47, no. 3, pp. 1021–1030, 2014.

[169] F. Zirari, A. Ennaji, S. Nicolas, and D. Mammass, "A Methodology to Spot Words in Historical Arabic Documents," in *the ACS International Conference on Computer Systems and Applications* . IEEE, pp. 1–4, 2013.

[170] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[171] J. A. Rodríguez-Serrano and F. Perronnin, “Handwritten word-spotting using hidden markov models and universal vocabularies,” *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[172] K. Terasawa and Y. Tanaka, “Slit Style HOG feature for Document Image Word Spotting,” in *the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain.* IEEE, pp. 116–120, 2009.

[173] J. A. Rodriguez and F. Perronnin, “Local gradient histogram features for word spotting in unconstrained handwritten documents,” *in the 1st International Conference on Frontiers in Handwriting Recognition*, pp. 7–12, 2008.

[174] T. Van der Zant, L. Schomaker, and K. Haak, “Handwritten-word spotting using biologically inspired features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1945–1957, 2008.

[175] T. M. Rath and R. Manmatha, “Features for word spotting in historical manuscripts,” in *the 7th International Conference on Document Analysis and Recognition, Edinburgh, UK*, IEEE, pp. 218–222, 2003.

[176] L. Rothacker and G. A. Fink, “Segmentation-free Query-by-string Word Spotting with Bag-of-Features HMMs,” in *the 13th International Conference on Document Analysis and Recognition, Tunis, Tunisia* , IEEE, pp. 661–665, 2015.

[177] A. Shahab, "UW3 and unlv datasets", German Research Center for Artificial Intelligence, www.iapr-tc11.org/mediawiki/index.php/Table_Ground_Truth_for_the_UW3_and_UNLV_datasets, , 2013.

[178] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A Realistic Dataset for Performance Evaluation of Document Layout Analysis," in *the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain.* IEEE, pp. 296–300, 2009.

[179] R. S. Saad, R. I. Elanwar, N. S. A. Kader, S. Mashali, and M. Betke, "Bce-Arabic-v1 dataset: Towards interpreting Arabic document images for people with visual impairments," in *the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, New York, USA, ACM Press, pp. 1–8, 2016.

[180] E. Saund, J. Lin, and P. Sarkar, "Pixlabeler: User interface for pixel-level labeling of elements in document images," in *the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain*, IEEE, pp. 646–650, 2009.

[181] D. D. Elena Zotkina, Himanshu Suri, "Gedi: Groundtruthing environment for document images," https://lampsrv02.umiacs.umd.edu/projdb/project.php?id=53, 2013.

[182] K. Chen, M. Seuret, H. Wei, M. Liwicki, and R. Ingold,, "Document, image, and video analysis DLA tool," http://diuf.unifr.ch/main/hisdoc/divadia.

[183] C. Papadopoulos, S. Pletschacher, C. Clausner, and A. Antonacopoulos, "The Impact Dataset of Historical Document Images." the Workshop on Historical Document Imaging and Processing, Washington DC, USA, pp. 123–130, 2013.

[184] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2, pp. 139–152, 2007.

[185] V. Lavrenko, T. M. Rath, and R. Manmatha, "Holistic Word Recognition for Handwritten Historical Documents," in *the 1st International Workshop on Document Image Analysis for Libraries.* IEEE, pp. 278–287, 2004.

[186] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription Alignment of Latin Manuscripts Using Hidden Markov Models," in *the 2011 Workshop on Historical Document Imaging and Processing.* ACM, pp. 29–36, 2011.

[187] A. Raza, I. Siddiqi, A. Abidi, and F. Arif, "An Unconstrained Benchmark Urdu Handwritten Sentence Database with Automatic Line Segmentation," in *International Conference on Frontiers in Handwriting Recognition.* IEEE, pp. 491–496, 2012.

[188] W. Pantke, M. Dennhardt, D. Fecker, V. Margner, and T. Fingscheidt, "An Historical Handwritten Arabic Dataset for Segmentation-free Word Spotting - HADARA80P," in *the 14th International Conference on Frontiers in Handwriting Recognition, Heraklion, Greece,* IEEE, pp. 15–20, 2014.

[189] Y. Zheng and D. Doermann, "LAMP dataset of Layer Separation,", https://lampsrv02.umiacs.umd.edu/projdb/project.php?id=61, 2010.

[190] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "Historical Document Layout Analysis Competition," in *International Conference on Document Analysis and Recognition, Beijing, China,* IEEE, pp. 1516–1520, 2011.

[191] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR2009 handwriting segmentation contest," *International Journal on Document Analysis and Recognition* , vol. 14, no. 1, pp. 25–33, 2011.

[192] F. Shafait, D. Keysers, and T. Breuel, "Pixel-accurate representation and evaluation of page segmentation in document images," in *the 18th International Conference on Pattern Recognition.* IEEE, pp. 872–875, 2006.

[193] V. Nikos and K. Ergina, "Complex layout analysis based on contour classification and morphological operations," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 220–229, 2017.

[194] C. Christian, P. Stefan, and A. Apostolos, "Scenario Driven in-depth Performance Evaluation of Document Layout Analysis Methods," in *the International Conference on Document Analysis and Recognition.* IEEE, pp. 1404–1408, 2011.

[195] A. Antonacopoulos,S. Pletschacher, D. Bridson and C. Papadopoulos, "ICDAR2009 page segmentation competition," in *the 10th International Confer-*

ence on Document Analysis and Recognition, Barcelona, Spain, pp. 1370–1374, 2009.

[196] D. Bridson and A. Antonacopoulos, “A Geometric Approach for Accurate and Efficient Performance Evaluation of Layout Analysis Methods,” in *the 19th International Conference on Pattern Recognition.* IEEE, pp. 1–4, 2008.

[197] S. S. Bukhari, F. Shafait, and T. M. Breuel, “Layout Analysis of Arabic Script Documents,” in *Guide to OCR for Arabic Scripts*, V. Märgner and H. El Abed, Eds. London: Springer London, pp. 35–53, 2012.

[198] Y. Y. Tang, S. W. Lee, and C. Y. Suen, “Automatic Document Processing: A Survey,” *Pattern Recognition*, vol. 29, no. 12, pp. 1931–1952, 1996.

[199] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, “The Delaunay Document Layout Descriptor,” in *ACM Symposium on Document Engineering*, ACM Press, pp. 167–175, 2015.

[200] State Library of Berlin. Arabische sammelhandschrift. http://digital.staatsbibliothek-berlin.de/, 2014

[201] w3shools.come. extensible markup language. https://www.w3schools.com/

[202] G. Nagy, “Twenty years of document image analysis in PAMI ,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 38–62, 2000.

[203] M. Maroua, H. Pierre, G.-K. Petra, and M. Rémy, "Texture feature benchmarking and evaluation for historical document image analysis," *International Journal on Document Analysis and Recognition*, vol. 20, no. 1, pp. 1–35, 2017.

[204] B. Micheal, B. Jean-Luc, and I. Rolf, "Semi-automatic annotation Tool for Medieval Manuscripts," in *the International Conference on Frontiers in Handwriting Recognition, Kolkata, India.* IEEE, pp. 182–187, 2010.

[205] B. Thomas, "An algorithm for Finding Maximal Whitespace Rectangles at Arbitrary Orientations for Document Layout Analysis," in *the 7th International Conference on Document Analysis and Recognition, Edinburgh, UK.* IEEE, pp. 66–70, 2003.

[206] B. Thomas, "Two geometric algorithms for layout analysis," in *International workshop on document analysis systems.* Springer, pp. 188–199, 2002.

[207] L. Stephen WK, "A local-to-global approach to complex document layout analysis." in *MVA*, pp. 431–434, 1994.

[208] C. Kai, Y. Fei, and L. Cheng-Lin, "Hybrid page segmentation with efficient whitespace rectangles extraction and grouping," in *the 12th International Conference on Document Analysis and Recognition, Washington DC, USA.* IEEE, pp. 958–962, 2013.

[209] C. Maurer, Rensheng Qi, and V. Raghavan, "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary di-

mensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 265–270, 2003.

[210] T. Faisal and S. AlMaadeed, "Enabling indexing and retrieval of historical Arabic manuscripts through template matching based word spotting," in *the 1st International Workshop on Arabic Script Analysis and Recognition, Nancy, France*, IEEE, pp. 57–63, 2017.

[211] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *the 7th IEEE international conference on Computer vision, Kerkyra, Greece*, vol. 2, IEEE, pp. 1150–1157, 1999.

[212] L. Chergui and M. Kef, "SIFT descriptors for Arabic handwriting recognition," *International Journal of Computational Vision and Robotics*, vol. 5, no. 4, pp. 441–461, 2015.

[213] K. Zagoris, I. Pratikakis, and B. Gatos, "Unsupervised word spotting in historical handwritten document images using document-oriented local features," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 4032–4041, 2017.

[214] A. Batur, G. Tursun, M. Mamut, N. Yadikar, and K. Ubul, "Uyghur printed document image retrieval based on SIFT features," *Procedia Computer Science*, vol. 107, pp. 737–742, 2017.

[215] X. Zhang and C. L. Tan, "Segmentation-free Keyword Spotting for Handwritten Documents based on Heat Kernel Signature," in *the 12th International Conference on Document Analysis and Recognition.* IEEE, pp. 827–831, 2013.

[216] X. Zhang, U. Pal, and C. L. Tan, “Segmentation-free Keyword Spotting for Bangla Handwritten Documents,” in *the 14th International Conference on Frontiers in Handwriting Recognition.* IEEE, pp. 381–386, 2014.

[217] M. O. Assayony and S. A. Mahmoud, “An enhanced bag-of-features framework for Arabic handwritten sub-words and digits recognition,” *Journal of Pattern Recognition and Intelligent Systems*, vol. 4, no. 1, pp. 27–38, 2016.

[218] S. Yao, Y. Wen, and Y. Lu, “HOG based Two-directional Dynamic Time Warping for Handwritten Word Spotting,” in *the 13th International Conference on Document Analysis and Recognition.* IEEE, pp. 161–165, 2015.

[219] M. L. Bouined, H. Nemmour, and Y. Chibani, “New Gradient Descriptor for Keyword Spotting in Handwritten Documents,” in *the International Conference on Advanced Technologies for Signal and Image Processing* . IEEE, pp. 1–5, 2017.

[220] J. Lladós and G. Sánchez, “Indexing Historical Documents by Word Shape Signatures,” in *the 9th International Conference on Document Analysis and Recognition*, vol. 1, IEEE, pp. 362–366, 2007.

[221] X. Guo, H. Wei, and X. Su, “A Case Study of BoVW for Keyword Spotting on Historical Mongolian Document Images,” in *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics* . IEEE, pp. 374–378, 2016.

[222] J. Lladós, M. Rusiñol, A. Fornés, D. Fernández, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *International journal of pattern recognition and artificial intelligence*, vol. 26, no. 05, pp. 1263002, 2012.

[223] M. Rusinol, D. Aldavert, R. Toledo, and J. Llados, "Browsing Heterogeneous Document Collections by a Segmentation-free Word Spotting Method," in *the International Conference on Document Analysis and Recognition.* IEEE, pp. 63–67, 2011.

[224] Q. B. Dang, V. P. Le, M. M. Luqman, M. Coustaty, C. Tran, and J.-M. Ogier, "Camera-based Document Image Retrieval System Using Local Features-comparing SRIF with LLAH, SIFT, SURF and ORB," in *the 13th International Conference on Document Analysis and Recognition.* IEEE, pp. 1211–1215, 2015.

[225] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, pp. 1–1, 2001.

[226] J.-G. Wang, J. Li, W.-Y. Yau, and E. Sung, "Boosting dense SIFT Descriptors and Shape Contexts of Face Images for Gender Recognition," in *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops.* IEEE, pp. 96–102, 2010.

[227] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "SIFT Flow: Dense Correspondence Across Different Scenes," in *European conference on computer vision*. Springer, pp. 28–42, 2008.

[228] W. Abu-Ain, S. N. H. S. Abdullah, B. Bataineh, T. Abu-Ain, and K. Omar, "Skeletonization algorithm for binary images," *Procedia Technology*, vol. 11, pp. 704–709, 2013.

[229] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 9, pp. 869–885, 1992.

[230] J-M. Geusebroek, A. Smeulders J. Van De Weijer, "Fast anisotropic gauss filtering," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, 2003.

[231] X. You and Y. Y. Tang, "Wavelet-based approach to character skeleton," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1220–1231, 2007.

[232] E. Ebastien and G. Petra and O. Jean-Marc, "A comprehensive survey of mostly textual document segmentation algorithms since 2008," *Pattern Recognition*, vol. 64, no. 5, pp. 1–14, 2017.

[233] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, pp. 1–2, 2004.

[234] X.-h. Tan, W.-h. Bi, X.-l. Hou, and W. Wang, “Reliability analysis using radial basis function networks and support vector machines,” *Computers and Geotechnics*, vol. 38, no. 2, pp. 178–186, 2011.

[235] G. M. BinMakhashen and E.-S. M. El-Alfy, “Fusion of Multiple Texture Representations for Palmprint Recognition Using Neural Networks,” in *International Conference on Neural Information Processing.* Springer, 2012, pp. 410–417.

[236] A. Asi, “MATLAB CODE and Bukhari dataset,”, http://www.cs.bgu.ac.il/ $~$abedas,2014.

Vitae

- **Contact Information:**
    - Name: Galal Munassar Abdullah Bin Makhashen
    - Nationality: Yemen
    - Email: *binmakhashen@gmail.com, binmakhashen@kfupm.edu.sa*
    - Permanent Address: Hadhramout Governorate, Rep. of Yemen
    - Present Address: Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, P.O. Box 5010, Dhahran 31261, Saudi Arabia.
- **Education Background:**
    - Received Bachelor of Science (BSc) degree in Computer Science in 2005 from Hadhramout University of Science and Technology (HUST)
    - Received Master of Science (MSc) degree in Computer Science from King Fahd University of Petroleum and Minerals (KFUPM) in 2013
    - Submitted this dissertation to fulfill the requirements of a PhD degree in Computer Science and Engineering from KFUPM.