ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

PAGE SEGMENTATION IN AMHARIC DOCUMENT IMAGE COLLECTIONS

GEDION ASSEFA

JUNE 2013

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

PAGE SEGMENTATION IN AMHARIC DOCUMENT IMAGE COLLECTIONS

A Thesis Submitted to the School of Information Science of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Science

By
GEDION ASSEFA

JUNE 2013

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

PAGE SEGMENTATION IN AMHARIC DOCUMENT IMAGE COLLECTIONS

By
GEDION ASSEFA

Name and signature of members of the examining Board

Name	**Title**	**Signature**	**Date**
_____	Chairperson	_____	_____
Million Meshesha (Ph.D)	Advisor	_____	_____
Dereje Teferi (Ph.D)	Examiner	_____	_____

Dedication

To my Mother,

Who sacrifices everything she has for my success!

Acknowledgement

First of all, I would like to express my gratitude to my advisor, Million Meshesha (Ph.D), for his support, constructive comments and suggestion throughout the thesis work.

My deepest gratitude also goes to my friends Anteneh T., Wondwossen T., Zelalem M., Abreham Y., Temesgen T. Henok A., Abel T., and Yigermal H. for their encouragement and support in different aspects throughout my study.

I would also like to thank Biniam A. and Kibrom for providing me resources, and supporting ideas.

Finally, my special thanks goes to my family specially to my wife, Frehiwot Alemu, who love, support and encourage me all the time. I am grateful to have you all.

Abstract

The advancement and accessibility of digital computers and the introduction of the Internet and World Wide Web resulted in massive information explosion all over the world. Large amount of handwritten, typewritten and printed documents contain numerous information and knowledge of different areas. To make the information and knowledge embedded in these documents accessible to the public, it is desirable to digitize, organize and develop retrieval systems for such kind of documents. In response to this need, researchers are moving towards recognition-free approach since optical character recognition OCR engines have various limitations.

Researches have been conducted to develop Amharic document image retrieval (DIR) system without explicit recognition that retrieve information from document images relying on image features only. However, effectiveness of the system is highly affected by segmentation errors at word-level. Moreover, the system does not work on real-life document images in which images, graphics, logos, tables, etc. are embedded. This study attempts to integrate effective page segmentation technique that can work on documents which contain images, graphics, tables, etc. and improve word level segmentation.

Accordingly, page segmentation algorithms namely: Hough transforms, Connected Components (CC), Horizontal Run Length Smoothing (HRLS), Dilation and Watershed are tested. The performance evaluation showed that the integration of CC and Dilation is the best combination. Average Match Score of 0.865 in different level noisy document images, 0.93 in typewritten documents, 0.97 in documents containing pictures, 0.97 in documents containing tables and 0.45 in handwritten documents ('kum tshihuf') is scored. On the average, an increase of 2.34% F-Measure is scored in different level noisy document images. Degraded features of old documents, slimness of typewritten characters and font size variation had a great impact on the performance of the system which needs further attention by future researches.

Table of Contents

Dedication i

Acknowledgement ii

List of Tables vii

List of Figures viii

List of Equations ix

List of Algorithms x

List of Sample Codes (Listings) xi

List of Acronyms xii

CHAPTER ONE INTRODUCTION 1

1.1. Background 1

1.2. Statement of the Problem and Justification 3

1.3. Objectives of the Study 5

1.3.1. General Objective 5

1.3.2. Specific Objectives 5

1.4. Scope and Limitation of the Study 6

1.5. Methodology of the Study 6

1.5.1. Literature Review 6

1.5.2. Dataset Collection 7

1.5.3. Implementation Tool 7

1.5.4. Testing Procedure 7

1.6. Significance of the Research 8

1.7. Organization of the Study 9

CHAPTER TWO LITERATURE REVIEW 10

2.1. Information Retrieval (IR) 10

2.2. Document Image Retrieval (DIR) 11

2.2.1. Recognition Based Document Image Indexing and Retrieval 12

2.2.2. DIR Based on Keyword Spotting 13

2.2.3. DIR Based on Layout Structural Similarity 13

2.2.4. Signature Based DIR 14

2.2.5. DIR Based On Logo Matching 14

2.3. Steps in DIR 14

2.4. Document Image Segmentation 17
2.4.1. Text/Graphic Segmentation 18
2.4.2. Text Line and Word Segmentation 19
2.5. Document Image Segmentation Techniques 20
2.5.1. Top-Down Techniques 20
2.5.2. Bottom-Up Techniques 21
2.5.3. Hybrid Techniques 24
2.5.4. Watershed Based Image Segmentation 24
2.6. The Amharic Writing System 27
2.6.1. Amharic Characters 28
2.6.2. Amharic Numeration System 29
2.6.3. Amharic Punctuation Marks 30
2.7. Documents Written in Amharic Script 30
2.7.1. Printed Documents 30
2.7.2. Typewritten Documents 31
2.7.3. Handwritten Documents 31
2.8. Challenges of Amharic Script 31
2.8.1. Existance of Character Variants 32
2.8.2. Feature Similarity Among Characters 32
2.8.3. Font Variations 32
2.8.4. Formation of Compound Words 33
2.8.5. Rich Morphology 33
2.9. Related Research Works 34
2.9.1. Global Research Works 34
2.9.2. Local Research Works 36
CHAPTER THREE PAGE SEGMENTATION TECHNIQUES 38
3.1. Architecture of Amharic DIR System 38
3.2. Page Segmentation Techniques 39
3.2.1. Watershed Algorithm Based on Connected Components 40
3.2.2. Run Length Smoothing 41
3.2.3. Dilation 41
3.2.4. Connected Component Labeling 42

3.2.5. Hough Transform Algorithm 44
3.3. Performance Evaluation 47
3.3.1. GCE 47
3.3.2. Match Score 48
3.3.3. Precision, Recall and F-Measure 48
CHAPTER FOUR EXPERIMENTATION 50
4.1. Dataset Preparation 50
4.2. Page Segmentation in Amharic Document Images 51
4.2.1. Connected Components Labeling 51
4.2.2. Components Width, Height and Area Analysis 52
4.2.3. Hough Transform 55
4.2.4. Dilation 56
4.2.5. Watershed Segmentation 57
4.2.6. Horizontal Run Length Smoothing (HRLS) 59
4.3. Proposed Segmentation Technique 61
4.4. Performance Result 64
4.5. Integrating the Proposed Segmentation Algorithm with Amharic DIR System 65
4.6. Experimental Result of Amharic DIR System 66
4.7. Findings and Challenges 69
CHAPTER FIVE CONCLUSIONS AND RECOMMENDATIONS 71
5.1. Conclusions 71
5.2. Recommendation 72
References 74
Appendix I: Amharic Characters 80
Appendix II: Sample Codes 81
Declaration 84

List of Tables

Table 2.1 - The Seven Orders of Amharic Writing System [12] **28**
Table 2.2 - Characters Representing Labialized Velar Consonants [12] **29**
Table 2.3 - Ethiopic Numerals [11] **29**
Table 2.4 – Different Amharic Font Types **31**
Table 2.5 – Font Type, Size and Style Variation [11] **33**
Table 4.1 – Performance of Dilation and HRLS in Identifying Words **60**
Table 4.2 – Performance of Combined Segmentation Technique **64**
Table 4.3 – System Performance on Low - Level Noisy Document Images **66**
Table 4.4 – System Performance on Medium - Level Noisy Document Images **67**
Table 4.5 – System Performance on High - Level Noisy Document Images **68**
Table 4.6 – System Performance on Very High - Level Noisy Document Images **68**
Table 4.7 – System Performance on Different Document Image Types **69**

List of Figures

Figure 2.1 - Document Image Indexing Approaches [17] **12**
Figure 2.2 – The Overall Structure of Recognition Free DIR System [10] **15**
Figure 2.3 – A Composite Document and Its Parts **19**
Figure 2.4 - Segmentation of Figures and Figure Caption Candidates by CC Analysis [49] **22**
Figure 2.5 - Watershed Lines and Catchment Basins [69] **25**
Figure 2.6 - The Genetic Structure of Amharic Language [12] **28**
Figure 3.1 - Architecture of the Proposed Amharic Document Image Retrieval System **39**
Figure 3.2 - Basic Concept of Connected Components Approach [69] **40**
Figure 3.3 – Effect of Dilation [55] **42**
Figure 3.4 - Alternative Representation of Straight Line in (ρ, θ) Format [75] **46**
Figure 4.1 – Result of CC Labeling: **52**
Figure 4.2 –CC Area, Height and Width Analysis: **54**
Figure 4.3 – Implementation of Hough Transform with Different Thresholds: **56**
Figure 4.4 – Dilation Using Different Thresholds: **57**
Figure 4.5 – Implementation of Watershed Algorithm: **58**
Figure 4.6 – Dilation and HRLS Connected Words **60**
Figure 4.7 – Flow of Proposed Technique **62**
Figure 4.8 – Result of Proposed Technique at Each Steps: **63**
Figure 4.9 – Effect of Featuer Degradation: **67**
Figure 4.10 – The Effect of Thresholding in the Presence of Shadow **70**
Figure 4.11 – Thresholding Degraded Image **70**

List of Equations

Equation 3.1 – Thresholding Equation RLSA. 41
Equation 3.2 –Dilation Formula 42
Equation 3.3 – Slop Intercept Formula 1 45
Equation 3.4 – Slop Intercept Formula 2 45
Equation 3.5 –Alternative Representation of Lines 46
Equation 3.6 – Local Refinement Error Formula 47
Equation 3.7 – Global Consistency Error (GCE) 48
Equation 3.8 – Local Consistency Error (LCE) 48
Equation 3.9 – Match Score 48
Equation 3.10 – Precision 49
Equation 3.11 – Recall 49
Equation 3.12 – F-Measure 49

List of Algorithms

Algorithm 3.1 - Connected - Region Extraction (One Pass)..... 43

Algorithm 3.2 - Connected - Region Extraction (Two Pass) 44

Algorithm 3.3 - Generalized Hough Transform Algorithm..... 45

List of Sample Codes (Listings)

Listing 4.1 - Implementation of Connected Components 51

Listing 4.2 - Implementation of Hough Transform 55

Listing 4.3 - Implementation of Dilation 57

Listing 4.4 - Implementation of Watershed 58

Listing 4.5 - Implementation of Horizontal Run Length Smoothing 59

Listing 4.6 - Integrating the MATLAB Implementation with Java 65

List of Acronyms

AI:	Artificial Intelligence
CC:	Connected Component
DIR:	Document Image Retrieval
GCE:	Global Consistency Error
HRLS:	Horizontal Run Length Smoothing
IR:	Information Retrieval
LCE:	Local Consistency Error
MATLAB:	MATrix LABoratory
OCR:	Optical Character Recognition
RLSA:	Run Length Smoothing Algorithm
RXYC:	Recursive X-Y Cuts

CHAPTER ONE

INTRODUCTION

1.1. Background

The advancement and accessibility of digital computers and the introduction of the Internet and World Wide Web resulted in massive information explosion all over the world. Information is embedded in data usually in the form of a document or an audible or visible communication [1]. Most information is still recorded, stored, and distributed in paper format. The widespread use of computers, with the introduction of personal computers and word-processors in the late 1980's had the effect of increasing, instead of reducing, the amount of information held on paper [2]. The progress of technology and research in the fields of Information Retrieval (IR), Artificial Intelligence (AI), Digital Image Processing and Pattern Recognition brings the need to digitize, store, query, search and retrieve different documents to make the information accessible for public use efficiently and accurately [3]. IR primarily deals with representation, storage, organization and access to information items [4].

In a real world, there are two document types that need to be retrieved for users based on their need. These are text documents available in digital (ASCII or UNICODE representation) format and scanned image documents. Accordingly, document retrieval is either a text based or an image based [5]. As noted by Mesfin [5], researchers attempted two approaches for document image retrieval (DIR), recognition based DIR which use Optical Character Recognition (OCR) to convert the image documents to equivalent ASCII or UNICODE text documents or recognition free (DIR without explicit recognition). However, effective access to image document sources is limited by the lack of efficient retrieval schemes [6].

Using OCR tool is still not perfect and requires human correction that could incur a prohibitive cost for a large document collection [7]. Similarly, for a huge number of document images archived in digital libraries using OCR for the retrieval purpose is wasteful of resources and has been proven prohibitively expensive, especially the difficulty in post-OCR corrections [8].

Designing efficient Amharic DIR system without explicit recognition is one basic remedy to overcome limitations of the recognition system (OCR) in the short run, because character recognition from image documents that are printed in Amharic script is a challenging task due to: printing variations, large number of characters in the script, visual similarity of Amharic characters in shape, language related issues and degradations of documents [9].

There are six major tasks involved with retrieval of documents from document image collections. These are: preprocessing, segmentation, feature extraction, indexing, matching and displaying relevant document images in ranked order [10].

Preprocessing involves binarization (threshholding), skew correction and image enhancements such as filtering out noise and increasing the contrast [9]. Binarization is to automatically choose a threshold that separates the foreground from the background region of the document images.

After preprocessing, the document images need to be segmented to separate the set of words in them. Segmentation occurs at two levels. On the first level, text, graphics and other regions are separated. On the second level, text lines and words in the text image are located [9].

Then, feature extraction follows, which involves extracting the meaningful information from the document images [9]. Document images are indexed based on features and then searched to retrieve relevant documents as per information need of users. According to Abrehan [12], it is necessary to develop an index to organize bag of words in order to speed up searching from document collections.

In DIR without explicit recognition, when users enters their query in text format the query will be converted to image (by a process called rendering) in order to be compared with set of document images. Feature of the query image is extracted the same way as feature of document image words are extracted. Then, matching in document images can identify the word images of the documents that are more similar to the query word using the extracted feature vectors [11]. Finally, relevant documents are retrieved and presented to the user in ranked order based on the similarity measure scores. The main aim of ranking is to sort the retrieved documents according to their degree of relevance to the query provided by the user [10].

Similarity measurement, which is highly important to retrieve relevant documents, is done between query word images and document word images. Real-life documents usually contain text, images, graphics, logos, tables, etc. Page segmentation algorithms are used to extract word images from real-life document images. Therefore, performance of DIR system on real-life document images would be in question unless page segmentation technique which can work on document images that contains images, graphics, logos, tables, etc. is designed [10]. The difficulty of page segmentation in noisy document images and in documents containing pictures, graphics, and tables also need to be addressed.

1.2. Statement of the Problem and Justification

Large amount of handwritten, typewritten and printed documents contain numerous information and knowledge of different areas. To make the information and knowledge embedded in these documents accessible to the public, it is desirable to digitize, organize and develop retrieval systems for such kind of documents. In response to this need, researchers are moving towards recognition-free approach (DIR) since OCR has various limitations [10]. For example if an OCR engine miss a single character in a given word, there is no possibility of retrieval for that word since recognition based approaches often use exact matching for similarity measurement.

There are attempts to develop Amharic DIR system. Mesfin [5] designed a DIR system without explicit recognition that searches for relevant documents by accepting a single query word form users. Abreham [12] continued to enhance the performance of the system by integrating indexing technique. Then, Adane [11] came up with better matching and feature extraction techniques. A research done by Biniam [10] integrates noise removal techniques for real-life Amharic document images. Biniam [10] also came up with a system that accept and process multiple words of query. The above mentioned researches addressed noise removal, thresholding, indexing, and feature extraction techniques for Amharic DIR system. However, Biniam [10] reported that the system's effectiveness is highly affected by segmentation errors at word-level and recommended further research work to tackle this problem.

Real-life document images usually contain both text and non-text elements (images, graphics, logos, signatures, tables, seals, etc.). The system developed so far does not work on real-life document images that embed images, graphics, logos, tables and lines in addition to text. This is because the system is not able to segment text and non-text regions in document images before word level-segmentation. Hence, lack of effective page segmentation technique may result in segmenting non-text areas as text (word), segmenting a single word as two or more words or merging more than one word together as a single word which adversely affects effectiveness of DIR systems. Therefore, the main purpose of this study is to explore effective page segmentation technique to enhance the performance of Amharic DIR.

To fill the above mentioned gap, this research addressed the following research questions:

- What are special features of Amharic words and word separators in real-life documents?
- Which page segmentation technique is effective for identifying text and non-text regions in Amharic document images?

- To what degree the performance of Amharic DIR system is increased by integrating page segmentation technique?
- What are the issues that need further researches in Amharic DIR system?

1.3. Objectives of the Study

1.3.1. General Objective

The general objective of this research is to integrate effective page segmentation technique that identify text regions from non-text, thereby enabling Amharic DIR system work in real-life document images.

1.3.2. Specific Objectives

To meet the general objective of the present study, the following specific objectives are set.

- To review previous international and local research articles, books and the Internet on DIR and page segmentation to understand the area (the state of the art) and what have been done so far on local languages;
- To identify the special features of Amharic words and word separators in real-life document images;
- To explore and select potential page segmentation techniques for Amharic document images;
- To prepare a document image corpus, queries and relevance judgment to measure the performance of the proposed system;
- To adopt page segmentation technique(s) and integrate it with the previous Amharic DIR system;
- To evaluate the performance of the system and report the findings with future research directions.

1.4. Scope and Limitation of the Study

Intending to improve the performance of Amharic DIR system developed so far by previous researches [5][10][11][12], this research explored and integrated effective page segmentation technique to the existing system. Modules for feature extraction, noise removal, indexing, searching and ranking were directly adopted from previous works.

The performance of the proposed page segmentation technique is measured in document images which contain pictures, graphics, and tables. The technique were also tested on typewritten documents, "Kum tsihuf" (handwritten Amharic documents) and different noisy documents. In addition to that, to see the effect of the proposed technique in Amharic DIR system and compare the performance with the previous works documents used by Biniam[10] were used.

One of the limitations of this study is that text feature recovery techniques can't be explored and integrated to enhance segmentation performance in noisy and degraded document images due to time limitation. The absence of standard corpus to measure the performance of the system is another limitation of the study. Limited number and size of documents made finding synonym words difficult for query expansion.

1.5. Methodology of the Study

The following methods have been used in order to achieve the objectives and answer the research questions of the study.

1.5.1. Literature Review

Books, journal articles, conference proceedings and the Internet about DIR in general and page segmentation in particular have been intensively reviewed in order to acquire detailed understanding of the subject matter and the research areas. Since this research was supposed to be a continuation of the previous researches and need to be integrated with them, local researches on Amharic DIR have been given more emphasis. Features

of Amharic words and word separators from scanned documents have also been studied to help selecting appropriate segmentation technique(s) for the language.

1.5.2. Dataset Collection

Documents that contain images, graphics, and tables; typewritten documents and handwritten documents ("Kum tsihuf") have been collected from different sources to test the performance of the proposed technique. Since there is no standard document image corpus and queries available to compare the performance of Amharic DIR system before and after integration of the proposed technique, document image corpus and queries used in the previous work[10] have been used. to measure the performance of the proposed system and identify its contribution to Amharic DIR. Documents containing images, figures, logos, documents containing tables and lines, typewritten documents, and handwritten documents ('kum tsihuf') are also included in the dataset.

1.5.3. Implementation Tool

Java programming language was used by previous researchers, Mesfin [5], Abreham [12] and Adane [11] for preprocessing, segmentation, feature extraction, indexing, query processing, similarity measurement, retrieval and ranking. Biniam [10] used MATLAB and Java together and integrate his work with previous works. Hence, MATLAB and Java programming language are used in this research to easily integrate the current work with the previous ones. The other reason to use MATLAB and Java is both languages have a lot of advanced built-in methods for image processing. And, the researcher is also familiar with both languages.

1.5.4. Testing Procedure

First, the effectiveness of the proposed segmentation technique was measured using Global Consistency Error (GCE) and Match Score. Then, the performance of the Amharic DIR system before and after integrating the proposed technique was measured using performance measurement techniques: precision, recall and F-measure. Precision is the fraction of retrieved documents that are relevant, recall is the fraction of relevant documents that are retrieved and F-measure is the weighted harmonic mean of

precision and recall that trades off precision versus recall. Five queries are used for each document image types in the experiment.

1.6. Significance of the Research

Large amount of documents articulated and printed in Amharic scripts are available in information centers, libraries, museums and government and private institutes [13].There is bulk of real-life and historical printed, typewritten, handwritten and special handwritten documents available that need to be digitized and accessible via the Internet and digital libraries [9]. Manually accessing these documents is time consuming. It is also costly to copy and make the documents available for large number of people.

Taking into consideration the need to make these documents easily accessible, some researches have been done to come up with Amharic DIR system [5][10][11][12]. This research attempts to contribute to the effectiveness of Amharic DIR by exploring and integrating effective page segmentation technique with the existing DIR system.

Libraries, museums, churches and other information centers can benefit by using the system to reach a number of users easily. Governmental and non-governmental institutions can also automate their office using DIR system to make information on printed documents easily accessible. Researchers on different disciplines such as: history, religion and sociology, politics, etc. are other beneficiaries of the system. Therefore, developing DIR system has several benefits for different parties.

To solve the problem of accessing such documents and to make all the benefits mentioned above real, applicable Amharic DIR system must be delivered. In this regard, the result of this research contribute a lot to the attempt of developing applicable Amharic DIR system.

1.7. Organization of the Study

This document is organized into five chapters. The first chapter presents the background of the study, statement of the problem, general and specific objectives of the study, methodology of the study, scope and limitation of the study and significance of the research results.

The second chapter is a literature review on information retrieval, document image retrieval and page segmentation techniques. It also includes a review on the history and characteristics of the Amharic writing system and different type of documents written in Amharic language. Finally, related research works (global and local) on document image retrieval and page segmentation are presented.

In the third chapter, the algorithms and techniques used in the study are presented. Five page segmentation techniques are discussed briefly, including equations and narrations of the algorithms. The chapter also includes the performance measurement techniques and formulas for the proposed page segmentation technique and DIR system.

The fourth chapter presents implementations of the proposed segmentation techniques and different experimentations using the proposed techniques. The chapter also addresses integration and evaluation of the performance of the system. The chapter finally addresses findings and challenges during the experiment.

The last chapter (chapter 5) presents conclusions based on the findings of the study and forward recommendations for further research works in the area of DIR.

CHAPTER TWO

LITERATURE REVIEW

The advancement and accessibility of digitization tools and repositories such as digital libraries and the Internet resulted in massive information explosion all over the world. Most information is still recorded, stored, and distributed in paper format; the widespread use of computers, with the introduction of personal computers and word-processors in the late 1980's had the effect of increasing, instead of reducing, the amount of information held on paper [2]. Thus, the need arises to digitize, store, query, search and retrieve different documents to make the information accessible for public use efficiently and accurately [3].

2.1. Information Retrieval (IR)

At the present time, IR is the basic technology behind search engines and an everyday technology for many Web users [14]. IR deals with representation, storage, organization of and access to information items [4]. According to Mandel [14], information retrieval is the key technology for knowledge management which guarantees access to large corpora of unstructured data. It helps user to find specific document(s) containing the information they are looking for from large document collections.

These days, we have large amount of documents in electronic formats (textual documents). And many IR tools have been developed for retrieving information from these textual documents [10]. Very often, text collections need to be processed indexed and searched by retrieval systems. Besides, information retrieval has great contribution towards Web search engines and an everyday duty for many Web users [14].

There are also lots of handwritten, typewritten and printed documents in the real world. These documents are scanned for archiving or in an attempt to move towards a

paperless office and are stored as images (Document Images). The economic feasibility of creating a large database of document image brought a tremendous need for robust ways to access the information [15]. Hence, image based document retrieval systems (DIR systems) appeared to introduce document images to retrieval world.

2.2. Document Image Retrieval (DIR)

Due to maturity of database technologies compared to image understanding technologies, in early image databases, images were often manually processed and analyzed to take advantage of automatic database organization, storage, and retrieval capabilities [16]. Manual processing usually involved associating a set of keyword descriptors with each image. According to Shin and Doermann [16], manual indexing can be much expensive for large databases. In addition to this, the subjective and possibly myopic interpretation by the person creating the index, and the limited expressiveness of keywords, are also challenges in manual indexing. Consequently, the problem of automated processing and retrieval of images by content has evolved as an active area of research.

According to Manesh & Shirdhonkar [15], DIR is a very attractive field of research with the continuous growth of interest and increasing security requirements for the development of the modern society. However, complex documents present a great challenge to the field of DIR. Presence of noise, handwriting, signature ,logos, machine-print with different fonts and rule lines impose a lot of restrictions to algorithms that work relatively well on simple documents.

Mohammadreza & Reza [17] classified document image indexing and retrieval approaches in to two: “Traditional indexing” and “Today indexing”. Figure 2.1 shows the classification of document image indexing approaches.

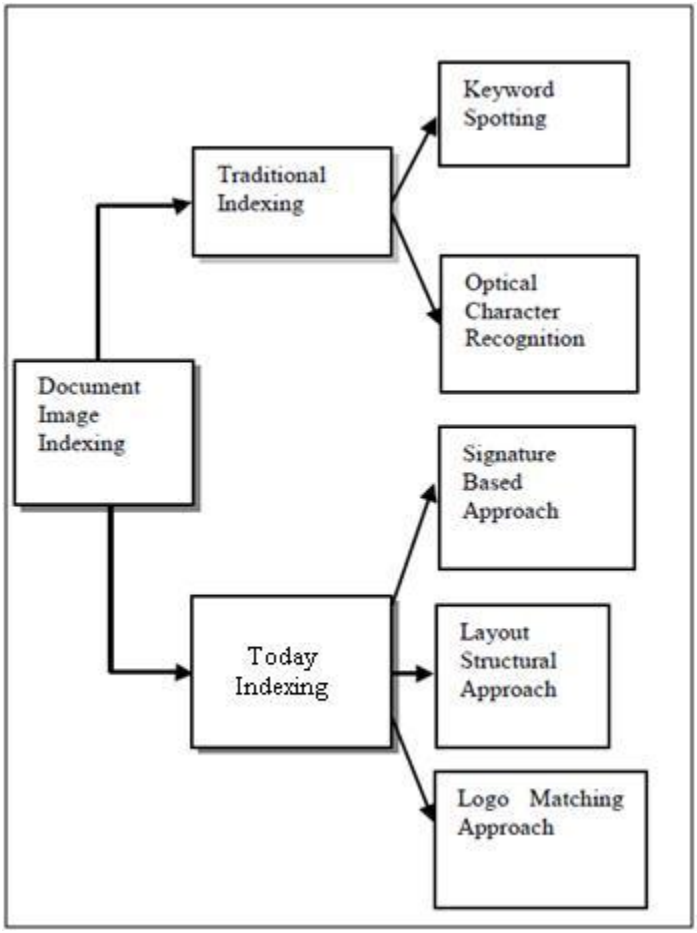


Figure 2.1 - Document Image Indexing Approaches [17]

There are two types of DIR systems in "traditional indexing" approach: recognition based systems which implement the use of OCR for recognition and recognition free (document image) retrieval systems. And there are three approaches in "today indexing" or "new method indexing"; layout structure based, signature based and logo based DIR [17].

2.2.1. Recognition Based Document Image Indexing and Retrieval

Recognition based approach uses optical character recognition (OCR) tool to transform characters, which were contained in the digitized printed documents (document images) into a machine-editable text (e.g. ASCII or Unicode representation format) to apply text based indexing and retrieval [7][18][19]. Retrieval of text documents that are similar with respect to content has been addressed by researchers in information

retrieval for many years; however, the techniques are highly dependent on the quality of the OCR [16].

Different researchers attempted to develop a robust OCR tool for printed and handwritten Amharic documents for more than a decade [13][20][21]. However, developing robust OCR for Amharic script is still challenging and long-term process.

2.2.2. DIR Based on Keyword Spotting

DIR based on keyword spotting is DIR without explicit recognition. It converts users query to image query by the process called query rendering and retrieval will be based on image similarity. Many interesting works have been done on the area of keywords extraction in document images [22][23][24][25]. This approach is composed of two operations: online and offline operations [26].

The offline operation involves three basic steps; preprocessing, word spotting and feature extraction. Finally, feature of each word will be saved in a database [26]. While the online procedure consists of the interface for end user, word image constructor, preprocessor and feature extractor. Similarity measurement between query features with indexed word features in the database and displaying results are also part of the online procedure.

2.2.3. DIR Based on Layout Structural Similarity

Unlike other techniques for searching the text within a document, searching documents according to their layout structure is based on the appearance and not the textual content found in a document [16]. The layout of a particular document often contains a significant amount of information that can be used to identify a document stored in a large database. Shin and Doermann [16] described DIR based on layout structure as follows:

> "The general premise for searching documents based on their layout structure is that the layout structure of a document often reflects its type. For example, business letters are in many ways more visually

> similar to one another than they are to magazine articles. Thus, a user searching for a particular document while knowing the class of documents is able to more effectively narrow the group of documents being searched (pages 608-609)".

2.2.4. Signature Based DIR

In searching complex documents, such as a repository of archival office documents, relevant documents (documents signed by the same author) are retrieved based on relating the signature in a given document to the closest matches within a database of documents; this is known as the signature retrieval task. Having a database of signed documents, there would be a need to relate a document to other documents in this database which have been signed by the same author [27].

2.2.5. DIR Based On Logo Matching

Most business and government documents appeared to seal logos which are used as a declaration of document source and ownership. Considering DIR, logos provide an important form of indexing that enables effective exploration of data. Searching for a specific logo is a highly effective way of retrieving documents from the associated organization from a large collection of documents [15].

2.3. Steps in DIR

Although it is described by different scholars differently [3][10][11][28], preprocessing, segmentation, feature extraction, indexing, matching and displaying relevant document images in ranked order are the major steps in recognition free DIR. Figure 2.2 below shows the overall structure of recognition free DIR system.

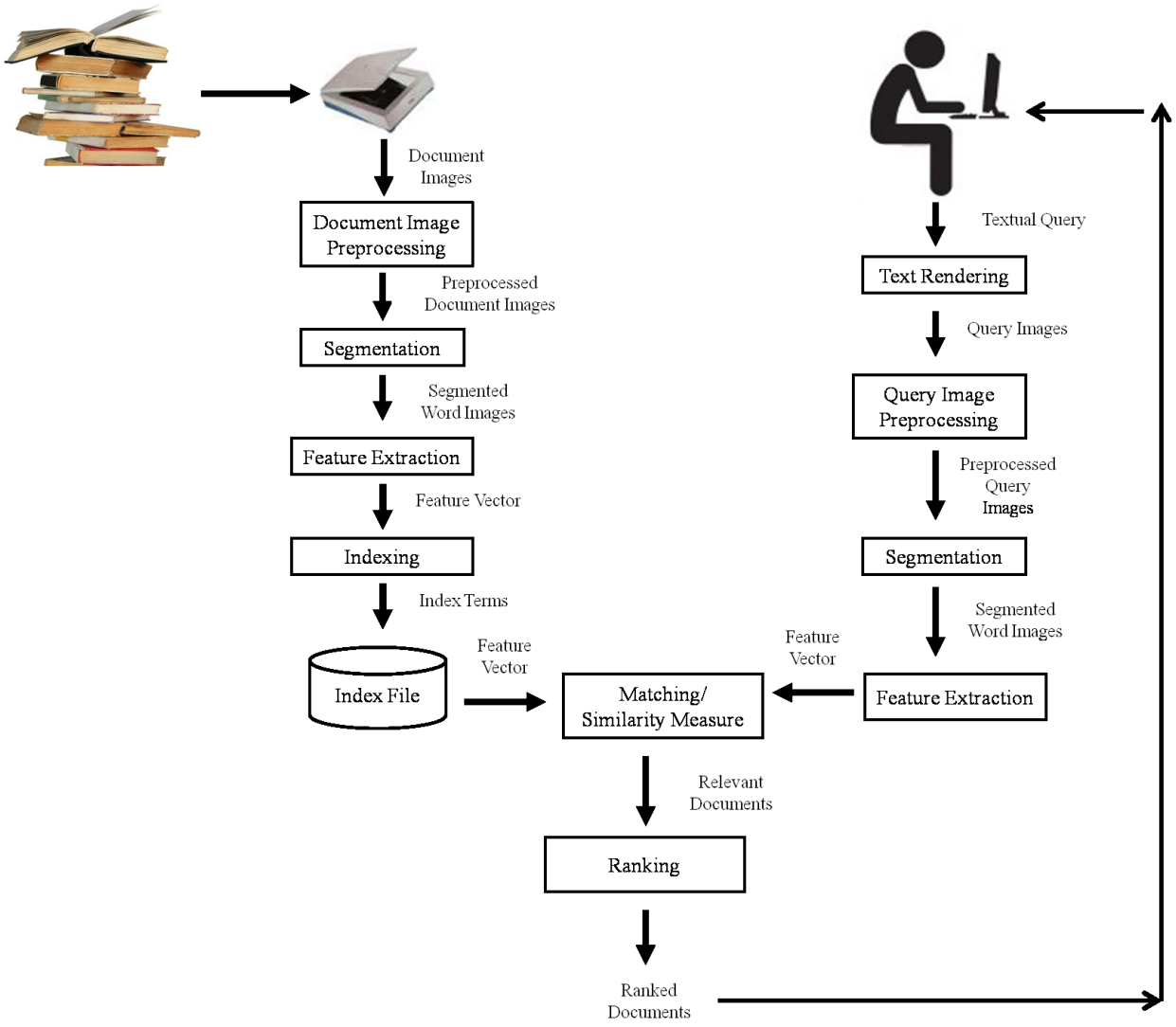


Figure 2.2 – The Overall Structure of Recognition Free DIR System [10]

Preprocessing is the first step in document image processing [3][28]. Major preprocessing tasks while working with document images are noise reduction, binarization or thresholding and skew correction [10]. Paper documents in real world are highly exposed to noises. In addition to that, the images collected by different type of sensors are generally contaminated by different types of noises [29]. Therefore, before manipulating the information in the scanned images, preprocessing tasks must be conducted. The purpose of preprocessing in DIR is to improve the quality of the document image being processed; it makes the succeeding steps in DIR easier [30].

Noise is anything that is irrelevant with the textual information [10]. Karthikeya [31] connects noise with the devices used to capture images. According to Karthikeya [31] noise is the random variation of brightness or color information in images produced by the sensor and circuitry of a scanner or digital camera. Removing any type of noise from document images is one of the preprocessing tasks. There are different image filtering techniques available for this purpose, the known one is low-pass filtering techniques like: mean filter, median filter, adaptive median filter, etc [10].

In document images, the groups of pixels representing objects of interest are called foreground pixels and the rest are called background pixels [32]. Binarization (thresholding) techniques helps to automatically choose a threshold that separates the foreground region with a single intensity (ON) and background region with a different intensity (OFF) [11].

Wu and Manmatha [33] categorize thresholding techniques into two groups: global and adaptive. Global technique binarizes the entire image using a single threshold value selected using the values at the valley of the intensity histogram of the image, by assuming two peaks one corresponding to the foreground and the other for the background. Adaptive algorithm, on the other hand computes a threshold for each pixel based on information extracted from its neighborhood; different threshold values are used based on the difference in intensity of image regions.

After preprocessing, the image is segmented to separate the set of words in a document. Segmentation is the process of extracting from the image domain one or more connected regions satisfying uniformity [35]. The detail is discussed in section 2.4.

Feature extraction in DIR involves extracting the meaningful information that represents the document image. Features are extracted from the segmented document images at word or character level. Feature extraction reduces the storage requirement. As a result, the system becomes faster and effective in retrieving information [9][15].

Generally, features can be classified as: General features and Domain specific features [36]. General features are application independent features, and according to the abstraction level they can be classified as: pixel-level features calculated at the position of each pixel, local features calculated over the results of subdivision of the image band on image segmentation and global features calculated over the entire image. In contrast, domain specific features are application dependant features [11].

Features extracted from segmented image are indexed for ease of searching [12]. It is essential to develop an index to speed up searching from a document collection. Index is particularly important for efficiently use computer storage devices such as disks. Although disks permit rapid access to consecutive records, access to particular region of the disk is slow. By locating the particular regions that contain the documents which hold the record of interest for a given query, an index file helps to efficiently use disks. Records in index file can be scanned using different search algorithms [34].

In DIR the task of matching is to compare query features with the indexed features of the word images that present in the database of documents [19]. Matching in document images can identify the word images of the documents that are more similar to the query words through the extracted feature vectors [15]. Similarity is defined as a mapping function between the feature vectors that represent the content of images; a positive real value which is chosen to quantify the degree of resemblance between the compared images [37].

Once the relevant documents for users query are identified by matching algorithms, the results will be displayed in ranked order such that most relevant documents are listed at the top of the list. Ranking is used to sort the retrieved documents according to their degree of relevance [5][10][11][12].

2.4. Document Image Segmentation

Segmentation occurs at two levels; on the first level, blocks of text, graphics and other parts are separated. On the second level, text lines and words in the text image are

located [9]. Text/graphic segmentation and extraction of words and characters in document images are very useful for applications such as information retrieval using word spotting or optical character recognition [38]. The degraded quality of documents poses different problems such as characters broken into multiple components, text at the back of document images appearing on the front, etc., thus making extraction of the words and characters very difficult [39][40].

2.4.1. Text/Graphic Segmentation

Page segmentation into text and non-text components is an essential preprocessing step before OCR operation and word spotting. In document images, basic shapes of text characters are limited in number, but shapes of the non-text components including drawing, logos, graphs, etc. are unlimited. Therefore, OCR engines and word spotting algorithms treat both text and non-text components differently, such that they only recognize text components and then arrange recognized text and images of non-text components in an output document using layout information.

Figure 2.3 shows the composition of a given real world document that contains printed text, table, figure and noise. Given such composite documents, text/graphics segmentation techniques help identify text part(s) which is essential for word spotting and character recognition in DIR systems.

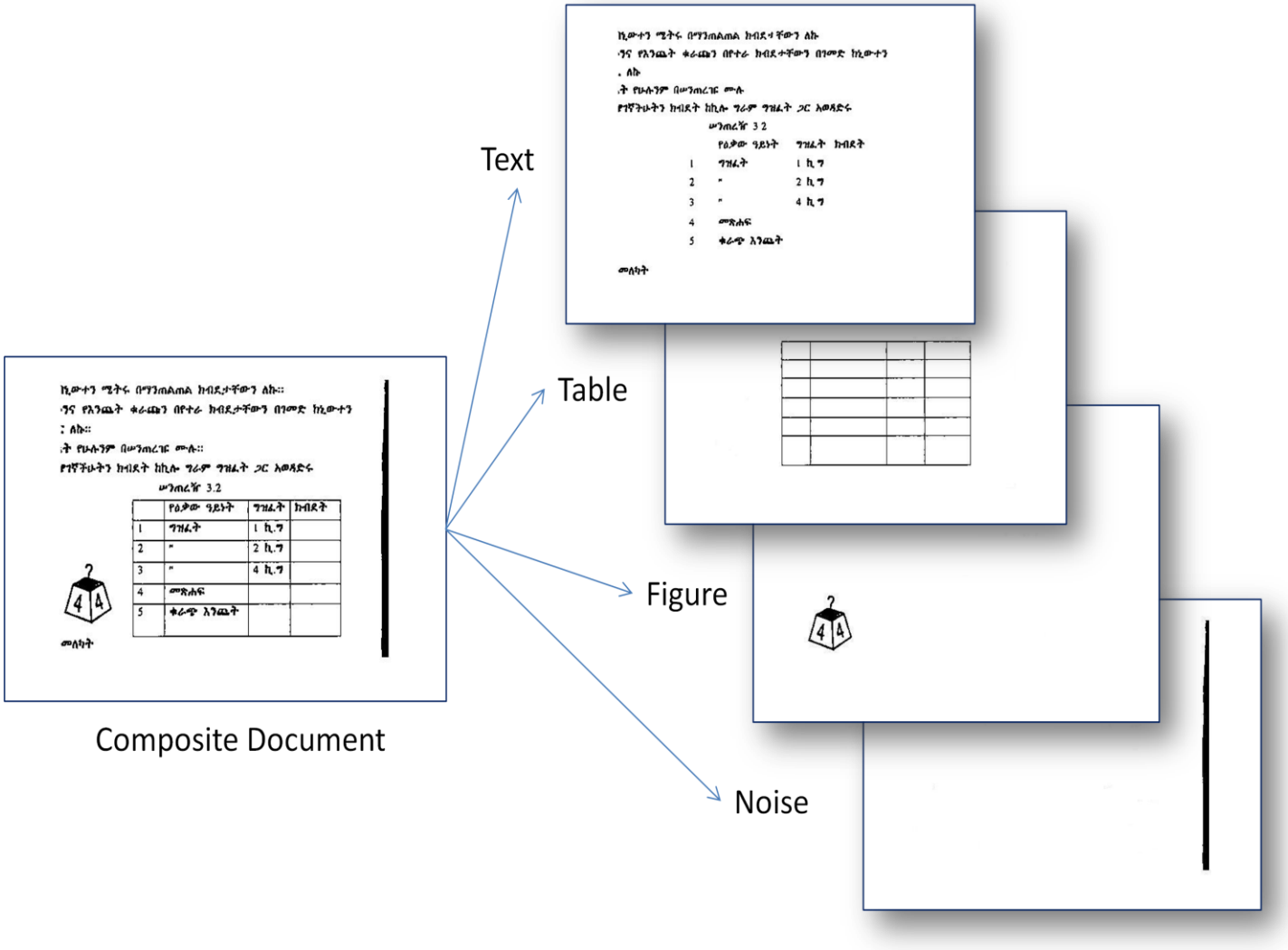


Figure 2.3 – A Composite Document and Its Parts

2.4.2. Text Line and Word Segmentation

According to Million [9] after text part is separated from graphics, tables and other parts, the next step is extracting text lines and words in the image. Line segmentation is a process of scanning horizontally text part of an image document for identifying parts that hold text line and parts that are blank [5].

After lines that contain text are identified, the next step is word identification (segmentation). Identification of word boundaries requires the task of distinguishing words from word spaces. The presence of spaces that precede or succeed a character makes it difficult to identify a word separator from a character separator. Hence, it complicates the job of word boundary identification as one of the challenges in word level segmentation [41].

2.5. Document Image Segmentation Techniques

Different authors catagorize image segmentation techniques differently. Skarbek, et al [76], catagorize image segmentation techniques as: pixel based, edge based, area based and physics based techniques. Yatharth [77], grouped segmentation techniques into: threshold techniques, edge based methods, region based methods and connectivity-preserving relaxation-based methods. Cattoni et al [78], provide approaches for page segmentation as: Smeering based techniques, projection profile methods, texture based (local analysis) techniques and sturcture based techniques.

Traditionally, page segmentation methods are divided into three main groups: top-down, bottom-up and hybrid approaches [42][43][44][45][46] cited by Khurram [38].

2.5.1. Top-Down Techniques

Top-down techniques divide document images recursively from entire image to smaller regions [38]. According to Khurram [38], the most well known top-down methods are projection methods, histogram analysis, and space transforms (Fourier transform, Hough transform, etc.).

Khurram [38] noted that for top down segmentation methods to be effective, they need to have a prior knowledge about the document class and type (number of columns, width of margins).

The widely used top-down technique for DIR is projection profile. X-Y cut algorithm also called recursive x-y cuts (RXYC) algorithm is a top-down page segmentation algorithm which implement projection profile histogram technique [47]

X-Y Cut Algorithm: it follows a tree-based approach; the root of the tree represents the entire document page and all the leaf nodes together represent the final segmentation. The RXYC algorithm recursively divides the document into two or more smaller rectangular zones which represent the nodes of the tree. The horizontal and vertical projection profiles of each node are computed at each step of the recursion. Noise removal thresholds t^n_x and t^n_y are used to compute the valleys in the projection profile histograms. First the thresholds t^n_x and t^n_y are scaled linearly based on the current zone's width and height. Then, all bins of the histograms that contain values less than the scaled thresholds are set to zero. The valleys along the horizontal and vertical directions, v_x and v_y, are then compared to the corresponding predetermined thresholds t_x and t_y; if the valley is larger than the threshold, the node is split at the mid-point of the wider of v_x and v_y into two child nodes. The process continues until no leaf node can be split further [47].

2.5.2. Bottom-Up Techniques

Bottom-up techniques start with the smallest elements (pixels), merge them recursively in connected components or regions, and then in larger structures [38]. They make use of methods like connected component (CC) analysis, region-growing methods, run-length smoothing, neural networks and active contours [38][48][49][50][51][52].

Figure 2.4 shows the result of extraction of figures and the caption line candidates for the extracted figures using the rule-based CC position and area analysis approach in [49]. It uses Area Height analysis to identify figures (big components) and extract near connected components as figure caption candidates.

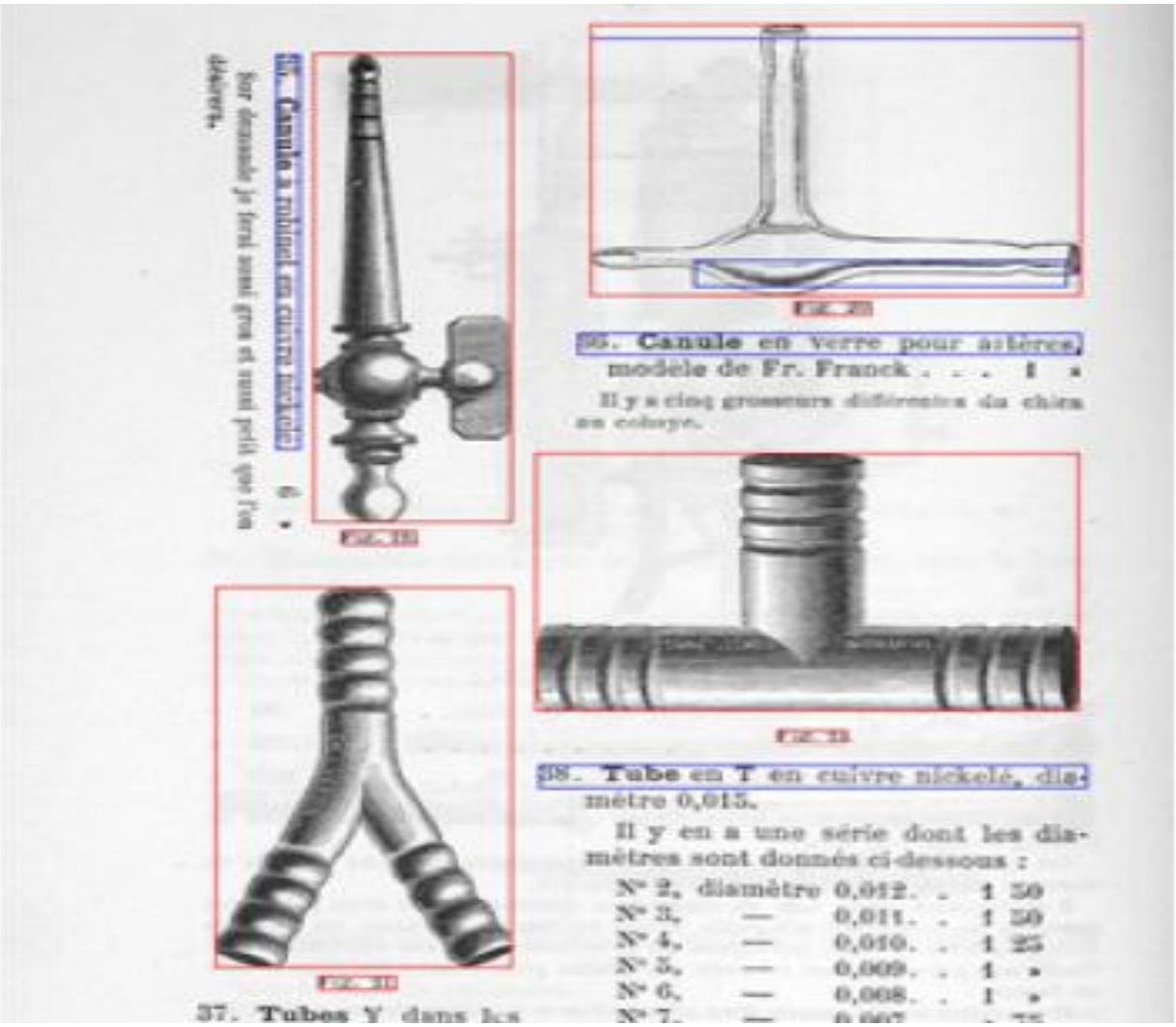


Figure 2.4 - Segmentation of Figures and Figure Caption Candidates by CC Analysis [49]

Shi and Govindaraju [46] stated flexibility of bottom-up techniques as an advantage. Khurram [38] on the other hand, argue, although these methods are efficient for modern and contemporary books, they are not so efficient for ancient medieval books because of the specificity of these ancient documents such as non-constant spacing between characters, words and images, etc.

Docstrum: docstrum algorithm is one of the bottom-up algorithms proposed by O'Gorman [53]. This approach is based on nearest-neighborhood clustering of connected components extracted from the document image. O'Gorman [53] proposed,

after noise removal, the connected components are separated into two groups, one with characters of the dominant font size and another one with characters in titles and section headings, using a character size ratio factor. Then, K nearest neighbors are found for each connected component. A histogram of the distance and angle of each connected component from its K nearest neighbors is computed. The peak of the angle histogram gives the dominant skew in the document image. This skew estimate is used to compute within-line nearest neighbor pairs. Then, text-lines are found by computing the transitive closure on within-line nearest neighbor pairings using a threshold. Finally, text-lines are merged to form text blocks using a parallel distance threshold and a perpendicular distance threshold.

Voronoi-Diagram Based Algorithm: as prposed by Kise et al. [54], Voronoi-diagram based algorithm first extracts sample points from the boundaries of the connected components using a sampling rate. Then, noise removal is done using a maximum noise zone size threshold, in addition to width, height, and aspect ratio thresholds. After that a Voronoi diagram is generated using sample points obtained from the borders of the connected components. The Voronoi edges that pass through a connected component are deleted to obtain an area Voronoi diagram. Finally, superfluous Voronoi edges are deleted to obtain boundaries of document components. An edge is declared superfluous if it satisfies any of the following criterion [54]:

- The minimum distance d between its associated connected components is less than the inter-character gap in body text regions.
- The minimum distance d between its associated connected components is less than the inter-line spacing times a margin control factor fm, or the area ratio of the two connected components is above an area ratio threshold.
- At least one of its terminals is neither shared by another Voronoi edge nor lies on the edge of the document image.

The output of the algorithm consists of arbitrarily shaped regions bounded by Voronoi edges; each Voronoi region is represented by its bounding box [54].

2.5.3. Hybrid Techniques

There are also some hybrid methods that combine and make use of both bottom-up and top-down approaches [38]. For example, connected component analysis for shape information and block separation for background block map [56] cited by Khurram [38].

Hybrid methods work very well for major text/graphic segmentation in both historical and contemporary pages (magazines, news papers, journals, etc.), but not for a very fine level segmentation of words and their individual characters in historical books [38].

2.5.4. Watershed Based Image Segmentation

Watershed transformation is a powerful mathematical morphological tool for the image segmentation [69]. The technique is more popular in the fields like biomedical and medical image processing, and computer vision [79]. In geography, watershed means the edge that divides areas drained by different river systems. If image is viewed as geological landscape, the watershed lines determine boundaries which divide image regions [69]. The watershed transform computes regions and edge lines (also known as watershed lines) [80]. Watershed Segmentation adopts the concepts from the techniques such as threshold based, edge based and region based segmentation [69].

There are mainly two classes of watershed algorithms: the flooding based watershed algorithms and rainfalling based watershed algorithms [69].

Flooding Based Watershed Algorithms

In traditional flooding based approach of watershed based image segmentation, image is considered as a topographic surface which contains three different types of points: points which indicate regional minimum, points where the water falling has highest probability to fall into a single minimum region and points where the water falling has probability to fall into more than one such a minimum region. For regional minimum, the groups of points satisfy second condition called watershed or catchment basin of that minimum and the groups of point satisfy third condition makes a crest line on

topographic surface termed as a watershed line. Figure 2.5 shows an example of the watershed line and catchment basin [69].

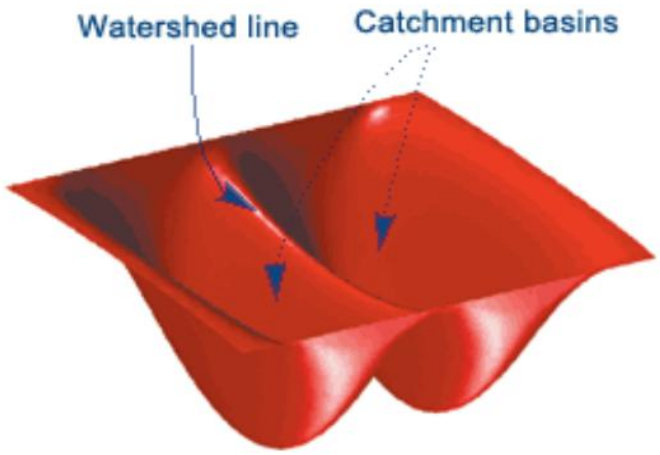


Figure 2.5 - Watershed Lines and Catchment Basins [69]

The basic concept of watershed algorithm used for the image segmentation is to find the watershed lines (boundaries). Imagine, holes at each regional minimum, and water is flooded into these holes with constant rate. The level of the water will rise in the topographic surface uniformly. When the rising water in different catchment basins is going to merge each other, then a dam is built to prevent merging of the water. Finally, flooding of water will reach at the point where only top of the dams are visible. These continuous dam boundaries are called the watershed lines [69].

Rainfalling Watershed Algorithms

Unlike traditional flooding based algorithm, the rainfalling algorithm extract mountain boundaries. The concept of the algorithm is that rain water drops fall on the mountain (topographic surface) and move to descending direction because of the gravity until they reach to the local minimum surface. The algorithm tracks the path of water drop for each point on the surface towards the local minimum, if rain drops pass through that point or fall on that point. A group of points make a segment when water drops related to them flow downwards to the same deepest location. When a point has more

than one path towards the different steepest surfaces then it can be allocated to any one of the local minimum [69].

Page segmentation algorithms are used to extract words in DIR with or without explicit recognition systems. Different researchers [9][26][29][38][46][47] implemented the algorithms discussed above for DIR systems. To come up with effective DIR system for Amharic language, implementing effective page segmentation technique which can work on all type of real-life documents is vital.

Before implementing page segmentation technique to Amharic DIR system we have to know special features of the Amharic script such as word delimiters, font types, types of documents, etc. The next sections present the Amharic scripts in detail.

2.6. The Amharic Writing System

Language is the human capacity for acquiring and using complex systems of communication, and a language is any specific example of such a system. Any estimate of the precise number of languages in the world depends on a partly arbitrary distinction between languages. However, estimates vary between 6,000 and 7,000 languages in number [62].

There are 70 or more languages spoken in Ethiopia, most belong to the Semitic and Cushitic branches of the Afro-Asiatic family [12]. Ge'ez, the language of the Ethiopian church, gave rise to the Semitic cluster of languages [58]: Amharic, Tigrinya, and Tigre. Amharic is the country's official language; it is spoken by more than half of the population. It is one of the languages which have their own writing system. The language's writing system was adapted from the Ancient Ge'ez writing system [58].

The earliest known inscriptions in the Ge'ez script date back to the 5th century BC [5]. The Ethiopic writing system has its origins in the same ancestral writing systems with those of European alphabets, namely the Semitic scripts that proliferated in the Middle East more than three thousand years ago [59].

Amharic script, which is a successor of Ge'ez and dates back to 300 AD, is used for writing in Ethiopia and Eritrea, for languages like Amharic, Tigre and Tigrigna [5]. Amharic writing system took all the symbols from Ge'ez writing system and adds some new symbols to the writing system. Ge'ez is still used as a language in liturgy of the Ethiopian orthodox and in church literatures [60].

Ge'ez language belongs to the class of sematic language, which was derived from the south Arabian alphabet called sabaean. Amharic is grouped in Afroasiatic family. The complete path of the language is:

Afroasiatic → Semantic → East Semantic → Ethio-Semantic → South → Amharic

Figure 2.6 depicts the genetic structure of the Amharic language [60].

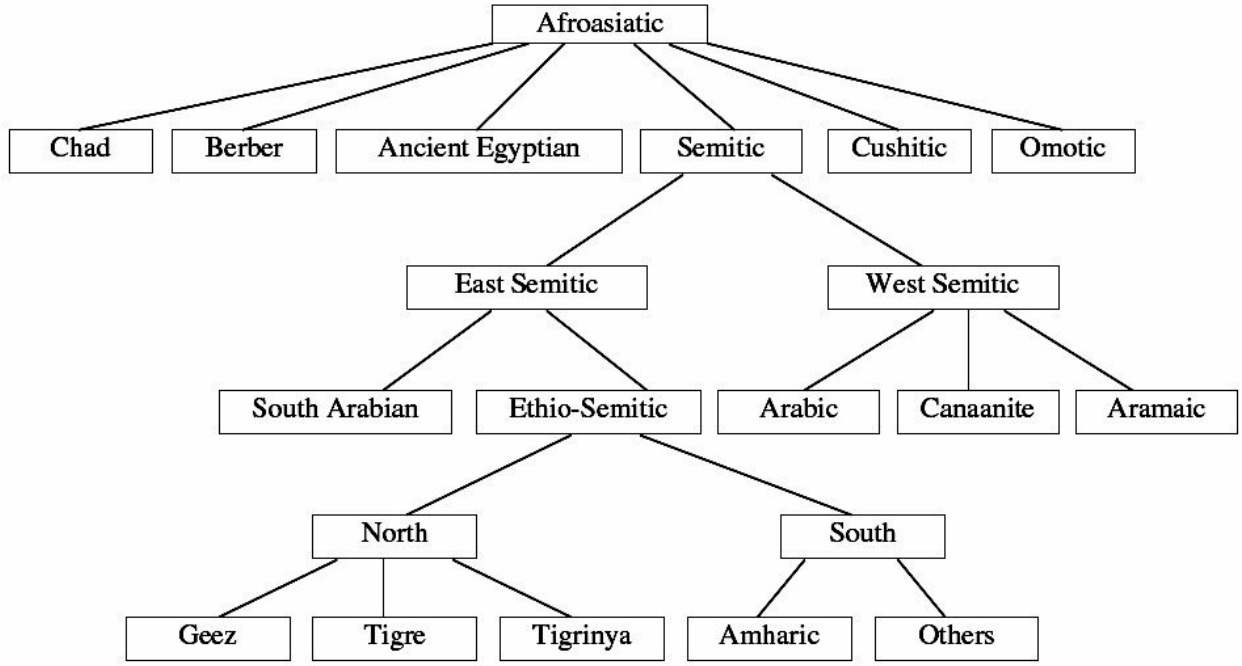


Figure 2.6 - The Genetic Structure of Amharic Language [12]

2.6.1. Amharic Characters

Amharic writing system has of thirty three core characters. The thirty three characters occur in one basic form and in six other forms know as orders, as shown in table 2.1. These orders are derived from the basic forms by more or less regular modification [63]. The seven orders of the Ethiopic represent the different sounds of a consonant- vowel combination known as syllabic.

1st order	2nd order	3rd order	4th order	5th order	6th order	7th order
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
Hä	hu	Hi	ha	He	h	Ho
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
Lä	lu	Li	la	Le	l	Lo

Table 2.1 - The Seven Orders of Amharic Writing System [12]

As shown in Table 2.2, besides the basic characters, there are series of derived characters to represent labialized velar consonants [12]. For example, these are velar sounds like /k/, /g/, /q/, and /h/ that are pronounced with the lips rounded regardless of the vowel.

	a	i:	a:	e:	(ə)		a	i:	a:	e:	(ə)
q^w	ቋ	ቍ	ቋ	ቌ	ቍ	k^w	ኰ	ኲ	ኳ	ኴ	ኵ
h^w	ኈ	ኈ	ኋ	ኌ	ኈ	g^w	ጐ	ጕ	ጓ	ጔ	ጕ

Table 2.2 - Characters Representing Labialized Velar Consonants [12]

2.6.2. Amharic Numeration System

As shown in table 2.3 below Amharic numeration system consists of basic single symbols for one to ten, for multiple of ten (twenty to ninety), hundred and thousand [60]. These numerals are derived from the Greek numerals with some modifications. Each symbol has a horizontal stroke below and above. There is no symbol for representing zero value in Amharic number system, and it is not a place value system, thus arithmetic computation using this system is very difficult. As a result, in most printed document Hindu- Arabic numerals are used [60].

Ethiopic	Arabic	Ethiopic	Arabic	Ethiopic	Arabic
፩	1	፰	8	፷	60
፪	2	፱	9	፸	70
፫	3	፲	10	፹	80
፬	4	፳	20	፺	90
፭	5	፴	30	፻	100
፮	6	፵	40	፼	1000
፯	7	፶	50		

Table 2.3 - Ethiopic Numerals [11]

2.6.3. Amharic Punctuation Marks

There are about 17 punctuation marks in Amharic writing system [12]. Some of commonly used punctuation marks are:

- **ሁለት ነጥብ** (፡) - word delimiters
- **አራት ነጥብ** (።) - sentence delimiters (the equivalent of the full stop)
- **ነጠላ ሠረዝ** (፣) - the equivalent for comma
- **ድርብ ሠረዝ** (፤) - the equivalent of semi-colon

Word delimiter (**ሁለት ነጥብ**) is most commonly used in historic documents. However, nowadays, in computer writing style it is common to use a space as a word separator instead of using "፡" (**ሁለት ነጥብ**) [12].There are also some borrowed symbols; '?' (question mark), '!' (exclamation mark), arithmetic operators such as '+','-','*','/', brackets ('(',')'), quotation marks (", "), etc).

2.7. Documents Written in Amharic Script

Large amount of historical and recent documents written in Amharic language are available. These documents can be categorized into typewritten, printed and handwritten [10].

2.7.1. Printed Documents

There are a number of Amharic computer fonts available these days for Amharic writing system. 'Power Geez', 'Visual Geez', and 'Nyala' are some of the commonly used fonts in computer printed Amharic documents. An Amharic word written in different fonts is shown in Table 2.4 below.

Amharic Word	Font Type
ኢትዮጵያ	Geez-1
ኢትዮጵያ	Ge'ez -2
ኢትዮጵያ	Geez-3
ኢትዮጵያ	Visual Geez 2000 Main Font

Table 2.4 – Different Amharic Font Types

From the above example, we can see that words belonging to the same class but printed using different fonts greatly vary both in shape, width, height and line thickness.

2.7.2. Typewritten Documents

The first Amharic typewriter called Olivetti Lexicon 80 was made in 1950 [64]. Since then, a number of documents are produced in the form of books, magazines, correspondence letters, etc. As described by Dereje [64] height and width of the individual characters in a typewritten document vary greatly, but the space that is used to type a single character is proportional. As a results there exist connected characters in typewritten documents when two consecutive characters (especially the second, third and forth forms) take up all the space in between.

2.7.3. Handwritten Documents

Handwriting is the most dominant means of written communication till these days. It also brings difficulty to automation of handwritten documents [65]. In Ethiopia, handwriting is broadly used among the society, public institutions and public officials for many purposes. There is no clear rule that abandons cursive handwriting; however, people often write in a disconnected, but non-uniform manner [66].

2.8. Challenges of Amharic Script

As pointed out by different researches, the nature of Amharic writing system creates some challenges in developing Amharic retrieval system especially DIR systems [5][10][11][12][60]. The challenges in the Amharic writing system are discussed below.

2.8.1. Existance of Character Variants

One of the problems in Amharic writing system is the presence of character variants that are used interchangeably in writing. These characters include; (ሀ, ሐ, ኀ, and ኸ), (አ and ዐ), (ሰ and ሠ) and (ጸ and ፀ). For example to write the name of the famous athlete Haile Gebreselassie one can write it in twenty four different ways: **ሀይሌ ገብረስላሴ**, **ሐይሌ ገብረስላሴ**, **ኀይሌ ገብረስላሴ**, **ሃይሌ ገብረሥላሴ**, **ኃይሌ ገብረሥላሴ**, **ሐይሌ ገብረሥላሤ**, etc. As a result, designing retrieval system, especially DIR system for Amharic printed documents is somehow complex due to such variation [11]. If a search algorithm is implemented based on one of the representation; the others may be missed even if they represent the same object.

2.8.2. Feature Similarity Among Characters

There is also similarity in shape among different characters in Amharic writing system. Consider characters such as; (ደ and ጸ), (ደ and ጀ), (ሰ and ሸ), (ለ and ከ), etc [67]. Such similarities would be challenging for the retrieval systems in representing feature of words. For example, a system may group the word **ደመረ** and **ጀመረ** in the same cluster though they are different in their meaning. Because, the first letters of both words have similarities with few distinctions while the remaining two characters are similar. Thus, the two words are more similar than they are dissimilar in features as DIR systems compare features of word images. Therefore, such feature similarities among different characters affect the performance of Amharic DIR system.

2.8.3. Font Variations

There are different fonts produced for Amharic writing system, such as, Alpas, Brana, Agafari, Powergeez, Ge'ez, ALXethiopian, Visual Geez Unicode, Nyala, Addis98, etc. The existence of various fonts, in addition to different font sizes (10,12,14,16,…) and font styles: normal, *italic* and **bold** makes designing DIR system a challenging task.

The difference in font types, sizes and styles are resulted in different word lengths. For instance, as it is shown in Table 2.5 below, the word written in font type Ethiopia Jiret,

font size 12 and font style Normal is the shortest in length, whereas the word written in ALXethiopian with similar font size and style is the longest of all the others.

Font types	Font size 10	Font size 12	Font size 14	Font style Normal	Font style **Bold**	Font style *Italic*
Ethiopia Jiret	ምድርን	ምድርን	ምድርን	ምድርን	**ምድርን**	*ምድርን*
Nyala	ምድርን	ምድርን	ምድርን	ምድርን	**ምድርን**	*ምድርን*
GF Zemen Unicode	ምድርን	ምድርን	ምድርን	ምድርን	**ምድርን**	*ምድርን*
ALXethiopian	ምድርን	ምድርን	ምድርን	ምድርን	**ምድርን**	*ምድርን*

Table 2.5 – Font Type, Size and Style Variation [11]

Moreover, the word written in Ethiopia Jiret, is more condensed than others. Thus, variation in font types, sizes and styles have an impact on designing retrieval system for Amharic printed documents [11].

2.8.4. Formation of Compound Words

The Amharic writing system uses different ways to denote compound words. Compound words are sometimes written as a single word and other times as two separate words as there is no agreed upon spelling standard for constructing these words [60][68]. For example, the corresponding Amharic word for the English word "Church" can possibly be written as '**ቤተክርስትያን**' or '**ቤተ ክርስትያን**'. This variation of the writing system in writing compound words causes the same word to be indexed in different forms which affects the performance of Amharic DIR systems.

2.8.5. Rich Morphology

Amharic is a language with a very rich morphology; a word has many inflectional variations in the language. Thus, searching in Amharic text databases can be effective only if full account of the many word variants that may occur is taken in to account [9]. Towards grouping the many variants of Amharic words this richness in morphology creates challenge in stemming Amharic words.

2.9. Related Research Works

There are different attempts to deliver a robust DIR system with and without the implementation of OCR tool. OCR engines for African and Asian languages are not robust enough to be used for DIR systems. Moreover, the performance of OCR engines on ancient (historical) documents is in question. Therefore, developing recognition free DIR system becomes a hot research area. This section presents global and local researches done on the area of DIR without explicit recognition.

2.9.1. Global Research Works

Zagoris et al. [26] proposed a technique to addresses the document retrieval problem by a word matching procedure. The proposed technique performs matching directly in the images bypassing OCR and using word-images as queries. The system is designed to have two different parts: The offline and the online operation. In the offline operation, the archive of document images is processed and the results are stored in a database. The online operation basically consists of web interface, query rendering and, matching. The system can also be easily combined with page layout analysis techniques to develop a general document retrieval system.

The proposed method is based on word spotting by using a set of powerful features and a two-threshold rating compare and matching scheme. Specifically, it addresses the document retrieval problem by a word matching procedure. It avoids use of OCR by using only word images as queries. An experimental platform has been developed and implemented and in the web. As Zagoris et al. [26] reported, a number of test search results have shown a lot of potential. Specifically, the experiments showed high Recall and good Precision rates proportional to the rating thresholds predefined to the system.

Shi and Govindaraju [46] presented a novel approach for document page segmentation using a multi-scale technique. They implemented a local connectivity algorithm to transforms a document image into a parameter domain in which a parameter value at a pixel location represents a connectivity property for its neighboring foreground pixels in the original document image. Then, a top-down approach with a linear search is

implemented to segment document image regions at each scale levels as text block, text lines and graphics.

Shi and Govindaraju [46] stated their algorithm to be "a transform based multi-scale method" and the algorithm is robust for variations of document parameters. For their research especially emphasized on finding feasibility of their method, they have chosen images from several difference document image sources which include document pages scanned from IEEE journals, magazines and books. Images in three rough categories are scanned with 300 dpi resolution. The three categories of images are: images with simply aligned multiple columns such as IEEE journal pages, single column book pages and complex magazine pages with multiple columns and graphics embedded in text. They reported that the proposed method is a robust approach for segmentation of most printed documented images.

Khurram [38] explores some automatic techniques that would allow the retrieval of document images with both ASCII as well word image queries, and would help in automatic indexing/annotation of the document images. In [38] word spotting that finds all similar word images based on the query given to the system, has been thoroughly examined along with different character's string matching techniques.

The main contributions of the research as stated by Khurram [38] are: a detailed examination of retrieval approaches for historical documents, the development of a document image retrieval framework that allows text and image queries for information searching, and an automatic figure/caption pair indexing model by employing a fusion of symbolic and spatial information in the document image. Khurram [38] added, building such a system involves challenges on numerous levels: the noisy historical printed documents require adequate image pre-processing, binarization, segmentation and representation techniques, as well as a robust and scalable retrieval framework. The construction of a prototype system, which demonstrates the feasibility of the proposed techniques for a large collection of document images, is also described in [38].

2.9.2. Local Research Works

Summary of local researches on DIR without explicit recognition and future research directions as recommended by the researchers Mesfin [5], Abreham [12], Adane [11] and Biniam [10] are presented below.

The first attempt is made by Mesfin [5]. Mesfin [5] presented a DIR system that retrieve relevant documents based on users query. The document image is preprocessed, segmented at word level and the feature of each word is extracted. Then the textual query is rendered to convert into an image query, preprocessed, segmented and the feature is extracted. Thresholding segmentation is used to identify lines and words. The technique used for feature extraction considers the word shape analysis. The extracted feature of the image query is matched with the feature of the document images, at word level using cosine similarity measures. Finally relevant document images are retrieved in ranked order in response to user query.

Abreham [12] integrated inverted file indexing structure to improve the efficiency of Amharic DIR system. To this ends an inverted index file is created to store index terms after removing stop words and grouping together variant words. Prefix and suffix of word variants are detected by modifying cosine similarity measurement technique. The search result of the system is displayed in ranked order based on TF*IDF weight, and performance evaluation of the system shows a promising result. Abreham [12] recommended further researches to solve issues related to feature extraction, word variation detection and noise detection and removal.

Adane [11] incorporated feature extraction and matching techniques invariant to font types, size and style difference. According to Adane [11], eight feature extraction methods and four matching techniques are tested. Of the four matching schemes dynamic time warping is insensitive to font types, sizes and styles difference. The eight feature extraction techniques are tested for performance, and then each feature is combined systematically following best stepwise feature selection method. The result shows that combined features score better performance than individuals. Using the best

performer matching algorithm stemming is performed in image domain to handle word variants. Accordingly, promising experimental results are registered for word variants. The explored matching, feature extraction and stemming techniques are integrated with the previous Amharic DIR system and tested on noisy document images. As the experimentation shows, the performance of the current system outperforms the previous attempts.

Biniam [10] added noise reduction techniques to Amharic DIR. According to Binaim [10], combination of three noise reduction techniques: median, adaptive median and wiener filters, and three thresholding techniques: Otsu's, Niblack's and Sauvola's techniques are experimented in printed real-life documents plagued by low, medium, high and very high noise. Performance analysis shows that the best performing combination of denoising and thresholding techniques are wiener filtering and Otsu thresholding. Finally, the performance of the system is evaluated before and after the integration of the selected preprocessing techniques in which an average overall performance of 82.37% F-measure is registered in documents having low, medium, high and very high levels of noise.

Biniam [10] emphasized on word-level segmentation as future research direction. Because, the major challenge is segmentation error where the current segmentation algorithm either considers multiple words as one because of noise; on the other hands a single word as multiple words since the space between characters of a single word is increased when the noise is removed. So, the segmentation algorithm is unable to identify words in a noisy document since it uses predefined segmentation threshold value.

All the previous works [5][10][11][12] did not address text/graphics segmentation. Hence, in this study, page segmentation technique that works for text/graphics segmentation beside word-level segmentation is proposed. Therefore, this study focused on exploring and implementing page segmentation algorithms that is suitable for document images containing figures, images, tables, etc.

CHAPTER THREE

PAGE SEGMENTATION TECHNIQUES

The task of page segmentation is to divide the document image into homogeneous zones, each consisting of only one physical layout structure (text, tables, pictures, etc). Page segmentation techniques also help segment text part of the image to lines, words and characters. Both, recognition based and recognition free DIR systems use features of these segmented words and/or characters. Therefore, the performance of DIR systems highly depends on the page segmentation algorithm used.

The previous researches [5][10][11][12] used thresholding technique to segment lines and then words. This segmentation algorithm works for clean text document images with minimal noise level. But real-life documents are with various artifacts, they contain noise of different level. Documents may contain text and non text regions. This needs an effective segmentation technique that works well with any of Amharic documents.

3.1. Architecture of Amharic DIR System

The architecture of a typical DIR system is shown in Figure 3.1 below. There are indexing and searching subsystems in the DIR systems. The indexing subsystem, which is an offline process includes: noise removal, binarization, segmentation (both text/graphics and word-level), feature extraction and indexing. On the other hand, the searching subsystem, which is an online process includes: query segmentation, rendering, noise reduction, binarization, feature extraction, matching, ranking and displaying results to the user.

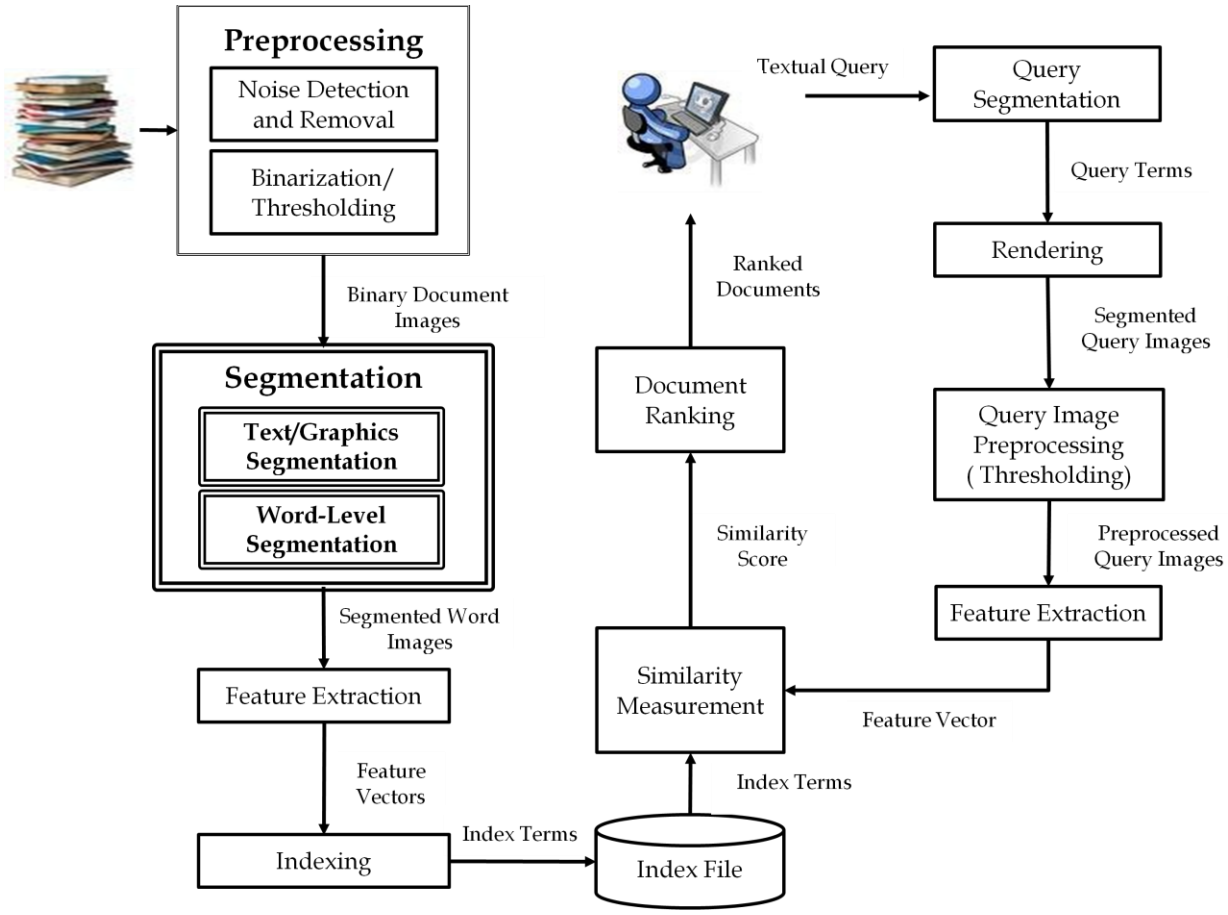


Figure 3.1 - Architecture of the Proposed Amharic Document Image Retrieval System
Rectangles in double line represent the focus of the present work.

This study focused on segmentation at two levels: text/graphics level and word-level. Real life documents include text, tables, pictures, etc. Therefore, the system segments text, graphics, pictures, etc and removes tables without removing its content as our focus is on text part of the document image. Words are segmented out of the text part because the system uses word features for indexing. As this study is a continuation of the previous works by Mesfin [5], Abreham[12], Adane [11] and Binaim [10] , it integrated different modules with the result of the study.

3.2. Page Segmentation Techniques

DIR without explicit recognition from real-life document images (consisting of text, images, logos, etc.) needs text/graphics and word and/or character segmentation as the

focus of such systems is on identifying words and/or characters that potentially represent the document images. For this purpose, this study explored five segmentation techniques namely: watershed transforms, run length smoothing, connected component labeling, whitespace analysis and constrained text-line detection. These techniques are experimented in different combinations on real-life Amharic document images.

3.2.1. Watershed Algorithm Based on Connected Components

Watershed algorithm based on connected components is one of the algorithms used to segment Amharic document images in this study. This algorithm gives the same segmentation results as the traditional watershed algorithm. At the same time, it has an advantage of lower complexity, simple data structure and short execution time. It connects each pixel to its lowest neighbor pixel and all pixels connected to same lowest neighbor pixel, make a segment [69].

As described in [69], the basic concept of connected components based watershed algorithm is shown in Figure 3.2. The original 6 x 6 image has three local minimum values indicated by gray boxes (3.2a). If a pixel is not a local minimum then it is connected to its lowest neighbors as shown by arrows in (3.2b), where m indicates a local minimum. All components directed towards the same local minimum make a segment and are given the same label value (3.2c).

7	4	8	12	11	3
7	7	8	12	11	7
13	13	15	16	16	13
19	19	18	17	15	7
20	18	17	16	15	5

(a) The original image

→	m	←	←	→	m
↗	↑	↖	←	↗	↑
↑	↑	↖	↖	↗	↑
↑	↑	↑	→	↘	↓
→	→	→	→	→	m

(b) Each pixel connect to lowest minimum

0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	2	2	2
2	2	2	2	2	2

(c) The image with labels

Figure 3.2 - Basic Concept of Connected Components Approach [69]

3.2.2. Run Length Smoothing

The run-length smoothing algorithm (RLSA) works on binary images where white pixels are represented by 0's and black pixels by 1's. The algorithm transforms a binary sequence x into y according to the following rule [50]:

1. if the number of adjacent 0's is less than or equal to a predefined threshold C, then change 0's in x to 1's in y.
2. 1's in x are unchanged in y.

For example, with C = 4 the sequence x is mapped into y as follows:

x : 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1
y : 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1

These steps link together neighboring black areas that are separated by less than C pixels. The RLSA is applied row-wise and column-wise to the document using thresholds, yielding two distinct bitmaps. These two bitmaps are combined in a logical AND operation. Then, connected component analysis is performed on this bitmap to obtain document zones. The mean horizontal run-length R_m of the black pixels in the original image, and the mean block height H_m are calculated. Then, a block is classified into a text block if [50]:

$$R < f_{tr}R_m \text{ and } H < f_{th}H_m$$

Equation 3.1 – Thresholding Equation RLSA.

where, f_{tr} and f_{th} are two thresholds, R is the horizontal run-length of the black pixels in the current block, and H is the block height.

3.2.3. Dilation

The dilation of an image F by a structuring element S is written as $F \oplus S$. To compute the dilation, we position S such that its origin is at pixel coordinates (x, y) and apply the rule:

$$g(x,y) = \begin{cases} 1 \; if \; s \; hits \; F \\ 0 \; otherwise \end{cases}$$

Equation 3.2 –Dilation Formula

Repeating this for all pixels coordinates, dilation creates a new image showing all the locations of a structuring element's origin at which that structuring element hits the input image.

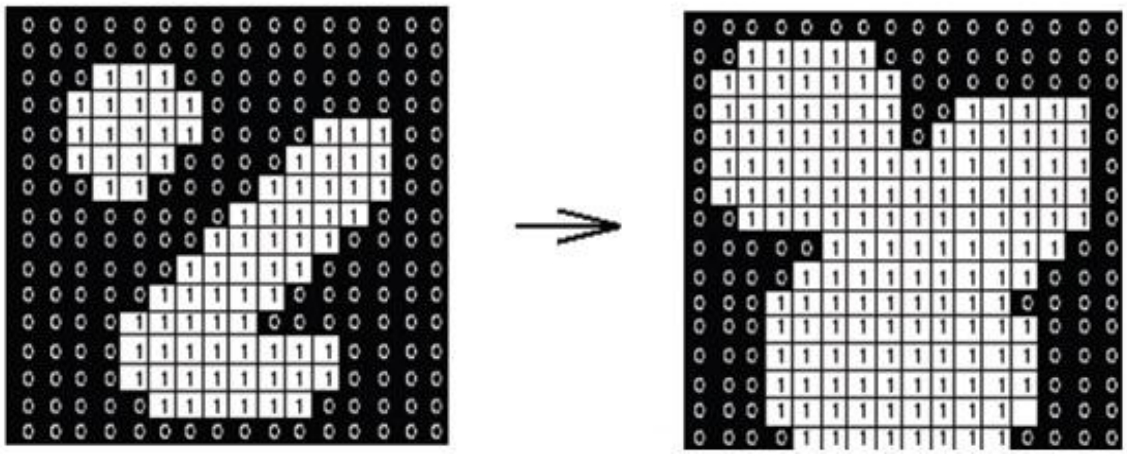


Figure 3.3 – Effect of Dilation [55]
Original binary image(left) result binary image after dilation (right)

Figure 3.3 above shows how the algorithm works. It turns 0 to 1, if any of its neighbors are 1.

3.2.4. Connected Component Labeling

Although color images and data with higher dimensionality can also be processed, connected component labeling is used in computer vision to detect connected regions in binary digital images. It is an algorithmic application of graph theory, where subsets of connected components within an image are uniquely labeled based on a given heuristic [70].

This algorithm has two versions: one pass and two passes. One pass version goes through each pixel only once and for each pixel in an image, all the neighbor pixels are tested for connectivity to label connected components. It takes more memory space and

processing time than the two pass version especially in processing images with large number of small sized connected elements. However, it gives the same result as the two passes version [70].

Algorithm 3.1 below is a one pass version of connected component labeling algorithm.

Algorithm 3.1 - Connected - Region Extraction (One Pass)

1. Connected-component matrix is initialized to size of image matrix.
2. A marker is initialized and incremented for every detected object in the image.
3. A counter is initialized to count the number of objects.
4. A row-major scan is started for the entire image.
5. If an object pixel is detected, then following steps are repeated until (Index !=0)
    5.1. Set the corresponding pixel to 0 in Image.
    5.2. A vector (Index) is updated with all the neighboring pixels of the currently set pixels.
    5.3. Unique pixels are retained and already marked pixels are removed.
    5.4. Set the pixels indicated by Index to 1 in the connected-component matrix.
6. Increment the marker for another object in the image.

The two pass version scans the image two times; in the first pass, the algorithm goes through each pixel and checks the pixel above and to the left. And using these pixel's labels (which have already been assigned), it assigns a label to the current pixel. And in the second pass, it cleans up any mess it might have created, like multiple labels for connected regions [71].

The algorithm below (Algorithm 3.2) presents the two passes version of connected component labeling algorithm.

Algorithm 3.2 - Connected - Region Extraction (Two Pass)

First Pass:

1. Iterate through each element of the data by column, then by row (Raster Scanning)
2. If the element is not the background
    2.1. Get the neighboring elements of the current element
    2.2. If there are no neighbors, uniquely label the current element and continue
    2.3. Otherwise, find the neighbor with the smallest label and assign it to the current element
    2.4. Store the equivalence between neighboring labels

Second Pass:

1. Iterate through each element of the data by column, then by row
2. If the element is not the background
    2.1. Relabel the element with the lowest equivalent label

3.2.5. Hough Transform Algorithm

In digital images analysis, a frequently arising problem is detecting the simple shapes like straight line, circle or ellipse. In most of the cases an edge detector can be used as a pre-processing stage to obtain image points or image pixels that are on the desired curve in the image. But due to limitations in either the image data or the edge detector there may be missing or isolated or disjoint points or pixels on the desired curves as well as there may be spatial deviations between the ideal line or circle or ellipse and the noisy edge points as obtained from the edge detector. As a result, it is often non-trivial to group the extracted edge features to an appropriate set of lines, circles or ellipses. The purpose of the Hough transform is therefore, to address this type of problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects [75].

Let us consider a single isolated edge point or pixel (x, y) in the image plane. There could be infinite number of lines that could pass through this point. Each of these lines can be characterized as a solution to some particular equation [75]. A line can be expressed in the slope-intercept form as:

$$y = mx + c$$

Equation 3.3 – Slop Intercept Formula 1

where, m is the slope of the line with respect to x axis and c is the intercept on y axis made by the line. And any line can be characterized by these two parameters pair (m, c). For each line that pass through a given point (x, y), there is a unique value of c for each value of m, given by:

$$c = y - mx$$

Equation 3.4 – Slop Intercept Formula 2

Every point in image space (x, y) corresponds to a line in parameter space (m, c) and in the reverse way, each point in (m, c) space corresponds to a line in image space (x, y).

The Hough transform works by letting each feature point (x, y) vote in (m, c) space for each possible line passing through it. These votes are counted and stored in an accumulator.

Suppose that a particular (m, c) has one vote, this means that there is a feature point through which this line passes. If it has two votes, it means that two feature points lie on that line. If a position (m, c) in the accumulator has n votes, this means that n feature points lie on that line [75].

Algorithm 3.3 - Generalized Hough Transform Algorithm

1. Find all of the desired feature points in the image.
2. For each feature point
   2.1. For each possibility *i* in the accumulator that passes through the feature point
   Increment that position in the accumulator
   Find local maxima in the accumulator
   If desired, map each maxima in the accumulator back to image space

Alternative Representation of Lines

The slope-intercept form of a line discussed above has a problem with vertical lines as both m and c are infinite. To eliminate this problem of representing the point in the (m,c) space another way of expressing a line in (ρ, θ) form is used as:

$$x \cos \theta + y \sin \theta = \rho$$

Equation 3.5 –Alternative Representation of Lines

One way of interpreting this is to drop a perpendicular from the origin to the line. θ is the angle that the perpendicular makes with the x-axis and ρ is the length of the perpendicular. θ is bounded by [0, 2π] and ρ is bounded by the diagonal of the image. Instead of making lines in the accumulator, each feature point votes for a sinusoid of points in the accumulator. Where these sinusoids cross, there are higher accumulator values. Finding maxima in the accumulator still equates to finding the lines.

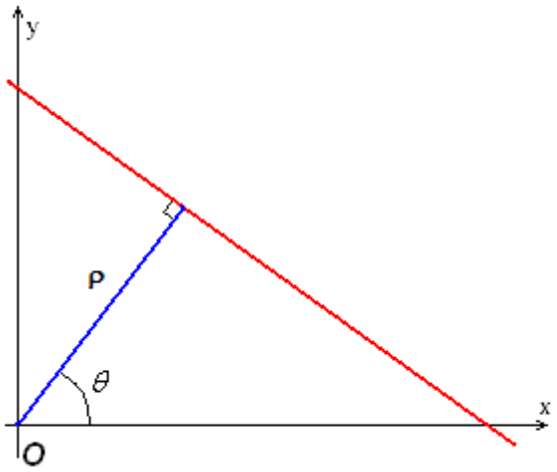


Figure 3.4 - Alternative Representation of Straight Line in (ρ, θ) Format [75]

The (ρ, θ) plane is sometimes called as Hough space for the set of straight lines in a two dimensional image space. The steps of implementation can be summarized below.

- For each image data point, a number of lines are plotted going through it, all at different angles.

- For each line a line is plotted which is perpendicular to it and which intersects the origin.
- The length and angle of each dashed line is measured.
- These the steps are repeated for each data point.
- A graph of length against angle, known as a Hough space graph, is then created.

For line matching in Hough Transform, the orientation of the line is one of the parameters. The orientation parameter can be changed sequentially in an incremental way to find out all the lines oriented in different directions [75].

3.3. Performance Evaluation

To measure the performance of the page segmentation technique proposed in this study, GCE and Match Score results are used. Precision, Recall, and F-measure are used to measure result found by integrating the proposed technique with the previous Amharic DIR system.

3.3.1. GCE

David et al. [74] define segmentation simply as a division of the pixels of an image into sets. And a segmentation error measure takes two segmentations S_1 and S_2 as input, and produces a real-valued output in the range [0..1], where zero signifies no error. First, define a measure of error at each pixel that is tolerant to refinement. Let \ denote set difference, and |x| the cardinality of set x. If $R(S,p_i)$ is the set of pixels corresponding to the region in segmentation S that contains pixel p_i, the local refinement error is defined as [74]:

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \backslash R(S_2, p_i)|}{|R(S_1, p_i)|}$$

Equation 3.6 – Local Refinement Error Formula

Note that this local error measure is not symmetric. Given this local refinement error in each direction at each pixel, two segmentation error measures are defined. Global

Consistency Error (GCE) forces all local refinements to be in the same direction. Local Consistency Error (LCE) allows refinement in different directions in different parts of the image. Let n be the number of pixels:

$$GCE(S_1, S_2) = \frac{1}{n} min\left\{\sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i)\right\}$$

Equation 3.7 – Global Consistency Error (GCE)

$$LCE(S_1, S_2) = \frac{1}{n}\sum_i min\{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\}$$

Equation 3.8 – Local Consistency Error (LCE)

3.3.2. Match Score

Phillips and Chhabra [61], explained match score as: the number of matches between the entities detected by segmentation algorithm and the entities in the ground truth. Below is the formula to calculate match scores.

Let *I* be the set of all image points, G_j the set of all points inside the j ground truth region, R_i the set of all points inside the i result region, T(s) a function that counts the elements of set s. MatchScore(i,j) which represents the matching results of the j ground truth region and the i result region is given by [61]:

$$MatchScore(i, j) = \frac{T(G_j \cap R_i \cap I)}{T(G_j \cup R_i \cup I)}$$

Equation 3.9 – Match Score

3.3.3. Precision, Recall and F-Measure

The two most frequent and basic measures for IR systems effectiveness are precision and recall [72]. Precision is the percentage of retrieved documents that are relevant, while Recall is the percentage of relevant documents that is retrieved from the total number of relevant document in the collection [4].

$$Precision = \frac{RR}{RR + IR}$$

Equation 3.10 – Precision

$$Recall = \frac{RR}{(RR + RN)}$$

Equation 3.11 – Recall

Where RR and IR are the number of relevant and irrelevant documents retrieved respectively and RN is the total number of relevant documents but not retireved.

Recall and precision are often conflicting goals in the sense that if one wants to see more relevant items (i.e., to increase recall level), usually more non-relevant ones are also retrieved (i.e., precision decreases). Hence, another effectiveness evaluation parameter called F-Measure is forwarded to balance this trade off [73].

$$F - Measure = \frac{2PR}{P + R}$$

Equation 3.12 – F-Measure

Where P and R are precision and recall, respectively.

Once suitable combinations of page segmentation techniques (i.e., text/graphics and word-level segmentation techniques) are selected, they are implemented and the best one is integrated with the Amharic DIR system. By preparing Amharic document image corpus an extensive experiment is also done to measure an improvement with Amharic DIR system.

CHAPTER FOUR

EXPERIMENTATION

The aim of this study is to experiment different page segmentation algorithms so as to select the suitable one for identifying text and non-text regions that are frequently happening in Amharic document images. The selected page segmentation technique is finally integrated with the Amharic DIR system.

For the experimentation HP Desktop computer with specification Intel® Core™ i3 CPU 550 @ 3.20 GHz (4 CPUs), 2GB RAM and Windows® XP professional edition operating system was used. MATLAB™ image processing toolbox 7.0 and Java™ programming language using NetBeans IDE 7.1.2 are used for developing prototype and integration.

4.1. Dataset Preparation

Since the goal of this research is to segment non-text areas (graphics, images, tables, logos, etc.) from text areas (text/graphics segmentation), Amharic document images containing graphics, tables, and pictures are collected and included in the existing data set prepared by Biniam [10].

The existing dataset that are collected by Binaim [10] includes real life documents (with different noise level) from books magazines, newspapers and regulations. This dataset also enables to evaluate the proposed technique in this work and measure to what extent the segmentation technique is insensitive to noise. Some handwritten (“kum tsihuf”) and typewritten documents are also included to see how the proposed techniques perform on such documents. Because, real life documents also include these kind of documents. And if we are supposed to come up with an Amharic DIR system used by public, these documents also need to be included in the dataset.

To summarize, the dataset prepared by Biniam [10], 50 (twenty five) Amharic document images which imbed graphics, images, tables, etc. and 5 (five) Amharic handwritten ("kum tsihuf") document images and 10 (ten) typewritten document images, are used in this study.

4.2. Page Segmentation in Amharic Document Images

All page segmentation algorithms are applied on noise filtered and binarized document images. Biniam's [10] methods for noise filtering and binarization are used. Built-in methods in MATLAB Image Processing Toolbox are used to implement connected component, watershed, dilation and Hough transform algorithms. To implement run length smoothing algorithm, the code is written in MATLAB programming language.

4.2.1. Connected Components Labeling

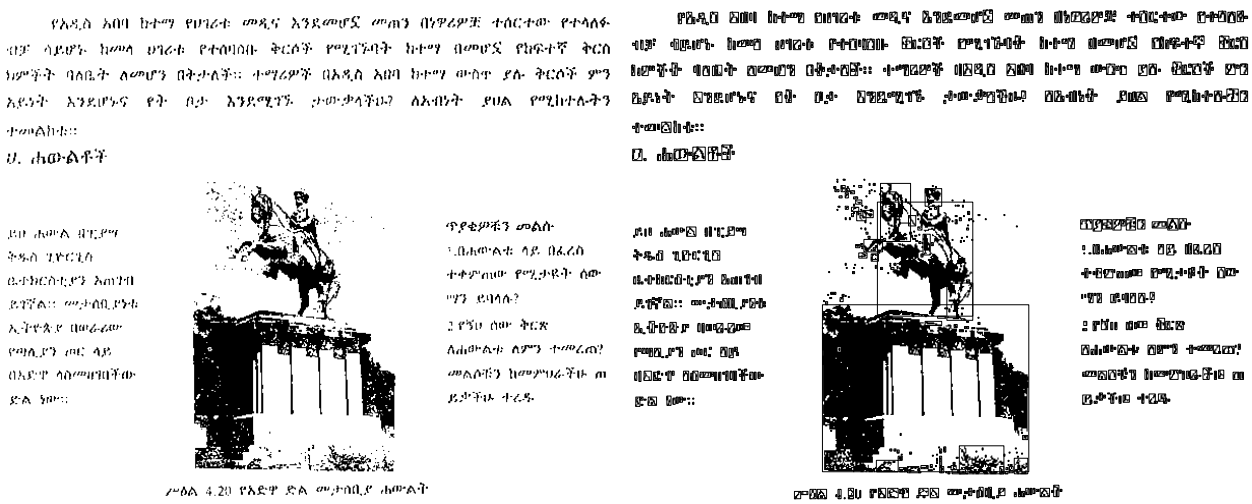
Connected components (CC) labeling algorithm which identify and label each connected component in a given binary image is implemented using MATLAB built-in method `bwconncomp()`. The code used to implement CC is given in Listing 4.1.

Listing 4.1 - Implementation of Connected Components

```
function [cc,num] = ConnectedComp(bw)
cc = bwconncomp(bw,4) % using 4 connectivity
%storing number of connected components
num=cc.NumObjects;
```

The function ConnectedComp() extract connected components and assign it to cc variable. In this study 4 connectivity of pixels are used.

Figure 4.1 below shows how CC algorithm identifies connected components in a given image.



(a) (b)

Figure 4.1 – Result of CC Labeling:
(a) Original binary image, (b) Segmented image using CC labeling

The algorithm works well in identifying connected components. However, parts of broken characters and disconnected figure elements are considered as separate components.

4.2.2. Components Width, Height and Area Analysis

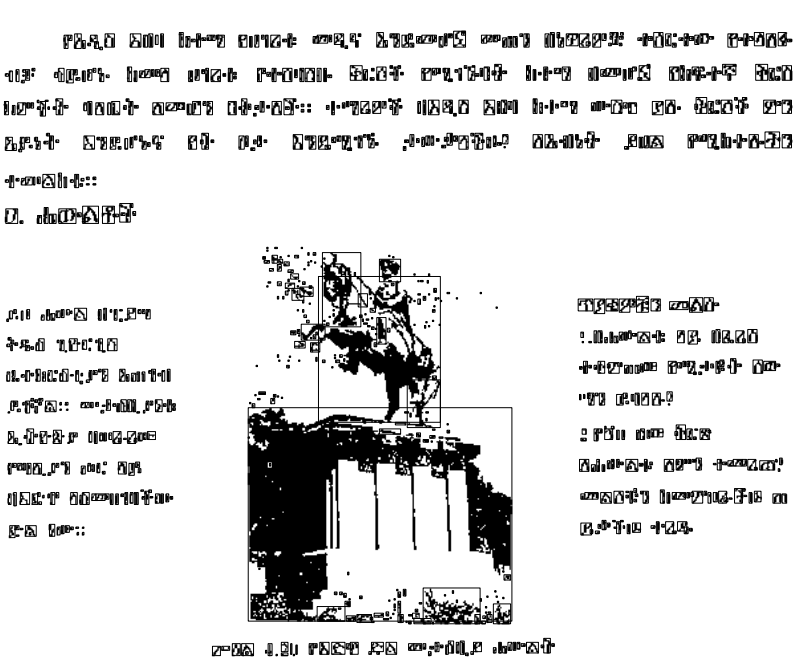
Components width, height and area analysis is used to identify big connected elements like: images, graphics, logos, etc. and small connected elements like punctuation marks and small dots. Images and graphics usually have larger area and height or width than normal text while punctuation marks and dots have smaller area and height or width. As we can see from Figure 4.2 below there are about six big components and a number of small components which are less than normal text.

Figure 4.2 (b) indicated that we have two connected components which have extremely bigger areas and other 10 components which have larger areas than the other components. But we want only six of them to be removed as they are not part of the text. Components height and width graphs 4.2 (c) and (d) clearly indicated the six bigger elements. Therefore, the researcher used the intersection of components area

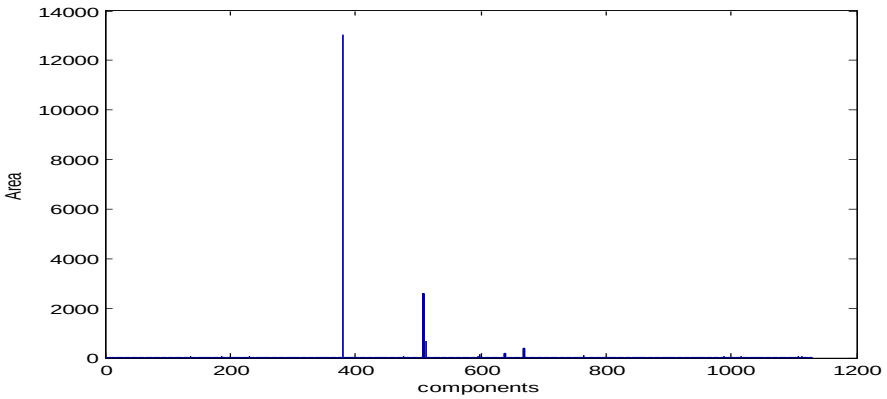
(greater than a threshold value 1.5*median area) and height (greater than a threshold value median height + 7) to remove large connected elements. That means if a given connected component has an area greater than 1.5 * the area of a median connected component and a height greater than height of median connected component +7, it is graphics, picture, table, etc. or segment of graphics, picture, table, etc. The threshold values are found by experiment.

It is also proved that we can remove punctuation marks, vertical lines, horizontal lines, and unnecessary dots form the image by setting thresholds. Threshold values of median area/8, median height/4 and median width/3 are used. If the area of a given connected component is less than the area of the median connected component divided by eight or its height is less than the height of the median connected component divided by 4 or its width is less than the width of the median connected component divided by 3, then it is either part of punctuation marks or small dots in the document image.

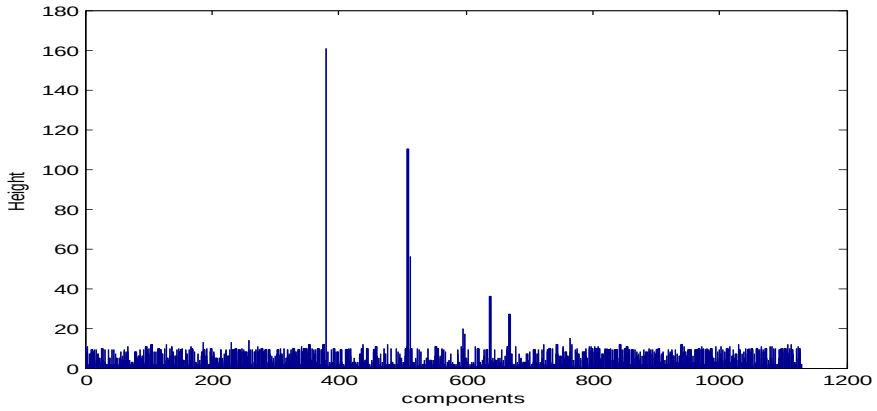
The thresholds used to detect large elements also works to detect tables in the document images.



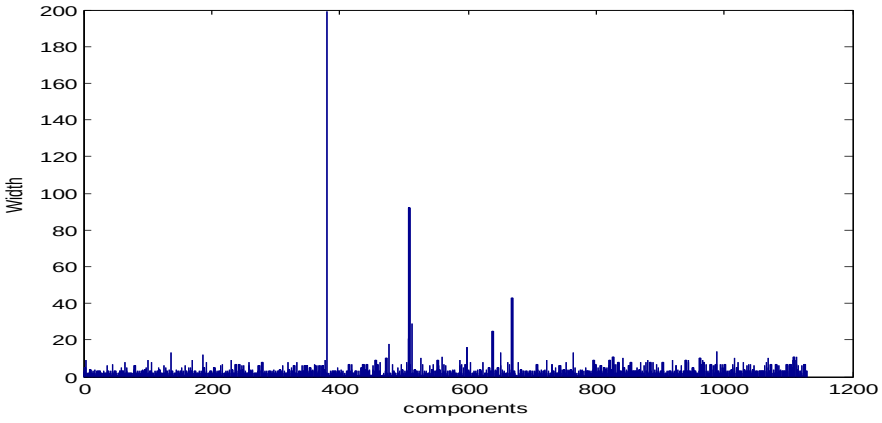
(a)



(b)



(c)



(d)

Figure 4.2 –CC Area, Height and Width Analysis:

(a) CC segmented binary image, (b) Components areas graph, (c) Components height graph (d)Components width graph

4.2.3. Hough Transform

The purpose of using Hough transform in this study was to test if it helps detect lines from document images and delete them, because our concern is on text not on lines in document images. If we have table that contain text, we have to identify the table (frame) and the content (words).

Listing 4.2 below shows the codes used to implement Hough transform algorithm.

Listing 4.2 - Implementation of Hough Transform

```
function [bw1] = HughLine(bw)
bw1 = edge(bw, 'canny', [], 2);
[H,theta,rho] = hough(bw1);
peaks  = houghpeaks(H,100,'threshold',ceil(0.2*max(H(:))));
lines = houghlines(bw1,theta,rho,peaks);
```

The MATLAB built-in function `hough()` is used to implement the algorithm. Different peak values are tested to detect lines. However, when the peak value used detect all the lines, it also consider text parts with condensed foreground pixels as lines and when the peak value is adjusted to exclude the text parts from being considered as lines it missed some lines. Figure 4.3 shows result of using different peak values.

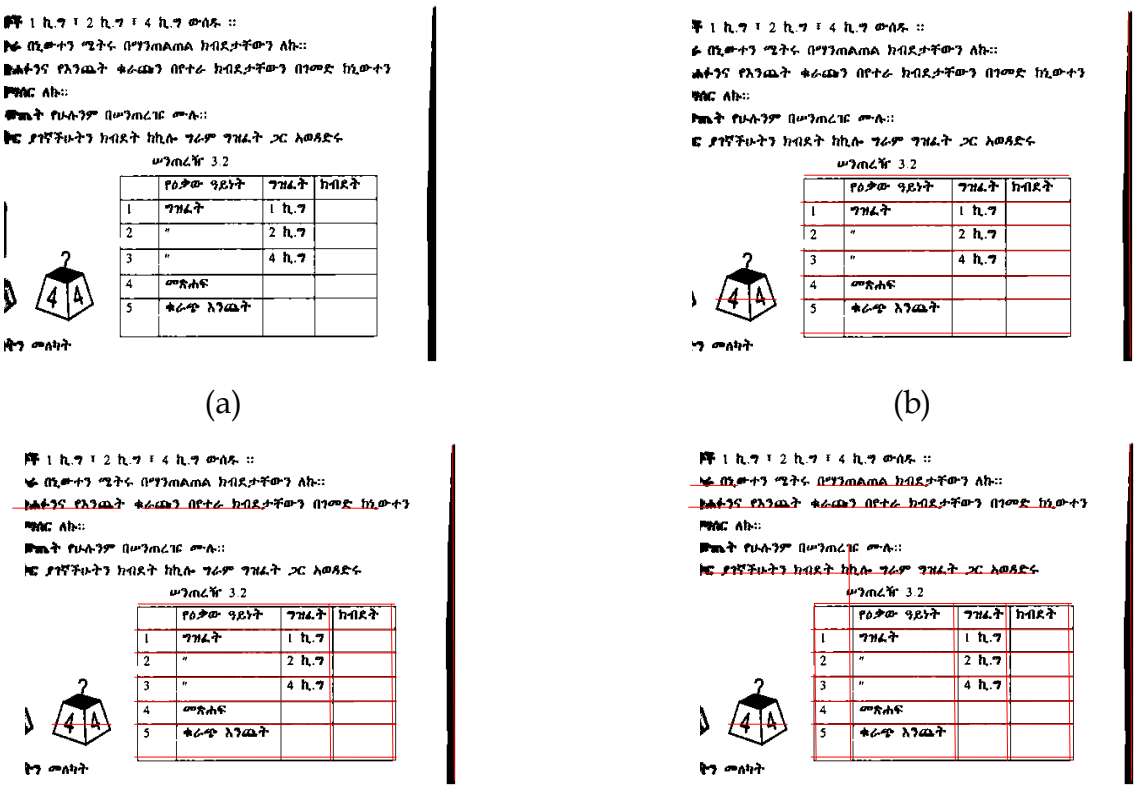


Figure 4.3 – Implementation of Hough Transform with Different Thresholds:
(a) Original binary image, (b) Identify some part of tables only, (c) Some text parts are detected as lines and
(d) Detect all lines and some more text areas as lines

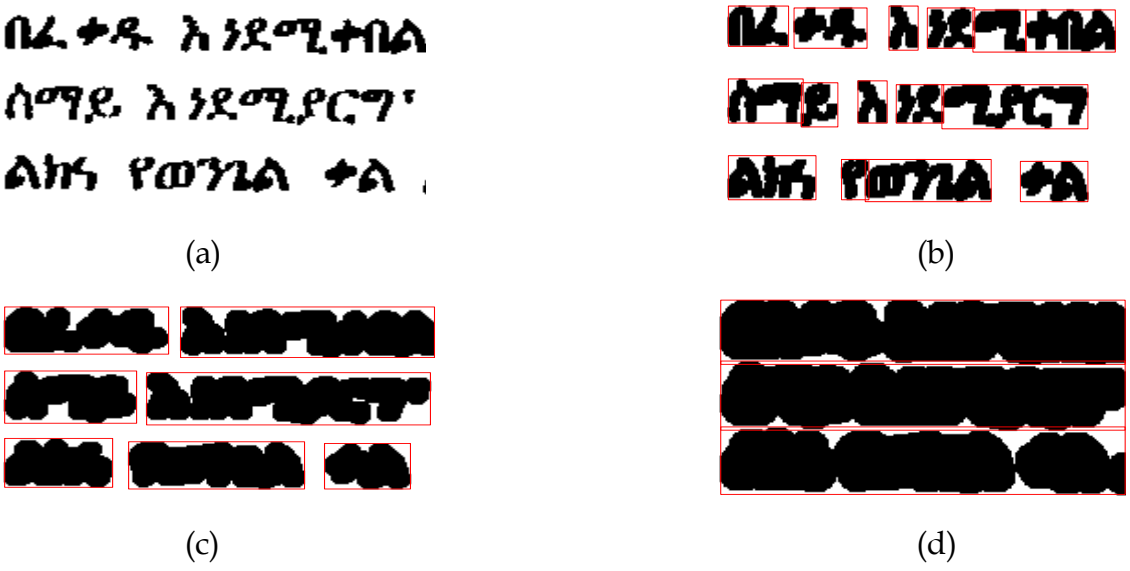
4.2.4. Dilation

Dilation is used to connect characters in words. Space between words is larger than space between characters in words. The threshold is identified by experiment. The threshold value identified by our experiment is median height of connected components divided by 3.8 (med/3.8) the median is taken from connected components left after removing small and large connected components. Median is choosen to remove the effect of shorter characters like "መ፧ሠ፧ወ" and taller characters like "ጀ፧ኗ፧ኸ".

Listing 4.3 - Implementation of Dilation

```
function [dialatedIm] = Dilat(bw,tresh)
dialatedIm = bwdist(~bw) >= tresh;
```

The experiment showed that using larger thresholds will result in merged words and using small threshold will result in over segmentation. Figure 4.4 below shows the result of applying different thresholds.



(a) (b)

(c) (d)

Figure 4.4 – Dilation Using Different Thresholds:
(a) Original image (b) Result of med/4.5 (smaller threshold) (c) Dilated by med/3.8 (d) Result of med/2.5 (larger threshold)

4.2.5. Watershed Segmentation

Watershed algorithm is implemented using MATLAB built-in function `WaterSh()`. The algorithm segments document images to word images when integrated with dilation or HRLS. Dilation and HRLS algorithms are used to connect characters in a given word. The code below (Listing 4.3) is used to apply watershed algorithm.

Listing 4.4 - Implementation of Watershed

```
function [Segmented] = WaterSh(bw)
Segmented = watershed(bw);
```

Figure 4.5 below shows the result of connected component based watershed algorithm. Dilation is used to connect characters in words. After dilation, the CC algorithm labels each word with the same tag. Finally the watershed algorithm segment the document based on the connected components labels as shown in Figure 4.5 (b).

The image in between the text is removed before segmentation by applying the CC area height analysis and thresholds discussed above.

(a) (b)

Figure 4.5 – Implementation of Watershed Algorithm:

(a) Original grayscale image, (b) Segmented image using combination of CC, dilation and watershed

Watershed segmentation algorithm attaches white spaces as part of a nearby component which creates difficulty in extracting coordinates of word images after segmentation.

4.2.6. Horizontal Run Length Smoothing (HRLS)

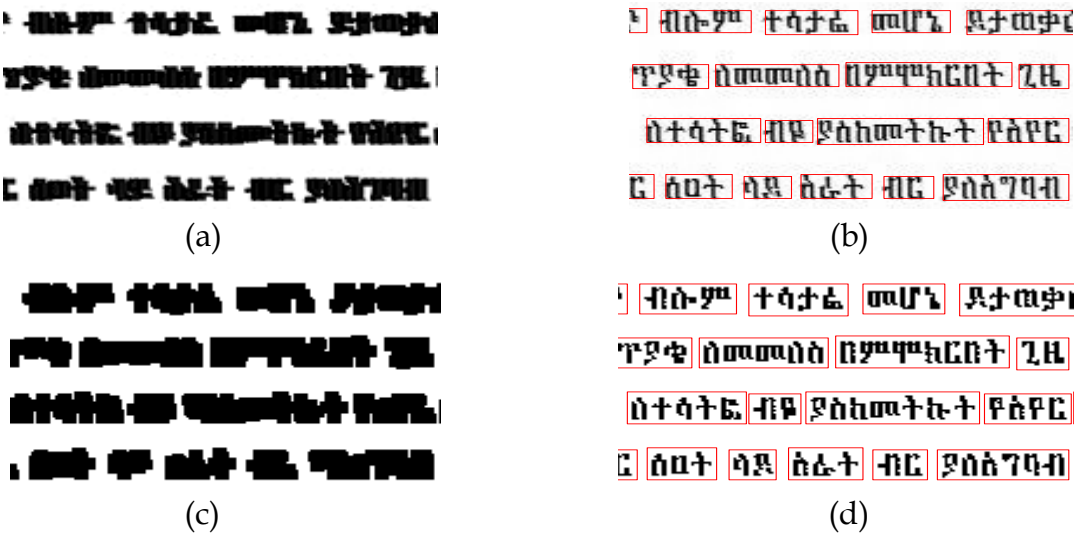
HRLS is the other algorithm tested to connect characters in words. The algorithm gives the same result with dilation. The only difference is dilation expand characters in all directions (up, down right and left) with the threshold amount while HRLS expand characters in only one direction (to the right). Therefore, the threshold used by dilation times two is used for HRLS.

Listing 4.5 - Implementation of Horizontal Run Length Smoothing

```
function [HRLSResult] = HRLS(bw1,bw2)
[rows, columns] = size(bw1);
for i = 1 : rows
    for j = 1 : columns
        if (j>4)
            if (bw2(i,j)==1)
                continue;
            else
                if(bw2(i,j-1)==1|bw2(i,j-2)==1|…
                   bw2(i,j-3)==1|bw2(i,j-4)==1)
                   bw1(i,j)=110;
                end
            end
        end
    end
end
```

The code shown above in Listing 4.5 shows implementation of HRLS which expand each character by four pixels.

For documents with small line spacing, HRLS can be used because Dilation may connect words in different lines. However, as it is observed from Figure 4.6 below the result of both algorithms (Dilation and HRLS) to segment words is the same in our document collection. Dilation is used in this study because the MATLAB built in function for Dilation is faster than the function written to implement HRLS algorithm and both performed equally in our document collection.



(a) (b)

(c) (d)

Figure 4.6 – Dilation and HRLS Connected Words

(a) HRLS connecting characters (b) Connected components created using the result of a (c)Dilation connecting characters (d) Connected components created using the result of c

Table 4.1 shows the average performance score in terms of GCE and match score on different level noisy document images.

Algorithm Used to Connect Characters in a Word	Proposed Technique			
	GCE		Match Score	
	Watershed	CC	Watershed	CC
HRLS	0.145	0.135	0.855	0.865
Dilation	0.145	0.135	0.855	0.865

Table 4.1 – Performance of Dilation and HRLS in Identifying Words

4.3. Proposed Segmentation Technique

Based on the implementation and test results, the researcher proposed a combined segmentation algorithm of CC, Dilation and Watershed algorithms. The flow of the proposed techniques is presented in Figure 4.7 below.

The input for proposed segmentation technique is cleaned binary document images using noise removal and thresholding technique proposed by Biniam [10]. The proposed combined segmentation technique first apply CC algorithm. This algorithm label each connected components with a unique tag.

The next step is to analyze connected components Area, Height, and Width. Based on the analysis we remove big components (graphics, images, tables, etc) and small components (punctuations, dots, horizontal and vertical lines). Then, we apply Dilation on the resulted image to connect characters in words.

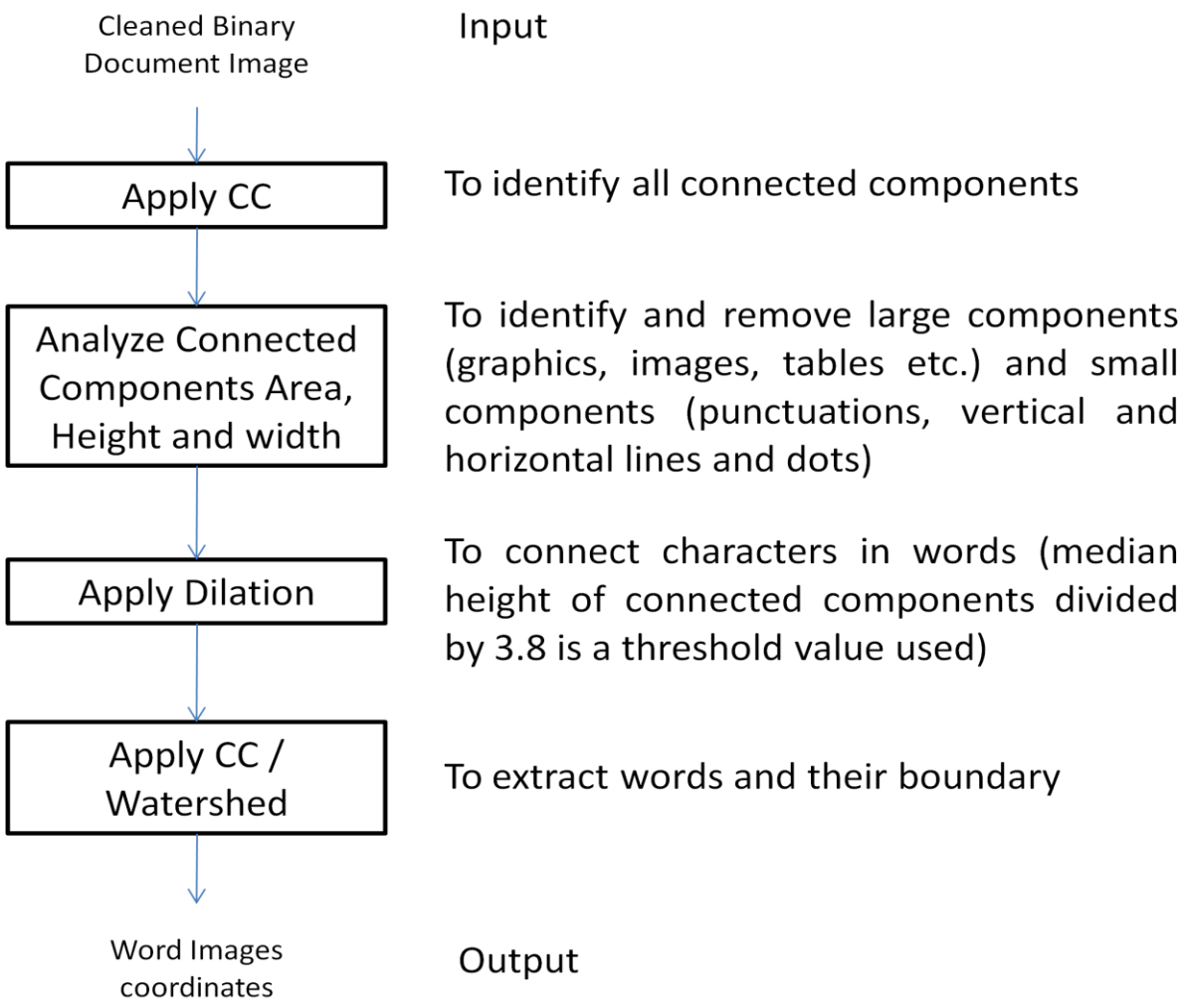
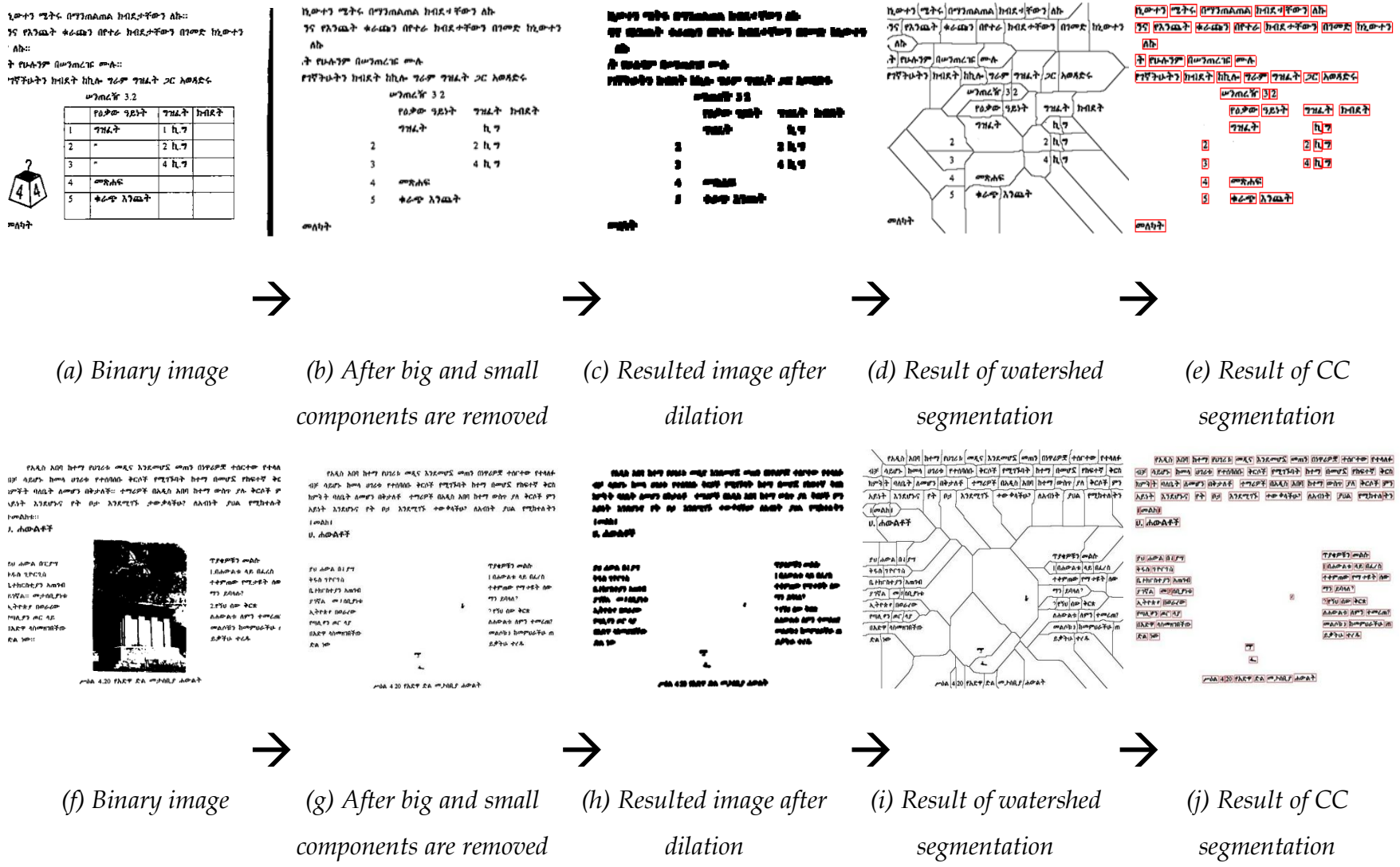


Figure 4.7 – Flow of Proposed Technique

Finally, we apply either CC or Watershed algorithm to extract words and their boundary (coordinates). The coordinates extracted by the technique is written to a text file and use as an input for feature extraction module proposed by Adane [11]. The performance results presented in the next section showed that CC is better than Watershed.

The figure below (Figure 4.8) shows how the proposed technique performs in steps presented above.



(a) Binary image *(b) After big and small components are removed* *(c) Resulted image after dilation* *(d) Result of watershed segmentation* *(e) Result of CC segmentation*

(f) Binary image *(g) After big and small components are removed* *(h) Resulted image after dilation* *(i) Result of watershed segmentation* *(j) Result of CC segmentation*

Figure 4.8 – Result of Proposed Technique at Each Steps:

The upper raw (a)-(e) shows the flow in the presence of table and the lower one (f)-(j) in the presence of picture

4.4. Performance Result

We have seen that connected components Area, Height and width analysis and Dilation are better than Hough transforms and HRLS to remove pictures, graphics, tables, dots, punctuations, etc. and connect characters in words respectively.

To decide about whether CC or Watershed best perform when integrated with connected components Area, Height and width analysis, and Dilation on different type of documents, three experiments are conducted. The experiments are conducted on noisy documents with different noise level, on handwritten documents ('kum tsihuf') and typewritten documents and on documents containing images and tables.

Experiment Results

GCE and Match Score results of the proposed combined segmentation technique on different document types are presented in table 4.2.

Document Type		Proposed Technique			
		GCE		Match Score	
		Watershed	CC	Watershed	CC
(i)	Low-Level Noisy Documents	0.11	0.06	0.89	0.94
	Medium-Level Noisy Documents	0.06	0.06	0.94	0.94
	High-Level Noisy Documents	0.18	0.15	0.82	0.85
	Very High-Level Noisy Documents	0.26	0.24	0.74	0.76
	Average	**0.152**	**0.128**	**0.848**	**0.873**
(ii)	Handwritten Documents ('kum tsihuf')	0.62	0.55	0.38	0.45
	Typewritten Documents	0.10	0.07	0.90	0.93
(iii)	Documents Containing Images	0.20	0.03	0.80	0.97
	Documents Containing Tables	0.05	0.03	0.95	0.97

Table 4.2 – Performance of Combined Segmentation Technique

The result indicated that integrating CC algorithm with components Area, Height and width analysis and Dilation gives better result. In all document types CC outperforms Watershed. Therefore, the integration of CC with CC Area, Height and Width analysis, and Dilation is proposed.

However, both algorithms poorly perform on handwritten documents. Because, all handwritten document images used very high level noisy images and features of the documents is highly degraded because of old age.

4.5. Integrating the Proposed Segmentation Algorithm with Amharic DIR System

The result of the above experiments showed that integration of CC with components Area, Height, Width analysis and Dilation performs better. The page segmentation module developed in this study using MATLAB is integrated with the previous Amharic DIR systems developed in Java. To integrate the MATLAB code with Java, MATLAB Builder JA software is used. MATLAB® Builder ™ JA enables to create Java ™ classes from MATLAB® programs. And the Java classes can be integrated into Java programs [71].

Listing 4.6 - Integrating the MATLAB Implementation with Java

```
public void ImageSegmentation()
{
    MWNumericArray n = null; // Stores input value
    Object[] result = null; // Stores the result
    ImageSegmentationClass ConComp = null;
        try
        {
            ConComp = new ImageSegmentationClass();
            ConComp.PageSegmentation();
            System.out.println("Image Segmentation Done!!.");
        }
        catch (Exception e)
        {
            System.out.println("Exception: " + e.toString());
        }
        finally
        {
            MWArray.disposeArray(n);
            MWArray.disposeArray(result);
            if (ConComp != null)
            {
                ConComp.dispose();
            }
        }
```

The code above in Listing 4.6 is written to call the MATLAB file that is compiled as Java package.

4.6. Experimental Result of Amharic DIR System

To evaluate the performance of Amharic DIR, noisy document image (with low-level, medium-level, high-level and very high-level noise), documents with tables and images, typewritten document and handwritten documents ('kum tsihuf') are used. The query words used for noisy document are those prepared by Biniam [10], for comparing performance improvement precision, recall & F-measure. The performance registered before and after integration of the proposed technique for different noise level noisy documents is presented below. Table 4.3 presents performance result for low-level noisy document images.

Query words	Before integration of the proposed technique			After integration of the proposed technique		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
መንግስት፡	100	100	100	100	100	100
ሥራ፡	100	100	100	100	100	100
አገልግሎት፡	100	67	80.24	100	100	100
ርዕስ	100	100	100	100	100	100
አበል	100	50	66.67	100	50	66.67
Average	**100**	**83.4**	**89.38**	**100**	**90**	**93.33**

Table 4.3 – System Performance on Low - Level Noisy Document Images

The performance for low level document images is increased by 3.95% F-Measure which shows that segmentation algorithm help the system detect more words in low level noisy document images. Although the segmentation algorithm used correctly segment the query word 'አበል' in all the documents containing the word, recall for the query word 'አበል' can't be increased. The cause is feature degradation due to preprocessing (thresholding) in some of the documents where the feature of the word is poor because of scanning problems or noise.

As presented in table 4.4 below, the integration of the proposed technique also increased the performance of the system for medium-level noisy document images by 3.8% F-Measure.

Query words	Before integration of the proposed technique			After integration of the proposed technique		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
አገር	100	50	66.67	100	75	85.71
ቅርስ	100	100	100	100	100	100
ቤት	100	100	100	100	100	100
ክፍል	100	50	66.67	100	50	66.67
ምስል	100	100	100	100	100	100
Average	**100**	**80**	**86.67**	**100**	**85**	**90.47**

Table 4.4 – System Performance on Medium - Level Noisy Document Images

Again the segmentation algorithm help the system detect more words and improve effectiveness in medium level noisy document images. Feature degradation is also a problem in medium-level noisy document images which decrease recall in two query words.

As depicted in table 4.5, the performance of the system showed lower performance on high level document images by 1.34% F-Measure. High-level and very high-level noisy document images are highly exposed to feature degradation. And the segmentation algorithm used could not identify more than 85% and 76% of words from high-level and very high-level noisy documents respectively. Therefore, it could not increase recall in these document images.

(a) (b)

Figure 4.9 – Effect of Featuer Degradation:
(a) Degraded features resulted in wrong segmentation (b) Degraded word segmented correctly but not recognized because of feature dissimilarity

Query words	Before integration of the proposed technique			After integration of the proposed technique		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
ቅዱስ	100	75	85.71	100	75	85.71
አለም	100	66.67	80	100	66.67	80
ምሥጋና	50	100	66.67	50	75	60
ኢትዮጵያ	100	75	85.71	100	75	85.71
አንድ	100	75	85.71	75	100	85.71
Average	**90**	**78.33**	**80.76**	**85**	**78.33**	**79.42**

Table 4.5 – System Performance on High - Level Noisy Document Images

Feature degradation while removing small components also reduces recall in one of the queries in high-level noisy document images.

Query words	Before integration of the proposed technique			After integration of the proposed technique		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
ቅዱስ	100	50	66.67	100	50	66.67
አንድ	100	100	100	100	75	85.71
ኢትዮጵያ	100	50	66.67	100	75	85.71
አለም	66.67	100	80	66.67	100	80
አብያተ	50	50	50	75	50	60
Average	**83.33**	**70**	**72.67**	**88.33**	**70**	**75.62**

Table 4.6 – System Performance on Very High - Level Noisy Document Images

As shown in table 4.6 retrieval effectiveness of the system is increased by 2.95% F-measure on very high-level noisy document images.

The performance of the system is also measured on documents containing images, documents containing tables, typewritten documents and handwritten documents ('kum tsihuf'). The previous system (before integration of the proposed system) does not work on the presence of pictures, and tables.

Accordingly, Table 4.7 presents experimental results of Amharic DIR system. F-Measure of 88.78% and 93.33% is achieved on documents containing images and tables respectively. However, the system can't retrieve documents from both typewritten and 'kum tsihuf' documents. As the segmentation algorithm works well on typewritten documents, the result is because of feature dissimilarity between typewritten words and computer written words. But, in historical documents the segmentation algorithm also does not work satisfactorily. Thus, the result is because of both segmentation error and feature dissimilarity.

Document type	After integration of the proposed technique		
	Precision	Recall	F-Measure
Documents containing images	83.33	95	88.78
Documents containing tables	100	90	93.33

Table 4.7 – System Performance on Different Document Image Types

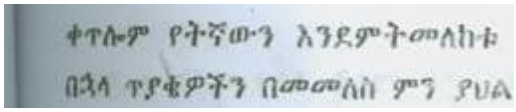
In addition to feature differences, feature of typewritten and 'kum tsihuf' documents is different from computer fonts and feature of these documents is highly degraded while thresholding and segmentation. Degraded feature of 'kum tsihuf' documents and slimness of typewritten characters exposed these documents for feature degradation while scanning, noise removal and thersholding, and removing small components while segmentation. We removed small components by considering them as punctuations and dots to improve feature extraction result.

4.7. Findings and Challenges

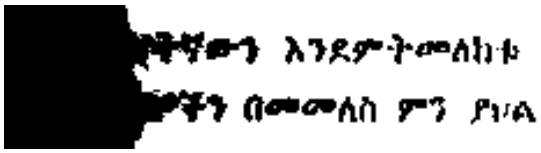
This study attempted to integrate effective page segmentation technique. The experiment results showed that connected components Area, Height and Width analysis performs better to remove both large components and small components. And

Integration of connected components Area, Height and Width analysis, Dilation and CC is proposed for segmentation.

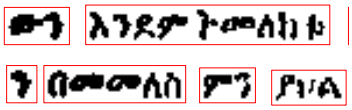
One of the challenges in this study is the existence of black shade in some document images. Black shades often introduced to document images while scanning. The shade hides part of text and connected to characters nearby in document images while preprocessing (thresholding). While trying to remove such shade as background by increasing the threshold value it highly degrade features of characters in the image. Figure 4.10 below shows the effect of the existence of black shade thresholding and segmentation.



(a) Original image



(b) Effect of thresholding



(c)Final effect on segmentation

Figure 4.10 – The Effect of Thresholding in the Presence of Shadow

The other challenging is thresholding which resulted in broken characters especially on very high-level noisy documents, 'kum tsihuf' and typewritten documents with degraded features. The figure below shows how thresholding such documents affects features of characters.

(a) Original image

(b)Thresholded image

Figure 4.11 – Thresholding Degraded Image

Although the technique proposed in this research performs well in text/graphics and word-level segmentation, it segments text with large font sizes as part of images because it mainly focuses on size to identify images, graphics, tables, etc.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

Large amount of handwritten, typewritten and printed documents contain numerous information and knowledge of different areas. To make the information and knowledge embedded in these documents accessible to the public, designing recognition based or recognition free IR systems is vital. Accordingly, researchers attempt to design recognition free retrieval system for different languages. This work is also to add on the attempt to design full-fledged Amharic DIR system.

5.1. Conclusions

The main objective of this study is to design effective page segmentation technique (i.e. text/graphics segmentation and word-level segmentation) which can be applied to document images in the presence of different level of noise, pictures, graphics, logos, tables, etc.

In this research, five page segmentation algorithms namely: Hough transforms CC, HRLS, Dilation and Watershed are evaluated. During the experiments made, it is found out that connected components Area, Height and Width analysis performs better to remove both large components (pictures, graphics, tables, etc.) and small components (punctuations, dots, vertical lines and horizontal lines).

Two of the algorithms (Dilation and HRLS) perform the same in connecting characters in words. MATLAB building function for Dilation is selected, because it is observed that it performs faster than HRLS in our experimentation.

Integration of connected components Area, Height and Width analysis, Dilation and CC performs better than integration of connected components Area, Height and Width analysis, Dilation and Watershed in all the three experiments made to identify the better

combination. The integration of the selected algorithms is tested on different document images (i.e. different level of noisy documents; documents containing pictures, graphics, logos, tables, etc.)

As it is shown in the experiment results, the performance of the proposed technique (i.e. components Area, Height and Width analysis, Dilation and CC) is measured to be: average GCE score 0.135 and average Match Score 0.865 in different level noisy document images, GCE score 0.07 and Match Score 0.93 in typewritten documents, GCE score 0.55 and Match Score 0.45 in historical documents ('kum tshihuf'), GCE score 0.03 and Match Score 0.97 in documents containing pictures, and GCE score 0.03 and Match Score 0.97 in documents containing tables.

By integrating the proposed technique with the previous Amharic DIR system an increase of 2.34% F-Measure is obtained in noisy document images (i.e. low – level, medium – level, high – level and very high – level noisy documents).

Moreover, 88.78%F-Measure for retrieving form documents containing images, and 93.33%F-Measure for retrieving form documents containing tables is scored. The experiment showed that the proposed technique is not suitable for handwritten document images ('kum tsihuf') unless feature recovery technique is integrated to it.

However, the proposed page segmentation technique segments text with large font sizes as part of images or graphics because it mainly focuses on size (Area, Height and Width) to identify images, graphics, tables, etc. Therefore, further researches to improve segmentation need to be conducted.

5.2. Recommendation

Although, this research contributed a lot for the attempt to develop a full-fledged Amharic DIR system, there are also some other issues that need to be addressed to improve effectiveness and efficiency of the current system.

- This study identifies and excludes images and graphics using their size. Consequently, it removes text with larger font size. Thus, tackling this problem is one of future research directions.
- The current system suffers from shades which are introduced during scanning as they hide some words and connected to some characters. Therefore, further researches need to be conducted to handle this problem.
- Although the current system segments well typewritten documents, retrieval results are very poor because "Geez 1" font is used for query rendering. So, the future researches would address retrieving such documents to be retrieved.
- Historical documents usually lose some features of characters and sometimes broken characters exist in such documents. Therefore, feature recovery techniques should be explored and integrated in future researchs.
- Future researches can also improve the performance of the system by exploring and integrating query expansion techniques suitable for Amharic DIR system.
- The thresholding algorithm used to change the document image into binary image format broke features of characters especially on very – high level noisy documents, historical documents and typewritten documents. Hence, improving the performance of the thresholding algorithm must be considered in future works.
- The performance of the system can be improved if future researches can design a stemmer for Amharic word variants in document images.
- Skew detection and correction is also a research issue for it had not been addressed by any of the researches so far including the current work.
- Developing DIR system to other Ethio-Semitic languages is also another open research area.

References

[1] Charles, O., Joan, S., and Richard, W. Studies on Information as an Asset. *Journal of Information Science* , 159-166, 2003.

[2] Marinai, S. Introduction to Document Analysis and Recognition. *IEEE Transactions on PAMI*, 27(1): 23-43, 2006.

[3] Akram, S., Dar, M., and Quyoum, A. Document Image Processing - A Review. *International Journal of Computer Applications*, 10(5): 234-243, 2010.

[4]. B. Ricardo and B. Ribeiro-Neto. *Modern Information Retrieval*. A Division of the Association for Computing Machinery: Addison-Wesley, ACM Press, 1999.

[5] Mesfin, W. *Amharic Document Image Retrieval without Explicit Recognition*. M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2009.

[6] Anand, K., Jawahar, C., and Manamatha, R. Efficient Search in Document Image Collections, In Proc. *of the Asian Conf. on Computer Vision (ACCV), Part I, LNCS 4843*, pages. 586-595, 2007.

[7] Chew, L., Weihua, H., Zhaohui, Y., and Yi, X. Imaged Document Text Retrieval Without OCR. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(6): 838-844, 2002.

[8] Shijian, L., Linlin, L., and Chew, L. Document Image Retrieval Through Word Shape Coding. *IEEE Transaction on Pattern Analysis and Machine Intellegnece,* 30(11):, 1913-1918, 2008.

[9] Million, M. *Recognition and Retrieval from Document Image Collections*. Ph.D Dissertation, International Institute of Information Technology , Hyderabad, India, 2008.

[10] Biniam, A. *Retrieval form Real-Life Amharic Document Images*. M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2012.

[11] Adane, L. Feature *Extraction and Matching in Amharic Document Image Collections*. M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia 2011.

[12] Abreham, G. *Searching in Amharic Document Image Corpus*. M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2010.

[13] Million, M., and Jawahar, C. Optical Character Recognition of Amharic Documents. *African Journal of Information and Communication Technology*, 3(2):, 53-66, 2007.

[14] Mandl, T. *Recent Developnents in the Evaluation of Information Retrieval Systems: Moving Towards Diversity and Practical Relevance*. Information Science University of Hildesheim Marienburger, Hildesheim,Germany, 2008.

[15] Manesh, B. and M. Shirdhonkar. Document Image Retrieval: An Overview. *International Journal of Computer Applications (0975 – 8887)* 1(7): 114-119, 2010.

[16] Christian Shin, David Doermann. Document Image Retrieval Based on Layout Structural Similarity. *IPCV*, 606-612, 2006.

[17] Mohammadreza Keyvanpour, and Reza Tavoli. Classification and Evaluation of Document Image Retrieval System. *WSEAS TRANSACTIONS on COMPUTERS*, 10(11): 329-338, 2012.

[18] Kareem, D. and Ossama, E. Retrieving Arabic Printed Document: A Survey. *IBM Technology Development Center*, Cairo, Egypt, 2006.

[19] Konstantinos, Z., Kavallieratou, and Nikos, P. A Document Image Retrieval System. *Engineering Applications of Artificial Intellegnece* , 872-879, 2010.

[20] Ermias, A. *Recognition of Formatted Amharic Text Using OCR Techniques*. M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 1998.

[21] John, C., and Fiaz, H. Amharic Character Recognition Using a Fast Signature Based Algorithm. *In Proc. of the IEEE Conference on Image Visualization*, pages 384-389, 2003.

[22] David Doermann. The Indexing and Retrieval of Document Images: A Survey. *Computer Vision and Image Understanding (CVIU)*, pages 287- 298, 1998.

[23] Million M. and C. V. Jawahar. Matching word images for content-based retrieval from printed document images. International *Journal on Document Analysis and Recognition*, 11(1): 29-38, 2008.

[24] Shuyong Bai, Linlin Li and Chew Lim Tan. Keyword Spotting in Document Images through Word Shape Coding. Proc. *of 10th International Conference on Document Analysis and Recognition*, pages 331-335, 2009.

[25] Shijian Lu, Linlin Li, and chew lim tan. Document Image Retrieval through Word Shape Coding. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1913-1918, 2008.

[26] K. Zagoris, N. Papamarkos and C. Chamzas. Web Document Image Retrieval System Based On Word Spotting. In *proc. of IEEE International Conference on Image Processing*, pages 477-480, 2006.

[27] Srihari S., et al. Document Image Retrieval Using Signatures as Queries. In *porc. Of Second International Conference on Document Image Analysis for Libraries*, 2006.

[28] R. Kasturi, L. O'gorman and V. Govindaraju. Document Image Analysis: A primer. *Sadhana,* 27(1): 3-22, 2002.

[29] L. J. Galbiati. *Machine Vision and Digital Image Processing Fundamentals*, New Jersey, Prentice Hall, 1990.

[30] H. K. A. Devi. Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script. *Asian Journal Image Processing*, 1(3): 161-165, 2006.

[31] A. Karthikeya. *Image Noise Reduction Algorithms*. Digital Signal Processing, ElectronicsBus Magazine, 2012.

[32] Pavan Kumar M. N. S. S. K. and Jawahar C. V. Information Processing from Document Images. In *proc. of Information Technology: Principles and Applications*, pages 522—547, 2004.

[33] Wu, V. and Manmatha, R. Document Image Clean-Up and Binarization. In *proc. of SPIE conference one Document Recognition,* San Jose, California, 1998.

[34] Salton and McGill. *Introduction to modern information retrieval*. McGram Hill Book Company, 1983.

[35] Skarbek, W., Koschan, A., Bericht, T., Veroffentlichung, Z., and Klette. Color mage Segmentation: A Survey. 1994.

[36] Ryszard S. Chora´s. Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems. *International Journal of Biology and Biomedical Engineering*. 1(1): 6-16, 2007.

[37] Noureddine Abbadeni. A New Similarity Matching Measure Application to Texture-Based Image Retrieval. In *proc. of the 3rd international Workshop on texture analysis and synthesis*, pages 1-6, 2003.

[38] Khurram K. Recognition. *Analysis and Retrieval of Historical Document Images*. Ph.D Thesis, Universite Paris Descarte, Paris, 2009.

[39] Antonacopoulos, A. and Karatzas. Emantics based content extraction in typewritten historical documents. In *proc. of the 8th International Conference on Document Analysis and Recognition* pages 48–53, 2005.

[40] Baird, H. S. Difficult and urgent open problems in document image analysis for libraries. *In 1st International workshop on Document Image Analysis for Libraries*, 2004.

[41] Le, D., Thoma, G., and Wechsler, H. Automated Borders Detection and Adaptive Segmentation of Binary Document Images. In *proc. of the 13th International Conference on Pattern Recognition*, pages 737-741, 1996.

[42] Okun, O., Doermann, D., and Pietikainen, M. *Page segmentation and zone classification: The state of the art: Technical report*, University of Maryland. 1999.

[43] Duong, J., Ct, M., Emptoz, H., and Suen, C. Extraction of text areas in printed document images. *In ACM Symposium on Document Engineering ,DocEng'01*, pages 157–165, 2001.

[44] Journet, N., Eglin, V., Ramel, J.-Y., and Mullot, R. Ancient printed documents indexation: a new approach. *In Pattern Recognition and Data Mining, Lectures Notes in Computer Science 3686*, pages 513 – 522, 2005.

[45] Journet, N., Mullot, R., Eglin, V., and Ramel, R. J. Analyse d'images de documents anciens:categorisation de contenus par approche texture. *In CIFED, Colloque International sur l'Ecrit et le Document*, 2006.

[46] Shi, Z. and Govindaraju, V. Multi-scale techniques for document page segmentation. In *proc. of Eighth International Conference on Document Analysis and Recognition (ICDAR)*, pages 1020–1024, 2005.

[47] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer,* 7(25): 10–22, 1992.

[48] Mitchell, P. E. and Yan, H. Newspaper document analysis featuring connected line segmentation. In *proc. of the Sixth International Conference on Document Analysis and Recognition,* pages 1181 – 1185. 2001.

[49] Faure, C. and Vincent, N. Simultaneous detection of vertical and horizontal text lines based on perceptual organization. *In 16th Document Recognition and Retrieval Conference*, 2009.

[50] Wong, K. Y., Casey, R. G., and Wahi, F. M. Document analysis system. *IBM Journal of Research Development*, 26:647 – 656, 1982.

[51] Tan, C. L. and Zhang, Z. Text block segmentation using pyramid structure. *In proc. of SPIE, the International Society for Optical Engineering,* 2001.

[52] Bukhari, S. S., Shafait, F., and Breuel, T. M. Segmentation of curled textlines using active contours. *In The Eighth IAPR Workshop on Document Analysis Systems*. 2008.

[53] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. On Pattern Analysis and Machine Intelligence,* 15(11): 1162–1173, 1993.

[54] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding,* 70(3): 370–382, 1998.

[55] http://www.ics.uci.edu/~dramanan/teaching/cs117_spring11/lec/morphology.pdf 2013.

[56] Ramel, J. and Leriche, S. Segmentation et analyse interactive de documents anciens imprimes. *In Traitement du Signal (TS)*, pages 209 – 222, 2005.

[57] G. N. Sarage and S. S. Jambhorkar. Noise Removal from Mammographic Image based on Mean and Median Filtering Technique. *International Journal of Advanced Research in Computer Science*, 2(4): 498-500, 2011.

[58] Edmond J. Keller. Microsoft ® Encarta ®, © 1993-2008 Microsoft Corporation, 2009.

[59] Coulmas, F. *Writing systems of the world*. Oxford, England, 1989.

[60] Bender, M.L, Sydney W. Head, and Roger Cowley. *The Ethiopian writing System: Language in Ethiopia*, Oxford University Press, London, 1976

[61]Phillips, I. and A. Chhabra. Empirical Performance Evaluation of Graphics Recognition Systems. *IEEE Trans. of Patt. Analysis and Machine Intell,* 21: 849-870, 1999.

[62] http://en.wikipedia.org/wiki/Languages, 2013.

[63] Hudson G. *Aspects of the History of Ethiopic Writing*. Bulletin of the Institute of Ethiopian Studies 25, pages 1-12, 2001.

[64] Dereje, T. *Optical Character Recognition of Typewritten Amharic Character*. M.Sc. Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia, 1999.

[65] Wondwossen M. *Optical Character Recognition for Special Type of Handwritten Amharic Text ("Yekum Tsifet"): Neural Network Approach*. M.Sc. Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia, 2004.

[66] Nigussie, T. *Handwritten Amharic Text Recognition Applied to the Processing of Bank Checks*. M.Sc. Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia, 2000.

[67] Thomas, B. The Ethiopic Writing System: A Profile. *Journal of the Simplified Spelling Society*, J19: 30-36, 1995.

[68] Seid, H. and Gamback, B. A Speaker Independent Continuous Speech Recognizer for Amharic. In *proc. of Interspeech*, 2005.

[69] Sameer, R. *Implementation of Watershed Based Image Segmentation Algorithm in FPGA*. MSc. Thesis, Universität Stuttgart, Stutgart, 2011.

[70] http://en.wikipedia.org/wiki/Connected-component_labeling, 2013.

[71] SUNG Siu Hang Aaron. *Comic Panel Extractor and Viewer for iPhone*. Final Year Project, The Chinese University of Hong Kong, Hong Kong, China, 2011.

[72] C.D. Manning, P. Raghavan and H. Schutze. *Introduction to information retrieval*. Cambridge university press, USA, 2008.

[73] Vihay V.Raghavan and Gwang S.Jung. A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. *ACM Transactions on Information Systems*, 7(3): 205-229, 1989.

[74] David M, et al. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms.

[75] Satadal S, et al. A Hough Transform based Technique for Text Segmentation. *Journal of Computing ISSN*, 2(2): 2151-9617, 2010.

[76] Skarbek, W., Koschan, A., Bericht, T., Veroffentlichung, Z., and Klette. *Color Image Segmentation: A Survey,* 1994.

[77] Yatharth, S. *Algorithms for Image Segmentation*. M.Sc. Thesis, Birla Institute of Technology and Science, 2006.

[78] R Cattoni et al. Geometric Layout Analysis Techniques for Document Image Understanding: a Review. *ICT-IRST Trento, Italy*, 1998.

[79] Md. Shakowat Zaman Sarker, Tan Wooi Haw and Rajasvaran Logeswaran, Morphological based technique for image segmentation, *International Journal of Information Technology*, 14(1).

[80] Manisha Bhagwat, R. K. Krishna and Vivek Pise. Simplified Watershed Transformation. *International Journal of Computer Science and Communication*, 1(1): 175-177, 2010.

[81] http://www.powershow.com/view/b22c3-MzIwO/Document_Image_Retrieval_powerpoint_ppt_presentation

Appendix I: Amharic Characters

	Ge'ez ä	*Ka'eb* u	*Salis* ï	*Rab'e* a	*Hamis* é	*Sadis* i	*Sab'e* o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ḥ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
q	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
t	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
h	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
n	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
a	አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
k	ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
w	ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
ạ	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
z	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
y	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
d	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
g	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ṭ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
p	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
ts	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ts	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
f	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
p	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

Appendix II: Sample Codes

Integration code

```
import ADIRsImageSegmentation.ImageSegmentationClass;
import com.mathworks.toolbox.javabuilder.MWArray;
import com.mathworks.toolbox.javabuilder.MWNumericArray;
public class ImageSegmentation
{
	public void ImageSegmentation()	// Default Constructor
	{
	MWNumericArray n = null; // Stores input value
	Object[] result = null; // Stores the result
	ImageSegmentationClass ConComp = null;
		// Stores ImageSegmentation instance
		try
		{
			/* Create new ImageSegmentationClass object */
			ConComp = new ImageSegmentationClass();
			/* Call the ImageSegmentationClass matlab file */
			ConComp.PageSegmentation();
			System.out.println("Image Segmentation Done!!.");
		}
		catch (Exception e)
		{
			System.out.println("Exception: " + e.toString());
		}
		finally
		{
			/* Free native resources */
			MWArray.disposeArray(n);
			MWArray.disposeArray(result);
			if (ConComp != null)
			{
				ConComp.dispose();
			}
		}
	}
}
```

Function to remove small components

```
function [bw,med] = RemoveSmall(bw,a,h,w)

[cc,num] = ConnectedComp(bw)

s = regionprops(cc, 'Area', 'BoundingBox');

hi=[]
wi=[]
ar=[]

for a=1:num
    hi=[hi (s(a).BoundingBox(4))];
    wi=[w (s(a).BoundingBox(3))];
    ar=[ar s(a).Area]
end
med=median(hi)

for b=1: num
    %if ((s(b).BoundingBox(4))<10|(s(b).BoundingBox(3))<10)
        %bw(cc.PixelIdxList{b})=0;
    if ((s(b).Area)<(median(ar)/a) |
(s(b).BoundingBox(4))<(median(hi)/h)|(s(b).BoundingBox(3))<(median(wi)/w))
        bw(cc.PixelIdxList{b})=0;
    end
end
```

Function to remove large components

```
function [bw] = RemoveLarge(bw)
[cc,num] = ConnectedComp(~bw)

s = regionprops(cc, 'Area', 'BoundingBox');

hi=[]
ar=[]
x=[]
for a=1:num
   hi=[hi (s(a).BoundingBox(4))];
   ar=[ar s(a).Area]
   x=[x a]
end

for b=1: num
    if
((s(b).Area)>median(ar)*1.5&(s(b).BoundingBox(4))>median(hi)+7|(s(b).BoundingBox(4)
)>median(hi)+8)
        bw(cc.PixelIdxList{b})=1;
    end
end
```

Dilation function

```
function [dialatedIm] = Dilat(bw,tresh)
%Apply dilation using bwdist() with a given threshold
dialatedIm = bwdist(~bw) >= tresh;
```

Connected components function

```
function [cc,num] = ConnectedComp(bw)
%Extracting connected components to variable cc
%using 4 connictivity
cc = bwconncomp(bw,4)
%storing number of connected components
num=cc.NumObjects;
```

The code to create bounding boxes

```
function [bw] = BoundingBox(bw,bi)
cc = bwconncomp(bw,4)
num=cc.NumObjects;
cord=[];
temp=[]

s = regionprops(cc, 'Area', 'BoundingBox');

figure, imshow(bi);
for i=1:num
rectangle('Position',s(i).BoundingBox,'EdgeColor','r','LineWidth',1);
end
for i=1:num
    temp=[s(i).BoundingBox(1)-0.5,s(i).BoundingBox(2)-0.5]
    cord=[cord ;temp]
end
```

Declaration

I declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source materials used for this thesis have been duly acknowledged.

_____

Gedion Assefa

June 2013

This thesis has been submitted for examination with my approval as university advisor.

_____

Million Meshesha (Ph.D)

June 2013