Addis Ababa University
College of Natural Sciences
School of Information Science

Segmentation of Real Life Amharic Documents for Improving Recognition

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

Berhanu Sahle

Advisor

Dereje Teferi (Ph.D)

June 2015

Addis Ababa University
College of Natural Sciences
School of Information Science

Segmentation of Real Life Amharic Documents for Improving Recognition

By

Berhanu Sahle

Name and signature of members of examining Board

Chairperson:	_____	_____	_____
	Name (typed)	Signature	Date
Advisor:	_____	_____	_____
	Name (typed)	Signature	Date
Examiner:	_____	_____	_____
	Name (typed)	Signature	Date
Examiner:	_____	_____	_____
	Name (typed)	Signature	Date

DECLARATION

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

Berhanu Sahle

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

Dereje Teferi (Ph.D)

DEDICATION

To my Loving Mother and Father,

Who sacrifices everything they have for my success!

ACKNOWLEDGEMENTS

First of all, I praise the almighty GOD and his mother ST. MARY for enabling me to complete the masters program at AAU. Thank you Holy Mother of GOD! Thanks again and again for giving me the strength, courage and the opportunity to do this.

I would also like to express my deepest gratitude to my advisor Dr. Dereje Teferi for being on my side from the start. He provided me with significant ideas and research directions since the begining of this research. Thank You So Much!

Another special thanks goes to Mr. Michael Abebaw who had graduated in 2014 from AAU. He had been generous in giving me his constructive ideas, valuable support and materials that I needed.

Finally, I would like to express my deep regards for my loving family mom, dad, and my sisters (Mekdi, Meski, and Lidu). Thank you all for your encouragement, love and support you gave me all those years. This is the result of your effort. Also, my utmost gratitiude is extended to my love Dr. Bethlehem T., and good friends especially Abel Y., Amanuel W., Andualem A., Dawit B., Dejene A., Ephrem Y., Ephrem N., and Tsegazeab G., whose love, friendship and ideas have supported me in one way or another in the preparation and completion of the study.

ABSTRACT

A huge amount of paper based documents with valuable information is available in churches, libraries, caves, governmental and private institutions in a printed, typewritten and handwritten format. To enable those documents accessible and searchable, Optical Character Recognition (OCR) systems play a vital role by converting them into their digital format. Some researchers attempted to develop Amharic OCR systems. However, OCR systems are not yet applicable for real life document images that contain column blocks, graphics, tables, lines, logos and other shapes. Moreover, the effectiveness of the system is highly dependent on the text segmentation output. This study attempts to explore an effective page and text segmentation method to improve the applicability and performance of Amharic OCR for real life documents.

Accordingly, a skew correction and page segmentation algorithms based on Hough Transform, Morphological Dilation, and Connected Component (CC) Analysis are tested, and 90.47%, 92.31%, 96.67% and 71.43% accuracy is obtained for detecting tables, graphics, column blocks and titles individually. Three noise filtering and two binarization techniques are tested and wiener coupled sauvola found to perform best. Text segmentation methods based on projection profile, morphological dilation and CC Analysis are experimented on four noise levels (i.e. low, medium, high and very-high) documents. Projection profile coupled vertical dilation performs best by scoring 100% accuracy to segment text lines in low and medium noise levels. An image smoothing based method is proposed and 99.18% accuracy is registered to extract lines from ink-bleeded documents. Vertical projection profile method is applied to extract words and 99.23%, 96.26%, 87.12% and 54.80% accuracy is registered for each noise levels respectively.

A new method based on CC Analysis is introduced to segment overlapping characters, and besides to detect and split connected characters. An accuracy of 87.61% and 82.29% is obtained for low and medium noise levels and 50.64% for high and very high noise levels. By integrating it with the Amharic OCR system, recognition accuracy rate of 79.13% and 59.07% are registered for the proposed and vertical projection profile method respectively, which is a promising result. However, since the developed character segmentation technique fails to segment characters with discontinuity, and detects long characters as connected character for real life documents, there is a need to explore noise tolerant segmentation methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABBREVIATIONS

ANN	Artificial Neural Network
ASCII	American Standard Code for Information Interchange
BMP	Bitmap
CC	Connected Component
CPU	Central Processing Unit
DIR	Document Image Retrieval
DPI	Dots Per Inch
GB	Giga Bytes
GHz	Giga Hertz
HMM	Hidden Markov Model
HRLS	Horizontal Run Length Smoothing
KPCA	Kernel Principal Component Analysis
LDF	Linear Discriminant Function
LVQ	Learning Vector Quantization
MSE	Mean Squared Error
OCR	Optical Character Recognition
PC	Personal Computer
PCA	Principal Component Analysis
PSNR	Peak Signal to Noise Ratio
QDF	Quadratic Discriminant Function
RAM	Random Access Memory
RDA	Regularized Discriminant Analysis
RGB	Red Green Blue
ROI	Region of Interest
SVM	Support Vector Machine
VQ	Vector Quantization

CHAPTER ONE
INTRODUCTION

1.1 Background

The development of powerful computers that can process large amount of data speedily as well as the invention of internet resulted rapid growth of digital technologies. Accordingly every activity of humans nowadays becomes very intertwined and suffused with them due to their out of imagination performance [17] [33]. In the ancient periods, communication and codification of information and knowledge were performed by writing, speaking, drawing, gesture, signs and symbols. The modern ways of codification includes recording information in different multimedia formats such as text, image, audio, video and animation. Over centuries, paper documents have been a principal instrument to make the progress of the humankind permanent and most information is still recorded, stored and distributed in a paper format [13].

We humans learn reading and writing skills through education and our ability grows to read most texts such as those printed in various fonts and styles, handwritten neatly or sloppily, characters with missing parts and misspelled words as the learning time increases [17]. This recognition process occurs thousands of times in a day using our capturing mechanism of an eye and interpreting or recognizing characters using our brain's experience. In computer world, this process is stimulated to Optical Character Recognition [35] [38] [43].

Optical Character Recognition (OCR) is a process that allows printed, typewritten and also handwritten text to be recognized optically and converted into machine readable code so that it can be accepted by a computer for further processing [27]. OCR is a field of research in pattern recognition, artificial intelligence and computer vision that includes hardware like optical scanner for converting documents to image, which are going to be an input to some processes in order to be converted to digital form [16] [22].

OCR is comparatively old in the field of pattern recognition and many studies were made since 1929. Tausheck is the first to get his patent of OCR in Germany and Handel did the same in US by 1933 [1]. It has been a dream that machines could read characters and numerals until the age of computer in the 1950's [50]. Eikvil [23] presented the generations of OCR into four based on

the versatility, robustness and efficiency of the system. The first is between 1870, where Carey invented a retina scanner to the 1950. The second is from 1965 to 1975, machine printed and hand-written characters were recognized and began to be used in letter sorting on postal services. The third is from 1975 to 1985 where people try to apply OCR in poor quality documents as well as in unconstrained handwritten character sets and making them less expensive and more efficient. The current generation is characterized by recognizing complex documents that contains mixed texts, graphics, and mathematical symbols in low-quality noisy documents [17] [23] [50].

OCR systems are categorized as either online or offline recognition systems based on their method of data acquisition. Online recognition systems use digitizers for direct capture of data through writing with order of strokes, pen up and down information whereas the offline recognition systems takes input from optical scanners or digital cameras [45].

OCR includes some basic steps which are classified in different ways on various literatures to perform the recognition process. Some of these steps are document digitization, preprocessing of the digitized document images by applying techniques such as binarization, noise removal, thinning, underline detection and removal and others, segmentation, feature extraction, classification and post-processing tasks in a ranked order [35] [43].

Digitization is the process of producing document images by converting paper based documents into digital form through scanning, using camera or by on-screen pen up and pen down information. For scanning, most OCR systems use a resolution that range between 300 dpi to 1000 dpi for better accuracy in text extraction. Use of high resolution while scanning real world historic documents that are very noisy or degraded is necessary. It is the first phase in document image recognition systems [6] [21] [67].

Preprocessing is the most important and critical step which is very helpful to enhance the performance of recognition. It mainly focuses on correcting the deficiencies found on document image that might be introduced due to reasons such as data acquisition process, problem on capturing device, and age of document. It prepares data for the next subsequent activities [6].

Noises are defined as "*the random variation of brightness or color information in images produced by the sensor and circuitry of scanner or digital camera and anything that is irrelevant*

in digital images is considered as a noise"[13]. The preprocessing stage of recognition system is very helpful to make the pattern recognition problem simple through reduction of such noises, degradations and inconsistencies over document images without the loss of vital information. Some of the techniques employed under preprocessing include noise filtering (reduction), binarization, skew detection and correction, underline detection and removal, size normalization, thinning, etc. [6] [67].

After preprocessing, the document images need to be segmented to separate a set of figures, tables and other lines, text lines, words and characters from document image through a process of separation of an image into regions that contain pixel groups that are similar in value [66]. It occurs at two levels; the first level performs text/graphics separation and the second level performs segmentation of text lines, words and characters of the document image [45].

The next step is extracting features that can uniquely represent one character from another by a process known as feature extraction. It is responsible for extracting each of these features from the matrices of digitized characters so that the characters can easily be recognized by the classifier [17] [18] [63].

Classification is the decision part of the OCR systems that receive input from the output feature extraction phase which are the features extracted to produce the predicted ASCII representation of the input character. This module mainly performs two main tasks of *learning* and *testing phases*. Learning phase builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels. The classifier then creates a model based on the training data so that it will be used in testing and recognition process for the given input feature vectors to predict the ASCII representations [34].

After the classification task is performed and recognition is done, this phase will take the results as an input to further process to check and maintain errors. This is due to classification algorithms that are not perfect and always make mistakes because they are making predictions. Especially for the degraded documents and for alphabets that are very similar, misclassification of characters is always there. According to Eikvil [23], the use of spell or error checkers on the result of OCR systems can enhance the performance of recognition process by correcting errors made by classification stage [23].

Nowadays, the application of OCR becomes more important in various business and governmental areas such as library and office automation, bank check processing, as a reader for the visually impaired people, data entry from passport, postal automation and so many other applications [45]. It is common to find PC-based OCR systems that are commercially available for most languages in the world, and they become less expensive, faster and more reliable due to less expensive electronic components [67]. They can also be applied for the purpose of recognition based document image retrieval. However, the performance of the system relies heavily on the quality of the scanned images [13].

Most OCR systems are developed to work with Latin-based scripts and they are now used for practical problem solving activities nevertheless few papers are available on the indigenous scripts of African languages [45].

1.2 Statement of the Problem and Justification

Ethiopia is a country at the Horn of Africa that has its own writing system and calendar. There are about 100 languages that can be classified into four groups namely; Semitic, Cushitic, Omotic, and Nilotic. Among those various languages, Amharic is one of the Semitic languages that use Ethiopic scripts for writing purpose. It is the most dominant language in Ethiopia spoken by roughly 30% of the population as a mother tongue and also additionally 20% use it as a second language and totally, half of the population uses Amharic language which makes it an official language of the country and medium of communication and working language in most of the regional states [13] [78].

Amharic is one of the languages in Africa having its own indigenous scripts and writing systems. It was spoken since 13th century and started to use for writing purpose in 19th century. Due to the long history of Amharic writing system, huge amount of information is available inside churches, caves, governmental and private institutions including information centers, libraries, museums, etc. in printed, typewritten and hand-written format [67]. Therefore, there is a need to bridge the gap that exists between the rich information formats but yet inaccessible, not easily reachable and non-searchable ones into their corresponding computer representation with high accuracy.

The studies of Amharic OCR have been made starting from the first attempt made by Worku [66]. He investigated the application of OCR techniques for Amharic text. After that Ermias [24] further worked on recognition of formatted printed Amharic texts. Dereje [21] studied typewritten Amharic documents recognition. On the same year, recognition of printed characters using Artificial Neural Network (ANN) as feature extractor and classifier was studied by Berhanu [11]. Million [44] studied to generalize previously adopted algorithms for printed characters with different fonts and sizes. Yaregal [67], Yaregal and Bigun [68]-[79] also conducted studies on Amharic recognition for printed and handwritten characters of Ethiopic scripts by extracting structural/topological patterns of characters (primitives). Further, Million and Jawahar [46] [48], Abay [1] and Michael [43] worked on recognition of real-life documents.

On the recent experiment made by Michael [43], he faced erroneous segmentation of characters which has a direct impact on the effectiveness of recognition. He applied vertical projection profile to segment words and characters. However, characters with discontinuity in their body, overlapping pixels, touched pixels, and adjacent pixels with neighboring characters are failed to segment successfully. Figure 1.1 shows segmentation errors observed in previous investigation of Michael [43].

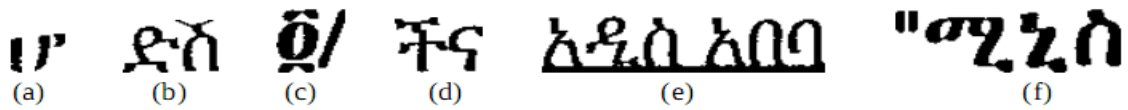


Figure 1.1: Sample segmentation errors observed during previous investigation of Michael [43]
(a) Discontinuity in character "ሀ" produced two individual characters, (b) Overlapped characters considered as one character , (c) adjacent black pixels between consecutive characters, (d) Touched pixels between characters, (e) Underlined characters, (f) Single pixel variation from threshold value

Yaregal [70] studied Multifont size-resilient recognition system for Ethiopic script by extracting structural building blocks (primitives) of each character. He suggested the recognition accuracy can still be improved by dedicating more efforts on character segmentation [70].

According to Million [45], one of the difficulties in Amharic document image recognition and retrieval is degradation. It is a reason for the failure of segmentation and subsequent phases of OCR. Denoising is often necessary and first step to be taken before the document image passes through the next stages. Even though, a number of attempts were made before, it is still an active

research area because observed results show that there is a loss of vital information from those document images.

Previous studies were tested on a dataset from real life Amharic documents such as bible, newspaper, regulation and fictions with a different level of degradations such as dusty noise, large ink-blobs, vertical cuts and ink from facing pages. Degradations can be caused by scanning devices and transmission media errors, aging, photocopying, faxed documents, problems in data acquisition, and inferring natural phenomena [45]. Figure 1.2 shows an example of real life Amharic document image that contain ink-bleeding noise.

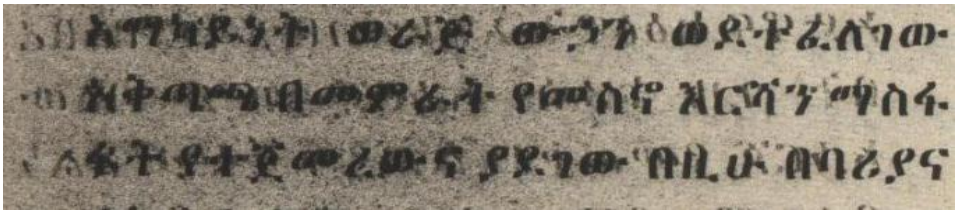


Figure 1.2: Sample Document Image Taken From Biniyam [13] with Ink-Bleeding Noise

As compared to Latin scripts, upper-case and lower-case letters are absent in Amharic writing system. On the other hand, like English, the writing mode is from left to right and top-to-bottom. Words are separated with blank space nowadays but historic documents of Ethiopic scripts has used two-dots to separate words and sentences end with four-dots and paragraphs with recognized horizontal space [47]. Figure 1.3 presents one of the reasons (i.e. two-dots) that make segmentation difficult for historic documents.

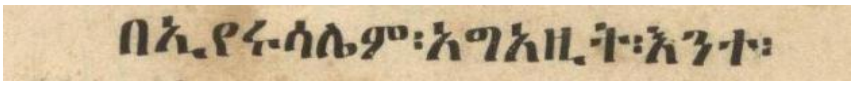


Figure 1.3: Two dots that are used to separate words in historic documents

Gedion [26] conducted a study of page segmentation method to segment tables, graphics or pictures, text lines and words from the document image collections for DIR system using different techniques. The study of page segmentation is crucial because real-life document images usually contain both text and non-text elements. However, the problem faced by Gedion includes; some parts of a text were considered as a table or line and vise-versa, and also some parts of the graphics were considered as a text.

Previous studies were tested on limited datasets. However, the main concern of this study is to explore better page and text segmentation techniques for improving the performance of Amharic OCR as well as adopting better preprocessing methods that can work well on different levels of degradations and which can also work better with the selected segmentation techniques.

To fill the above mentioned gaps, this study attempts to answer the following research questions:

- What are the special features of real-life Amharic document images and what kind of degradations are observed on such documents?
- What suitable page segmentation technique can be applied for successfully segmenting table and other lines, text/graphics, column blocks, titles and other shapes found on real life document images?
- What suitable image processing techniques such as noise detection and removal need to be integrated to improve text segmentation?
- What suitable text segmentation techniques need to be integrated to improve the performance of Amharic document image recognition?
- How much improvement is registered on the performance of Amharic OCR system?

1.3 Objectives of the Study

1.3.1 General Objective

The main objective of this research is to integrate effective page and text segmentation techniques for improving the effectiveness of Amharic OCR on real-life document images.

1.3.2 Specific Objective

To meet the general objectives, the following specific objectives are drawn.

- To review different international and local researches, on document image recognition for real life documents for understanding of the area, approaches, algorithms and to know what was done before for non-Latin script recognition and understand the challenges.
- To study the unique characteristics of degraded Amharic document images.
- To collect training and test datasets from real-life historic documents of Amharic language that contains different level of degradations.

- To explore better page segmentation technique that can separate the text area from the graphics, tables and also to identify column blocks and titles.
- To adopt best preprocessing techniques that can help to improve text segmentation.
- To study suitable text segmentation techniques that can extract text lines, words and characters from real-life documents and integrate it with the previously developed Amharic OCR system.
- To evaluate the performance of the recognition system.

1.4 Scope of the study

By aiming the improvement of the performance of Amharic OCR systems for real-life documents developed so far by previous researchers, this research experiments some available preprocessing and segmentation techniques. Among many available algorithms, selected techniques are studied and tested on collected real-life document images with different level of degradations; experimenting and integrating the selected preprocessing and segmentation techniques to the Amharic recognition system is the main focus of the study. The test datasets are originally collected by previous researcher Biniyam [13] who made his study on the document image retrieval system and also some new real life document images that contain graphics, pictures, tables, columns are collected and merged with the dataset to experiment page segmentation methods. All the documents are taken from sample printed historic documents such as old books, newspapers, bible, and magazines with different level of noises.

However, this study is a continuation of previous work and it doesn't include the segmentation of handwritten and typewritten document images. Also due to time limitations, the study mainly focuses on the page and text segmentation techniques for Amharic OCR system.

1.5 Methodology of the Study

This research follows an experimental research; it is also named as empirical research that relies on experience or observation alone, often without due regard for system development and theory. It is data-based research, coming up with conclusions which are capable of being verified by observation or experiment [39]. It involves data preparation, system development and evaluation.

Therefore the following methods and techniques are used to undertake the research work for the achievement of specified objectives and to answer the research questions of the study.

1.5.1 Literature review

In order to have a deep understanding about the OCR system techniques, related literatures from different sources such as books, journal articles, conference papers and internet sources such as educational websites are reviewed. Also both past and recent studies of works on Amharic and other similar non-Latin languages are reviewed to have a better background on the better performing algorithms and techniques. Since this research is supposed to be a continuation of the previous studies of Amharic OCR and need to be integrated with them; local researches are given more emphasis.

1.5.2 Dataset Collection

For the purpose of training and testing, the datasets collected and organized by previous studies are collected and digitized. Mainly Amharic document images with different level of degradations that are collected by Biniyam [13] from previous study of Document Image Retrieval (DIR) for Amharic document images are used and some additional real life document images that contain graphics, pictures, tables, columns are collected and merged with the dataset to experiment page segmentation methods. The datasets are from various sources of newspapers, magazines, old books, and other historic documents. 26 of the document images are from Biniyam's dataset and 33 document images are the newly added document images; totally, 59 document images are gathered.

1.5.3 Implementation tools

Most previous studies use MATLAB® Image Processing ToolboxTM, due to availability of its rich libraries for image processing. One of the recent research made by Michael [43] also used MATLAB® Image Processing ToolboxTM and visual C# libraries. And also some other studies used Visual C++ and as well as WEKA machine learning tool.

Visual C# libraries and MATLAB® Image Processing ToolboxTM are integrated and used in this research because the researcher is familiar with this language and MATLAB is better for image processing and also for easy integration with the previous study.

1.5.4 Performance Evaluation

To measure the performance of noise removal and other preprocessing techniques, Mean Squared Error (MSE) and Peak Signal to noise ratio (PSNR) are used. MSE is a measure of an average of the squares of the errors, which is the difference between the estimator and what is estimated. It is a risk function corresponding to the expected value of the squared error loss. PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale [64].

For measuring the performance of both page and text segmentation algorithms, there are various techniques available. For this study, the performance is measured using the manual counting of the expected correct and wrong segmentation to calculate the accuracy percentage as used in Michael's [43] study.

After the integration of recognition algorithm with the proposed techniques, the accuracy of the OCR system are measured by a common and widely used methods such as recognition rate, error rate and rejection rate of test results from both testing and training sets. Eikvil [23] discussed that recognition rate measures the proportion of correctly classified characters from the total characters and conversely, error rate measures the proportion of characters erroneously classified. Rejection rate measures the proportion of characters which the system was unable to recognize. Michael [43] used recognition rate and error rate and this study also uses the same to evaluate the performance of the system.

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 [48]. There is bulk of historical printed, typewritten, and handwritten documents available that needs to be digitized and accessible via the Internet and digital libraries [45]. Manual conversion process of these documents which is by typing; is tedious, error prone and time taking. OCR systems can provide an automatic transformation into computer representation of these documents.

Libraries, museums, churches and other information centers can benefit by using the system to share documents to many users easily and simultaneously. Governmental and non-governmental institutions can also make information on printed documents effortlessly accessible. Researchers on different disciplines such as: history, religion and sociology, politics, etc. are other beneficiaries of the system. Therefore, developing Amharic OCR system has several benefits for different parties.

Further than the above major contributions, OCR systems play bigger roles such as:

- To help us to make the computer representation (digital format) of physically available printed, typewritten or handwritten real life documents. This makes the valuable documents electronically available for future references.
- To facilitate quick digital search in contents.
- To save storage space.
- To allow document modification if necessary.
- To assist visually impaired people as a reading tool for scanned documents by combining it with other systems such as speech synthesizers.
- To enhances document accessibility, availability and portability in various devices such as smart phones.

1.7 Organization of the study

This thesis is organized into five chapters. The first chapter discusses the background of the study, statement of the problem, the general and specific objectives, methodologies used, and scope of the research.

In chapter two, literature review on the overview of OCR system and phases conducted under OCR systems specially those focusing on preprocessing and segmentation methods are reviewed. Moreover, a brief review of the history, development and characteristics of the Amharic writing system and different types of documents, local related works on document image recognition and the challenges in building Amharic OCR are reviewed.

In chapter three, the proposed architecture of Amharic OCR system, the selected image preprocessing and segmentation techniques are concisely explained. The evaluation measures that are used for measuring the performance of each algorithm are also discussed.

Chapter four emphasizes the experimentation of selected preprocessing and segmentation techniques, and experimental results used to confirm the validity of the proposed techniques are presented and integration to recognition phase and its results are also presented.

Finally, based on the findings of the study, conclusion and recommendations of the research are presented in chapter five.

CHAPTER TWO
LITERATURE REVIEW

The development of Optical Character Recognition (OCR) is motivated by the need to cope with the available massive flood of documents in real life such as books, newspapers, bank checks, commercial forms, government records, posted letters, credit card imprints and also other historic documents in churches, museums and various institutions. OCR is automatic reading of optically sensed document texts of human readable characters to machine readable codes such as ASCII or Unicode [67]. To perform the recognition process, OCR systems involve major steps classified in various ways on different literatures and examining them is critical for the research.

2.1 Overview of OCR System

One of the emerging applications due to the accelerated advancement in pattern recognition that are not only challenging but also computationally demanding applications are OCR systems. The main objective of OCR system is to recognize characters that are found on optically sensed human readable document images that contains handwritten, typewritten or printed text and convert them into machine readable codes and enable document images to be accessible and editable. It can be described as a mechanical or electronic conversion of scanned document images to their digital format. It is very important for various purposes such as indexing, searching, editing and reduction of storage size [16] [57].

OCR became an active field of research since mid-1950's. Today, it is one of the most successful applications of automatic pattern recognition and it plays an important role in this modern world where there are heterogeneous representation of text based information. Accordingly it has immense potential in future where we want to track and locate every piece of information being exchanged [22] [61].

Based on the type of data acquisition method they use, OCR systems are broadly categorized into online and offline OCR systems. Online recognition systems use digitizers for direct capture of data through writing with order of strokes, pen up and down information whereas the offline recognition systems takes input data from optical scanners or digital cameras in the form of document images. However, some researchers argue that OCR is only instance of offline

character recognition, where the system recognizes the fixed static shape of a character and should not be confused with on-line character recognition [1] [61].

OCR systems nowadays became an integral part of document scanners and many other applications such as postal processing, banking security, script recognition, passport authentication and language identification [57].

To perform the recognition process, OCR systems involves basic steps which are classified in different ways on various literatures. However, Million [45] mentioned that most of the designs of OCR systems follow a modification of the common architecture. The framework of OCR systems that categorizes some of the basic steps undertaken into three general basic phases as Document Scanning Phase, Recognition Phase and Verifying Phase as presented in figure 2.1 [57].

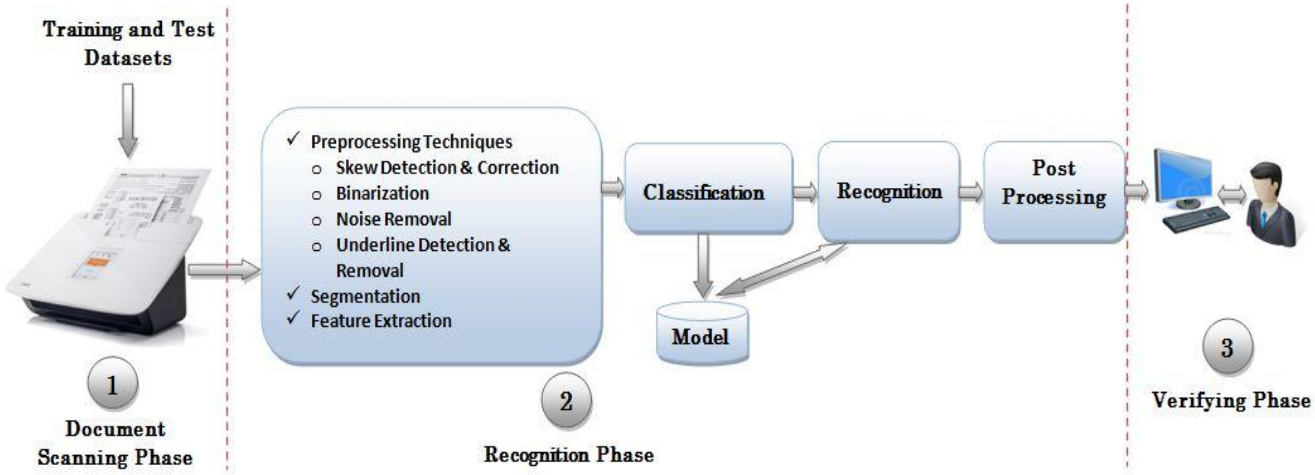


Figure 2.1 General Framework of OCR System as Adopted From [57]

2.1.1 Document Scanning (Digitization) Phase

It is described that OCR is a process of converting scanned images of documents with printed, typewritten or handwritten texts into computer editable format and this phase is the first phase that any OCR process starts and in some literatures it is termed as digitalization or image acquisition. It is responsible to produce document images by making conversion of paper based

documents into digital form by scanning, capturing using camera or through on-screen pen up and pen down information [1] [6] [67].

This phase is the stage where scanning device is used to scan the handwritten or printed documents in order to find the digitized form and the acquired images can be captured in different formats such as JPG, BMP, PNG, TIFF, etc. that range from a true color intake to binary form by adjusting the varying resolution of an image to make it in a way that a computer understands them as a matrix of pixels [1] [43]**.** The most common and dominantly used format for the case of OCR is bit map (.BMP).

One of the major concerns that must be dealt during this phase is the resolution of the scanned image measured in terms of Dots-Per-Inch (DPI) because it has a huge and direct impact on the performance of overall recognition process of OCR systems. DPI is a measure of spatial printing or video dot density; in particular the number of individual dots that can be placed in a line within a span of 1 inch (2.54 cm) in digital image [32]. It is a number of pixels the image spans in an inch of measure and higher resolution is preferred or needed for OCR systems because of the produced image must be better images in quality. Lower resolution tends to break thin lines, fill gaps that exist in characters, and also may cause noises on the document images.

A scanner's optical resolution is determined by how many pixels it can actually see. For example, a typical flatbed scanner will use a scanning head with 300 sensors per inch, so it can sample 300 dots per inch (dpi) in one direction. To scan in the other direction, it will move the scanning head along the page, stopping 300 times per inch, so it can scan 300 dpi in the other direction as well. This scanner would have an optical resolution of 300 x 300 dpi. Some manufacturers stop the scanning head more frequently as it moves down the page, so their machines have resolutions of 300 x 600 dpi or 300x1200 dpi [32].

However, using high scanning resolution also makes the produced image bigger in size which causes the wastage of disk space and slows down the scanning process and also other remaining processes in OCR system. The recent advances in scanner technology have made available resolution in the range of 600 DPI to more than 1200 DPI [1].

One of the online sources from ABBYY Technologies (https://www.abbyy-developers.eu) which are the known providers of OCR system recommends a resolution rate according to the font size of text document that is going to be scanned as follows.

> *"For regular texts written in English scripts using font sizes of 8 to 10 points, it is recommended to use 300 dpi resolutions and if the scans have a smaller resolution, for example 200 dpi, then the 10 point font will be too small and will contain broken pixels or "missing pixels" in the document image. Therefore for the smaller font text sizes (below 8 points), using 400-600 dpi resolution is recommended [2]".*

Therefore, we must note here that as a scanner resolution increases, the quality of document image also increases.

Dereje [21] explained most OCR systems use a resolution between 300 dpi and 1000 dpi to get better accuracy in text extraction and recognition. Studies show that using high resolution is necessary while scanning real world historic documents which are very noisy and degraded.

2.1.2 Recognition Phase

After the scanning of document images, the next basic and major step of OCR systems is recognition phase. It is the difficult phase which employs several sub-phases that are vital for the recognition or conversion of document images into their ASCII or UNICODE representation of characters. The sub-phases involved in recognition phase are *Pre-Processing, Segmentation, Feature extraction, Classification and Post Processing Techniques* [23].

2.1.2.1 Pre-Processing techniques

Scanned document often contain noises that mainly arises from printer, scanner, paper quality, age of document or due to other reasons [48]. The images collected by different type of sensors are generally contaminated by different types of noises and paper documents are very sensitive to degradation of integrity. Therefore, before manipulating the information in the image, preprocessing tasks must be conducted on the scanned image to improve its quality [13].

The main goal of this phase is to enhance the performance of recognition process through increasing the accuracy and performance of next steps such as segmentation of pages, text lines, words or characters through performing a serious of operations during preprocessing stages to

organize the information and correct deficiencies in document images. The operations includes processes such as detecting skew angle and correct it, detecting text area, fixing some of the noises and degradations of an image, and some other activities are performed [6] [53] [56].

Preprocessing tasks aims that making the pattern recognition problem simple. Million [48] noted that because of the level of degradations in digitized real life document images such as magazines, books and newspaper, there is a need to apply noise filters so as to reduce the effect of degradation during the recognition process [48]. So, it is helpful to reduce noises, degradation and inconsistencies over the document image without the loss of vital information.

Algorithmic steps involved in preprocessing phase are divided into two based on their necessities in every OCR systems as *mandatory* and *optional steps*. Mandatory steps are the one that are very important and every OCR system should incorporate them. For example, binarization could be one of them. On the other hand, the optional steps can be taken or left based on the type of document or problem domain they are applied. One example of these steps include noise detection and removal, underline detection & removal, size normalization, skew detection, slant removal, thinning, filtering, contour smoothing, document restoration etc. [55] as cited by [43].

Real life document images are not free from degradations of different types and levels, therefore it is very important to include optional steps like noise detection and removal. According to Million [45], degradation of printed texts in documents images may come from different sources and he mentioned that it mostly occur due to the quality of paper used and/or ink drops from printers, photocopy or fax machines, and scanner quality. Generally he identified and categorizes noises in the document images into four groups as salt and pepper, cuts, blobs and erosion. Figure 2.2 presents these noise types in Amharic document images [45].

- **Salt and Pepper Noise:** This type of Noise is caused by errors in data transmission and scanning of documents. It is a common form of noise observed in real life document images and it is distributed all over the image flipping white pixels to black (i.e. pepper) if it is black background and black pixels into white (i.e. salt) if it is a foreground.
- **Cuts and Breaks:** such type of degradation occurs due to the paper quality, folding of paper and print font quality. It corrupts the part of document image by reversing black pixels into white and it creates the problem like breaking the continuity on the shape of characters.

- **Blobs:** occurs due to large ink drops within the document image during printing, faxing or photocopying process. It is a flipping of pixel values into black and the existence of this noise merge or separate character components.
- **Erosion of Boundary Pixels:** Such kinds of degradations are caused by the erosion of boundary pixels that can affect the image through changing either black pixel to white or vice versa. It is mostly happened by the imperfections in document scanning and mostly it is observed at the boundary of the image.

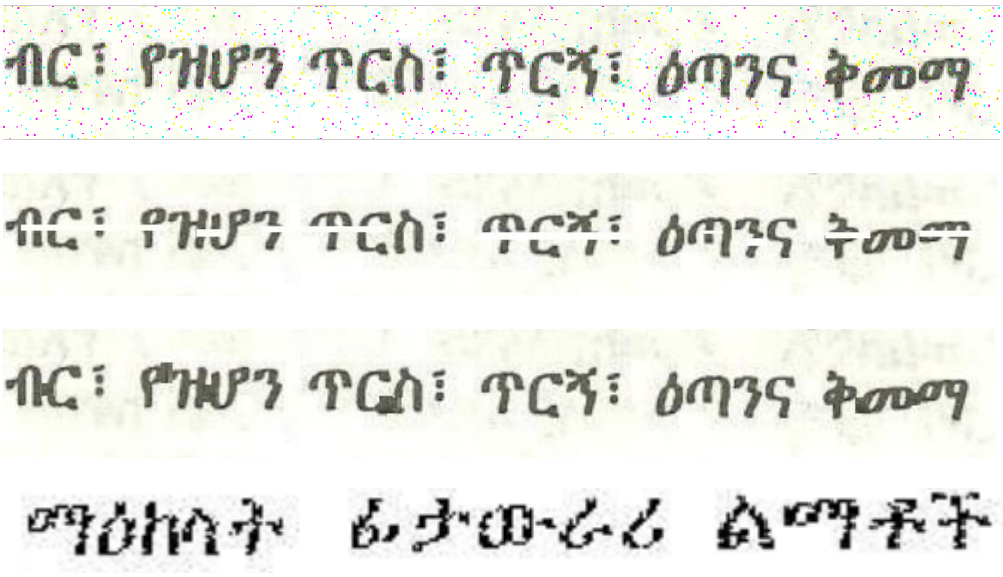


Figure 2.2: Sample Amharic word images with Salt-and-Pepper, Cuts and Breaks, Blobs and Erosion of boundary pixels (shown from top to bottom)

Moreover, based on the classification presented in [13], image noises are classified as the physical noise, digitization noise, filtering noise and storage or transmission noise. **Physical noises** are noises that are related with the physical readability or integrity of original information in the document and it is classified as *internal noise* that includes paper aging, paper texture, carbon copy effect, scratches, cracks and inadequate printing whereas *external noises* including folding marks, filing and staple punching, stain, thorn-off regions, worm holes, readers annotations, and highlighting, physical blur and sun burn. **Digitization noises** are introduced by the digitization process. **Filtering noises** are suitable manipulation of the digital file may degrade the information that exists in the digital format of the document instead of increasing it. **Storage/Transmission noises** are occurred due to the error or inefficiency in storage algorithms or from the network transmission errors [13].

Such types of noises are one of the factors that affect the accuracy of other stages of recognition which have a huge impact on the result text recognized [50]. Once the image is scanned and we identified the noise, noise removal or filtering (denoising) algorithms need to be applied on it in order to remove noises detected on the document image without losing vital information because the noisy images are obscured and the image details are lost. Specially to provide public access to historical and ancient document image collections, they must be preprocessed first to remove background noises and become more legible [13].

Noise removal is the process of removing noise from a signal or image which is always present in digital images. It can be created at the same time when image is captured and it is a part of image data recorded by the imaging device. The most important reason for noise reduction is to obtain easy way of recognition where extraneous features will otherwise cause subsequent errors in recognition [13].

There are several noise removal algorithms available for filtering noises on document images. Motwani [51] presented classification of image denoising methods (figure 2.3) and generally, he classified them as *Spatial Domain Filtering* and *Transform Domain Filtering*. Spatial domain filtering is also classified into *Linear* and *Non-linear* methods. Linear methods contain *Mean* and *Weiner Filters* whereas non-linear methods contain *Median* and *Weighted Median Filters*. On the other hand transform domain filtering is classified as *Data Adaptive Transform* and *Non-Data Adaptive Transform* and several algorithms are available under those methods which can be further referred from Motwani [51] and the following figure show the classification of noise removal methods that has been discussed so far [51].

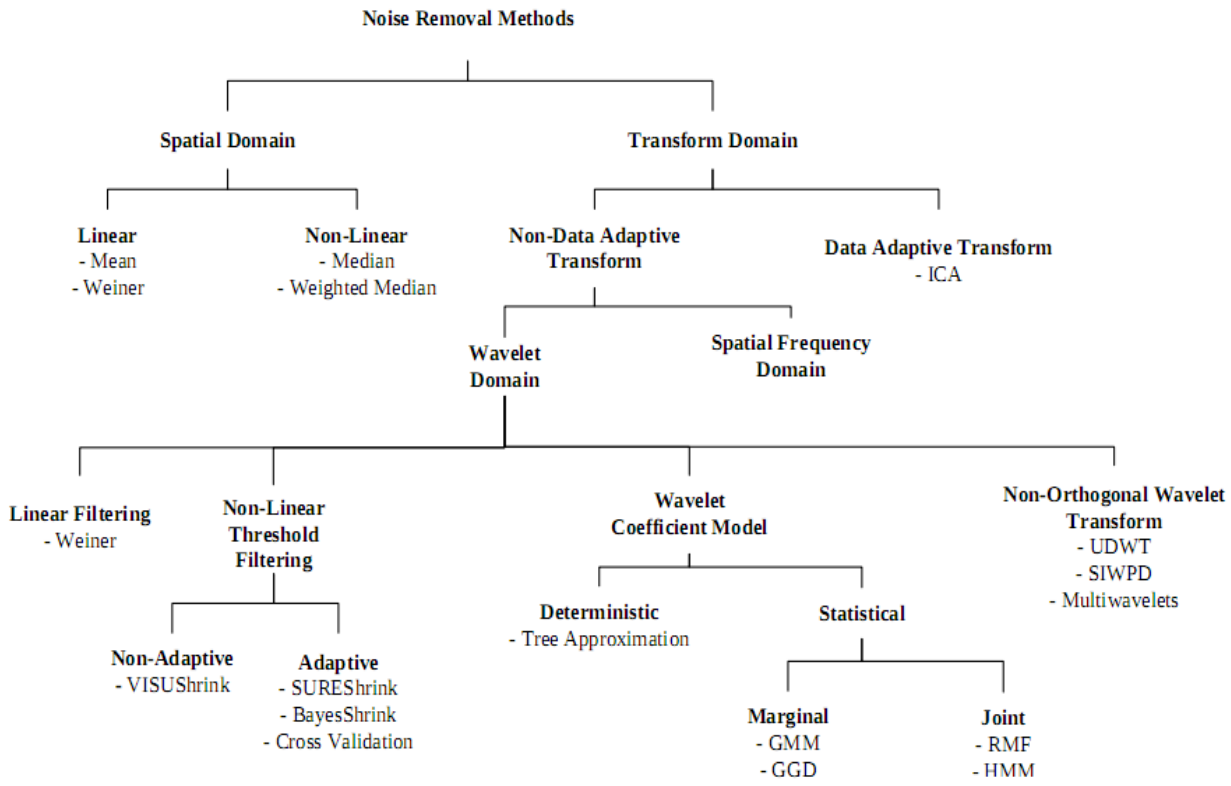


Figure 2.3: Classification of Noise removal methods as presented by Motwani [51]

In preprocessing task, there are also other operations that must be done based on the problem domain on hand. For example, if we need to detect and remove the underlines in our document image, we have to apply underline detection and removal algorithms to the preprocessed image because it will cause an error on other steps of recognition [24].

Another concept that is discussed by Dereje [21] is about image restoration techniques. He noted that, the first step in a preprocessing stage of the OCR system is the reduction of noises to enable other stages of OCR system performs well. During such process, there is a drawback to the document image which is a loss of vital information from the character image. So there are different image restoration techniques that we can apply to enhance or fix the corrupted or lost data due to the degradation of document image. This will help to enhance the performance of OCR system on recognition [21].

There are also other preprocessing tasks like skew detection and correction, thinning, size and normalization methods that are performed on the preprocessing stages of OCR system and for further reading about these and many other tasks of preprocessing refer [6] and [67].

2.1.2.2 Segmentation

Segmentation is a process of separation of an image into regions that contain pixel groups that are similar in value [15] [66]. It is also explained as symbolization or extraction of characters from pixel array [44]. It is considered as the most important part of recognition system because of the direct dependency of correct recognition on correct segmentation of characters. It is applied after the image filtering and other preprocessing tasks are done. Million [45] explained that segmentation occurs at two levels. On the first level; text/graphics, columns, tables and other parts are separated named as *page segmentation* and on the second level; text lines, words and characters in the image are located known as *text segmentation*.

Gedion [26] discussed that application of page segmentation algorithms to document images to separate text and non-text components is essential preprocessing step before other OCR operations including text segmentation. He also noted that there are limited numbers of shapes of text characters but the shapes of non-text elements are unlimited. Therefore OCR engines treat both text and non-text components differently, such that they only recognize text components and then arrange recognized text and also the graphics of non-text components in an output document using layout information [26].

After the text/graphics segmentation of document image is performed, the next stage is extraction of lines, words and characters from the document image known as text segmentation. OCR systems can jumps to any of the three parts of text segmentation but for proper organization of recognized text, it is important to segment lines, words and characters respectively [53]. The correctness in each stage of text segmentation assures the efficiency on the result of recognition.

Various literatures categorize image segmentation techniques in different ways. Text image segmentation algorithms are categorized in different ways by different scholars and traditionally, they are divided into three main group as *top-down, bottom-up* and *hybrid approaches* [25].

Top-down approaches perform segmentation on the document image starting from the entire image to smaller regions recursively. Most of well-known top-down methods are *Projection Profile Methods, Histogram Analysis* and *Space Transforms* (Fourier Transform, Hough Transform, etc.) and the methods are explained in detail in [25].

Bottom-up approaches are the reverse of top-down method where they start segmentation with the smallest element of document image that are pixels and merge them recursively in connected regions or components and then in larger structures. Some of the methods used here are *Connected Component (CC) Analysis, Region-Growing Methods, Run Length Smoothing, Neural Networks* and *Active Contours.*

Hybrid approaches are methods that combine and make use of both bottom-up and top-down approaches; For example, connected component analysis for shape information and block separation for background block map. They work very well for major text/graphic segmentation in real life documents but not for a very fine level segmentation of words and their individual characters in historical books [26]. Many other categorizations and algorithms are presented by various scholars and [36] can be referred for further reading.

According to Dereje [21], segmenting a character from a degraded real life document image is a trouble for OCR systems. The major problems faced errors on text segmentation were; the overlapping features of Amharic scripts, broken characters due to the degradation of printed real life documents; adjacent black pixels between consecutive characters and single pixel variation causes different characters segmented as one; inefficiency on underline detection and removal and also other preprocessing techniques and connected characters produced some difficulties on the result of character segmentation [43] [54].

2.1.2.3 Feature extraction

Feature extraction is the process of extracting relevant features from the segmented character images to form a feature vectors. The result of feature extraction is a unique representation of characters that can be used by classifiers to recognize the input unit with target output unit [43]. Therefore efficient extraction of unique features character images makes the classifier works efficiently since it makes easier to classify different classes by comparing these features. Image features are meaningful and detectable parts of the image [1].

According to Mesay [42], the definition of feature extraction is noted as “extracting the raw data or information which is most relevant for the classification purpose, in the sense of minimizing the within class pattern variability while enhancing the between class pattern variability” [42]. It

has an impact on recognition error rate and studies must give more attention to it in order to apply OCR systems for real life document image.

Feature extraction for character recognition is broadly classified as *Structural/Topological features* and *Global/Statistical features*. Structural/topological feature is concerned with geometrical and topological properties of the character that includes examining features such as strokes either concave or convex, end points, branches, junctions, connectivity and holes etc. that exist in the character whereas global/statistical features are obtained from the arrangement of points constituting the character matrix. Some of the common techniques for statistical feature extraction include Zoning, Moments, Projection Histograms, N-tuples, Crossings and distances [63] and further discussions are made on Michael [43].

2.1.2.4 Classification

After the extraction of unique features that represent a character image, the next step is the classification stage which is the decision making stage using the extracted feature vectors. It is defined in [37] as a process of assigning the sensed data to their corresponding class with respect to groups with homogenous characteristics, with the aim of discriminating multiple objects from each other.

Classification generally performs two tasks: *Training* and *Testing*. Training is a process of extracting certain patterns from training datasets to store them in certain format called model and testing task is about using the designed model for future predictions or decisions made by the OCR system to predict a feature vector's ASCII or UNICODE representation [49].

Machine learning is a natural outgrowth of the intersection of Computer Science and Statistics. It is one type of Artificial Intelligence (AI) that provides computers with the ability to learn without being explicitly programmed [49]. It focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data, they are able to predict on the unseen examples. In order to perform this, the learner has to build a general model using classification methods to be used later to provide new predictions.

Yaregal [67] discussed in detail about the pattern recognition techniques. The aim of pattern recognition is classification and they are expected to perform either *supervised* or *unsupervised classification*. Supervised classification, where a given pattern has to be identified as a member

of already known or predefined class and classes are defined by the system designer whereas unsupervised classification, where a pattern needs to be assigned to a so far unknown class of patterns and classes are learned based on the similarity of patterns [67].

A variety of techniques have been implemented by scholars from the worldwide [67]. Verma and Ali [63] categorized that most commonly used classification techniques into *Statistical methods*, *Syntactic/Structural methods*, *Template matching*, *Artificial Neural Networks (ANN)*, *Kernel methods.*

Statistical methods are automatically trainable algorithms and they use a set of characteristic measurements usually called global features extracted from characters by partitioning the feature space. The purpose of statistical methods is to determine to which category a given pattern belongs by making observations and measurement process to prepare a set of numbers that is used to prepare a measurement vector. Some of the examples of techniques under this category are; K-NN, Bayesian classifier, Quadratic Discriminant Function (QDF), Linear Discriminant Function (LDF), Euclidean distance, cross correlation, Regularized Discriminant Analysis (RDA) [63]

Syntactic/Structural methods classify input patterns on the basis of components of the characters and the relationships among these components. Yaregal [67] and also his other studies with Josef Bigun [68]-[79] applied this method to develop Amharic recognition system for both printed and handwritten texts. This method use primitives of characters for classification. First the primitives of the character are identified and then strings of the primitives are checked on the basis of pre-defined rules. Verma and Ali [63] noted that a character is represented as a production of rules structure whose left-hand side represents character labels and whose right-hand side represents string of primitives. They noted that the method is good for recognition of handwritten texts [63].

Template matching is one of the simplest and common approaches of pattern recognition generic operation which is used to determine the similarity between two entities. In this approach, the prototype of the pattern that is to be recognized is available and a given pattern is compared with the stored ones to be recognized [63].

Artificial Neural Networks (ANN) is a method that simulates the way that human neural system works. It samples the pixels in each image and matches them to a known index of character pixel

pattern. Neural networks are ideal for specific types of problems, such as processing stock market data or finding trends in graphical patterns. Common types of ANN are feed-forward network, Convolutional Neural Network, Vector Quantization (VQ) Network, Auto-association Network, Learning Vector Quantization (LVQ) [55].

Kernel methods are the most important and powerful classification methods that include Support Vector Machines (SVM), Kernel Principal Component Analysis (KPCA), Kernel Fisher Discriminant Analysis (KFDA), etc. They are in a group of supervised learning methods that can be applied to classification to produce a model which predicts the target values of test data. Different types of kernel functions of SVM are Linear, Kernel, Gaussian Radial Basis Function (RBF) and Sigmoid [43] [63].

2.1.2.5 Post Processing Techniques

This phase in OCR system is performed after the result of the classification process to check errors that may have occurred due to the imperfections of classification algorithms. There can be many reasons for misclassification of characters. One example is structural similarity of characters and such errors are usually gone unseen unless additional post processing techniques as spell checker, dictionaries or error checker methods are applied. It can be done automatically or with the help of the user to identify misclassified characters and correct them [23].

Million [44] stated, "To enable the Amharic OCR system search for obvious errors and locate possible alternatives for unrecognized words, there is a need to develop post processing techniques such as spell checker, thesaurus, grammar, etc.". Verma and Ali [63] discussed that post processing step as the grouping of the symbols. They explained it as a process of performing the association of symbols into strings called grouping. Therefore this stage is very helpful for the improvement on the performance of recognition system.

2.1.3 Verification Phase

When OCR systems are dealing with noisy and poor quality documents, the result shows rejection of some characters that may go unseen. They are usually flagged by OCR systems for possible human intervention. Therefore, there is a need for human proofreading in order to

correct those errors. Also, after the recognition is done, inspection must be done to check the correctness of transcription done by the system [43].

2.2 Writing system

Understanding the attribute of each character and the allowed combinations between them to make up words, phrases and sentences is helpful to extract features as well as fix a set of rules such as dictionary specific to a given language. OCR systems heavily depends on the writing system of the language due to the essential elements of recognition systems such as character positioning, text orientation, word and character segmentation, and character forms that rely on understanding the language in question and its writing systems [1].

Humans started expressing thoughts through writing or symbol about 5,000 year's ego [21]. The first symbols they used were simple pictures of objects. Egyptians, for instance, used picture writing on monuments to convey information [21]. Language is the capacity for acquiring and using complex systems of communication [1]. The symbols used in the writing have to be understood by all users of the language and must deliver the same message to different readers. This art of expressing ideas through symbols is called *writing* and the nature of writing those symbols is known as *writing system*. Baye [10] presented the types of writing systems into three as *logographic, syllabic and alphabetic***.**

The logographic system was used first around 5000 years ago and place of origin is in the areas of Palestine and Syria. In this system, one symbol represents one word. For example a language of 100,000 words could have symbols as many as 50,000 - 60,000 and one example can be the old Chinese writing system [10].

The syllabic system represents a phoneme using a symbol. Phoneme is a combination of a vowel and a consonant. In this system, the numbers of symbols needed for a given language is determined by the number of basic sounds used. For example, if a language has 25 consonants and 5 vowels, the total number of phonemes is 125 (25x5). One example of this writing system is the Amharic writing system [10].

The third and last one is alphabetic system, which is originated from the syllabic systems used in the Semitic languages of the Middle East. It is also called Greek Alphabet because its origin is

tied to Greek and lately adopted by Romans and introduces it to their colonies which enable the wide use of the Latin language. Nowadays, it is used in western world and most parts of Europe, and also many African countries use it [10].

There is no exact estimate of on the number of human languages available in the world. However, estimates vary between 6,000 and 7,000 languages in number [30]. In Africa, more than 2,500 languages including regional dialects are spoken and most of the writing systems use a modification of Latin and Arabic scripts [48].

2.3 Amharic Writing system

Ethiopia is one of the countries in east Africa with a mosaic of ethnicities and many indigenous languages. There are more than 70 languages that are spoken and most of them belong to the Semitic and Cushitic branches of Afro-Asiatic family [26]. From those many set of languages spoken in Ethiopia, Amharic is the most dominant since 5th Century. It belongs to the Semitic group and it is one of the languages in Africa which have its own indigenous scripts and begun to be used for the writing purposes since 19th century. Nowadays, it is the second most spoken Semitic language in the world after Arabic and also both the official and working language of Ethiopia. It is the most commonly learnt language next to English throughout the country [48].

Semitic Sabean peoples of southern Arabia are the first to introduce syllabic writing system for Amharic language to the northern part of Ethiopia 2,500 years ago. As a result, the version of the script is known as Sabean script which is written from right-to-left. The system was used for a long time in the northern part of Ethiopia until the Axumite time whence it gave way to Geez [10].

Geez, the language of the Ethiopian Orthodox Tewahedo Church throughout medieval and modern times, gave rise to the Semitic cluster of languages [26]. It became a written language only after it took 24 of the 29 Sabean characters and modified 16 of them into a different look. It also took 2 additional characters from the Greek script, namely, P and PP ("ጰ" and "ጳ"). The style of writing was also restricted and modified to left to right. Geez used such a script and writing system between the 4th and 7th century [4]. The following figure 2.4 shows the change in the form of the 16 characters [43]. Geez also adopted a number of different vowels instead of the one used by Sabean [10].

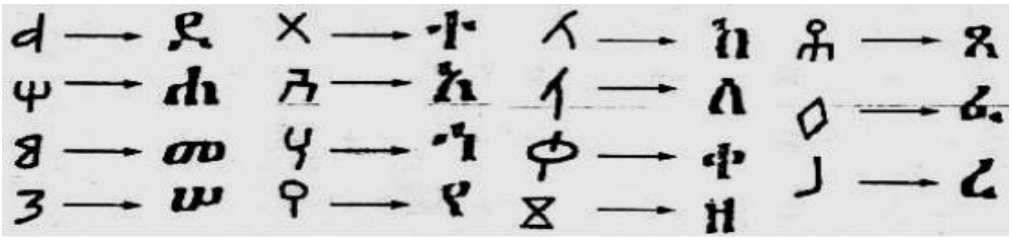


Figure 2.4: Evolution of Sabean to Geez Michael [43]

Baye [10] also discussed that Amharic also took the Geez scripts and became a written language just like Geez took and modified the Sabean scripts to become full-fledged written language. On top of the 26 characters, Amharic also needed additional characters to represent sounds that it acquired from Cushitic languages. This was done by placing a small bar (or hat) on top of 7 characters (such as "ሸ", "ቸ", "ጀ", "ኸ", "ኬ" and "ዠ") that were inherited from Geez [10].

Amharic is written in the unique and ancient Ethiopic script (inherited from Geez, a Semitic language). Nowadays, its complete set (see Annex I) is now effectively a syllabary requiring over 300 glyph shapes including characters, punctuations and numbers [48].

2.3.1 Amharic Characters

Amharic writing system has 33 basic characters. There are other 6 orders derived from the basic forms and represent syllable combination consisting of a consonant and vowel except the 6th order, which may represent either the consonant alone or the consonant followed by a vowel. Totally, they become 7 orders, as shown in Table 2.1 [26] [67].

1st order	2nd order	3rd order	4th order	5th order	6th order	7th order
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
Ha	Hu	Hi	Ha	He	H	Ho
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
La	Lu	Li	La	Le	L	Lo

Table 2.1: The seven order of Amharic writing system

As Table 2.2, besides the basic characters, there are series of derived characters to represent labialized velar consonants. For example, these are velar sounds like /k/, /g/, /q/, and /h/ that are pronounced with the lips rounded regardless of the vowel [26]. These are called the modified version of characters (labialization characters) such as "ሏ", "ሟ", "ሯ", "ሿ", "ቧ", "ቷ" etc. which

are 44 in number. But, according to Million [44] among those labialized characters, only 20 (such as “ሟ”, “ሯ”, “ሷ”, “ቿ”, “ቷ” etc.) of them are common and usually listed as an appendix to the main list [19] (see Annex I for the complete list).

ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ		ቧ			
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኩ	ኳ	ኴ	ኵ

Table 2.2: Sample of characters representing Labialized Velar Consonants

Amharic characters use more than one orthographic representation for the same sounds which are taken as a problem in the writing system and most of the scholars suggest that eliminating such repetitive characters for better computer representation [26] [43]. Some of the examples of such letters are “ሀ ፣ ሐ ፣ ኀ”, “ሰ ፣ ሠ”, “ጸ ፣ ፀ”, “አ ፣ዐ”.

2.3.2 Amharic Numeration Systems

Amharic numeration system (see Table 2.3) consists of basic single symbols for one to ten, for multiple of ten (twenty to ninety), hundred and thousand and they are presented in figure 2.6. These numerals are derived from the Greek numerals with some modifications and each of the symbols has horizontal strokes below and above [10]. There is no representation of zero in Amharic number system and using this system of arithmetic purpose is very difficult.

	1	2	3	4	5	6	7	8	9
× 1	፩	፪	፫	፬	፭	፮	፯	፰	፱
× 10	፲	፳	፴	፵	፶	፷	፸	፹	፺
× 100	፻								
× 10.000	፼								

Table 2.3: Amharic Numeration System

2.3.3 Amharic Punctuation Marks

Million [44] noted that Amharic writing system uses eight punctuation marks for different purposes and some of them are structurally similar with Latin languages and used for same context [44]. Some of the commonly used punctuation marks are:

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

Word delimiter (ሁለት ነጥብ) is most commonly used in historic documents but the modern writing styles commonly use space as a word separator. Some of the borrowed symbols are question mark (‘?’), exclamation mark (‘!’), arithmetic operators such as ‘+’,’-‘,‘*‘,‘/‘, brackets (‘(‘,‘)‘), quotation marks (“,”), etc.).

2.3.4 Features of Amharic Writing System

The basic characteristics of Amharic writing system are quite similar as English language. They are summarized at [43] as follows;

- ✓ It is written in a horizontal direction and from top to bottom.
- ✓ White spaces are used as separation of words.
- ✓ There is proportional spacing between characters.
- ✓ Characters are written in a disconnected manner (e.g. “አበበ”).
- ✓ Most of upper strokes are attached to main character (e.g. “ሸ”,“ኸ”) but for some characters (e.g. “ሸ”), numbers “፩, ፪, ፫, …” and punctuation marks “ ።, ፡, ፣ ”, there is disconnection between strokes
- ✓ It has no lower and upper cases.
- ✓ A line of Amharic printed script lies at the same level, having no ascent and descent.
- ✓ Questions end by using symbol, question mark “?”.

2.3.5 Real Life Amharic Documents

There is a huge amount of Amharic documents such as letters, newspapers, magazines, historic and modern books, pamphlets, etc. are available in government and private offices, libraries and museums having different writing styles and formats. So, enabling such rich information items to their digital format for effective access, reliable storage, computability and long term preservation is very important. These real life documents can be categorized into printed, typewritten and handwritten [26] [48].

2.3.5.1 Printed documents

A number of different types of Amharic fonts for printed documents are available these days and commonly 'Power Geez', 'Visual Geez', and ‘Nyala’ are used [13]. The following Table 2.4 presents Amharic characters written in different fonts to clearly understand the features of printed Amharic text.

ኢትዮጵያ	Nyala
ኢትዮጵያ	Power Geez Unicode1
ኢትዮጵያ	Power Geez Unicode2
ኢትዮጵያ	Power Geez Unicode3
ኢትዮጵያ	Visual Geez Unicode

Table 2.4: Commonly used Amharic Fonts

We can observe from the above example that words belonging to the same class but printed using different typefaces varies both in shape, width, line thickness, etc. and also there are many possible representation of the same letter among various fonts that produce a character big in shape or small size [13]. Those printing variations become a challenge for the development of recognition systems.

2.3.5.2 Type written documents

According to Dereje [21], who conduct his study on recognition for typewritten documents, the first type writer is called Olivetti Lexicon 80 was made in 1950 and a number of documents are available in the form of books, magazines, letters, etc. He clearly described the characteristics of typewritten Amharic text and noted that the height and width of individual characters in a

typewritten document are not constant. However, the space between each character is proportional [21].

Biniyam [13] finally mentioned the main challenge of using Amharic typewriter is due to the dust filled print heads and other scrapes of ink from the ribbon, loop appendages of some characters and words appear as solid black circular image in most typewritten documents [13]. As a results there exist connected characters in typewritten documents when two consecutive characters (especially the second, third and fourth forms) take up all the space in between.

2.3.5.3 Handwritten documents

Handwriting is the most dominant means of written communication. It also brings difficulty to automation of handwritten documents [65]. According to Dereje [21], it has started in the form of Egyptian pictorial writing (hieroglyphics) that finally gave birth to most of the Middle Eastern scripts and continued as means of communication and recording information in daily life.

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 [43].

2.3.6 Degradation levels in real life documents

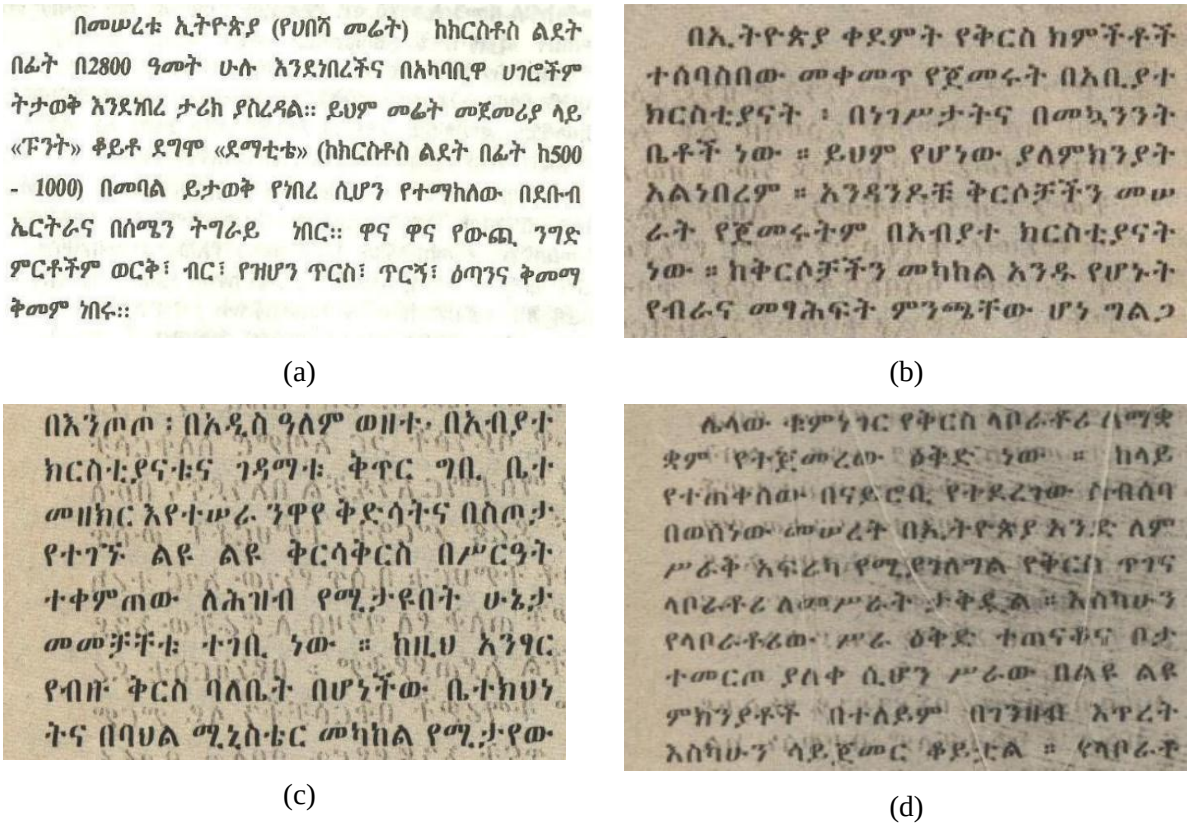
There are a number of paper based historic and other documents in real life which are poor in quality due to many reasons. As we have discussed in the preprocessing section 2.1.2.1, most of the noises appear on real life documents wear salt and pepper, cuts and breaks, blobs and Erosion of Boundary Pixels based on the degradation modeling discussed by Million [45].

Biniyam [13] classified these noises on real life documents based the noise prevalent in the document images and the pixel intensity, he classified them into low level, medium level, high level and very high level using the following criteria [13].

- ✓ **Criteria 1:** If a document image has less intensity and a little background noise behind words, then it is low level.
- ✓ **Criteria 2:** If a document image is affected by blob noise that connects different words together, blurring and higher background noise than low levels, it class is medium level.

- ✓ **Criteria 3:** If a document image characterized by considerable background and show-through noise from the back of the paper, it is classified as high noise level.
- ✓ **Criteria 4:** If a document image is plagued by blurring, aging and much more background and show-through noise, and blob noise than all other levels of noise, then it is very high level noisy document image.

This study uses those datasets prepared by Biniyam [13] for experimenting the Amharic document image retrieval system.



(a) (b)

(c) (d)

Figure 2.5: Datasets based on the level of degradations as presented by Biniyam [13]:
(a) low level noisy document (b) medium level noisy document (c) high level document and (d) very high level document

From the above categorization of degradation levels as collected by Biniyam [13], it is visible that there is an ink-bleeding noise for the document images above medium level degradation. And applying noise removal techniques that work for other noise types are not effective on those documents. Such types of noises are one of the difficulties for document image segmentation, especially segmentation of lines, words and characters is very challenging and have a great impact on recognition result.

2.4 Challenges in Building Amharic OCR

As I have mentioned in the previous sections, developing OCR system for Amharic real life documents is not a recent focus. Some studies have been made and progress is shown in every study. The area is still an active research area due to the following challenges that are still faced by researchers [45].

- **Large number of characters in scripts:** there are more than 300 characters which becomes a challenge to develop Amharic OCR because of the memory and computational requirements are very intensive. We need to design a mechanism to compress the dimension of character representation and select suitable classifier for better performance in recognition.
- **Printing variations:** the font faces combined with different font styles and sizes creates a challenge in study of developing an OCR system. The texts produced on such combination produce different versions of the same text which confuses the brain part of recognition, the classification stage. In order to solve such problems we need to look a way to normalize these variations into common.
- **Similarity of characters:** the main difficulty on the Amharic document image recognition is the similarity of characters that are even difficult to be differentiated for humans. He finally suggested that further studies must be made on exploring better feature extraction techniques for identifying unique features that represent the characters.
- **Degradation of documents:** scanned document images from a real life such as books, magazines and newspapers are always full of noises and low quality. Million [45] noted that the popular artifacts are excessive dusty noises, large ink-blobs, vertical cuts, low paper quality, low ink quality and also floating ink from facing pages. This is not only a problem of Amharic scripts, other non-Latin scripts even face this problems so we need to study carefully for appropriate design that can reduce such challenges.

2.5 Related Studies on Amharic OCR Systems

There are some investigations that are conducted by different researchers to contribute for the development of better performing Amharic OCR systems for real life document images with different writing style. In this section, the summary of some of the local researches conducted on

printed, real life, typewritten and handwritten OCR systems and future research directions are presented.

2.5.1 OCR Systems for Printed Documents

The first attempt to build Amharic OCR system was made by Worku [66] for printed documents with WashRa font and font size of 12pt by studying the features of Amharic Characters. He investigated to test the OCR algorithms that can work with other scripts for Amharic characters recognition by adopting stage by stage segmentation algorithm proposed by Pal and Chaudhuri [16]. He adopted the recognition algorithm on 33 base characters and their six forms of the Amharic characters. Stage by stage segmentation algorithm operates in three successive steps to detect lines, words and characters from the document image respectively [66].

In Worku's study, no image preprocessing or enhancement algorithms were used. He tested two algorithms and selected contour analysis to extract feature from the global and local aspect of the character. For recognition, he applied binary tree classifier and better performance was observed with 97.31% accuracy on his test set. However, the major failures of his study were on segmentation of connected and formatted characters of italic and underlined characters which results on his test on other Amharic real life document images from magazines, newspapers and books a very low performance and also for the italicized document images 0% were achieved and he finally recommended them as future research directions [66].

Ermias [24] conducted a study to incorporate preprocessing techniques to the adopted algorithm on formatted Amharic texts. Some of the processing steps he introduces include underline detection and removal algorithm used in Bangla script OCR study [16] by modifying a threshold value, thinning and StretchBlt() function of Turbo C++ for size normalization algorithm which causes an error of disconnection of characters and introduction of some noises to the character image [24].

For recognition purpose, Ermias adopted Worku's binary tree classifier algorithm to test the performance of the system. However, the result of the study was poor and he mentioned that such performance was achieved due to the width of characters are changed by adopted thinning algorithm and he finally recommended other methods of sizing bitmaps and recognizing superscript and subscripts for further investigation.

Berhanu [11] conducted a study on Amharic character recognition for printed documents which have the same font face (ALTEthiopian font) and size of 16pt. For the segmentation purpose, he adopted Worku's algorithm which is stage by stage for segmenting line and for character segmentation, he used MATLAB bounding box projection algorithm which considers every character as a connected points. Feed forward ANN with back propagation algorithm are both uses as a feature extractor and classifier and also image size normalization for feeding the ANN because of it only accepts a fixed number of inputs. The study shows poor recognition performance and he mentioned that further studies must be made on segmentation, image thresholding and noise removal algorithms must be made [11].

Million [44] conducted a study and made an attempt to come up with a generalized approach that enable previously adopted algorithms recognize Amharic text with a different font styles and faces. He tested two thinning algorithms before segmenting character and compared them by weather the result shows connectivity loss and reasonable processing time. He combined the two algorithms and come up with the hybrid algorithm which shows a remarkable result. For segmentation purpose, stage by stage segmentation is used and developed Visual C++ program to extract features and develop feature database using binary tree that is also used as a classifier. The result of his study was promising but poor performance was registered and he recommended that a generalized and more flexible recognition algorithm must be developed [44].

Yaregal [67] also tried to come up with an algorithm which is independent of font sizes. He applied stage by stage segmentation algorithm by modifying it to accommodate various font sizes. For feature representation, he used primitive structures that are found in Amharic characters by first developing a method that can identify boundary of characters. Then he came up with a tree that can hold pattern of these primitives and ANN were used as a classifier. He finally recommended that algorithms for primitive identification, extraction, and connection relationship should be developed and also segmentation algorithms should be studied due to the approach used were failed to segment italicized characters [67].

Yaregal and Bigun [69] also made an effort to develop automatic recognition of Ethiopic script by applying structural and syntactic techniques. The recognition system is developed by extracting primitive structural features and their spatial relationships. A special tree structure to represent the spatial relationship of primitive structures was used and for each character, a

unique string pattern is generated from the tree and recognition is achieved by matching the string against a stored knowledge base of the alphabet. Direction field tensor algorithm was applied and the structural features with their spatial relationships were extracted [69].

Direction field tensor is an ordinary way of estimating the local direction of pixels in an image neighborhood to compute the gradient field. It suppresses image intensity differences that do not fit to a line in a local neighborhood and amplifies those that fit to a line. Yaregal and Bigun [69] tested the performance of the character segmentation, primitive extraction and character recognition and scored 98%, 96% and 92% on clean printed documents and 93%, 94% and 86% for the newspaper and book document images respectively. Finally, they suggested that the percentage of accuracy can still be improved to higher level by working more on direction tensor and pattern matching algorithms.

2.5.2 OCR Systems for Typewritten Documents

Dereje [21] tested Worku's algorithm on a typewritten document which were from four different types of typewriters and poor performance was observed by scoring maximum result of 8.26% and he mentioned some justifications for the scored poor result such as sensitivity of recognition algorithm to the character features that is because the features and shape of typewritten characters vary from printed characters, low quality of the typewritten documents and their degradation, and also typewritten characters were not considered during Worku's study.

Dereje studied that most typewritten characters are connected which results misclassification. For example, in his study form his test set 20.06 % were connected characters. He applied two image restoration techniques to enhance image fixing noises appears on document image; the *mathematical morphology algorithm* for salt and pepper noise and *binary morphological algorithm* for subtractive and additive noise. He found binary morphological algorithm works better than the mathematical morphology algorithm but he mentioned that it was very slow.

For the segmentation of lines, words and characters, he tried to study the recursive segmentation algorithm. Recursive segmentation algorithm works by combining the segmentation and recognition algorithm together in a recursive manner and as a character is segmented; each of them is submitted to recognition algorithm. The characters are recognized in one step if it is segmented correctly and if connected characters are segmented, the recognition algorithm rejects

it. The size of the rejected character image is reduced from right and again submitted to the recognition algorithm until the character is recognized.

However, due to time limitation, he modified stage by stage segmentation algorithm which was originally suggested by Pal and Chaudhuri [16] to segment connected characters and 5% performance increase were observed due to this segmentation algorithm.

For feature extraction, he used contour analysis and they are used to create binary tree. He also included binary morphological filtering algorithm to remove noises in the image. The study shows an improvement of recognition performance. He recommended segmentation algorithms that can efficiency isolate Amharic typewritten character images and feature extraction schemes that are not sensitive to variations of same characters must be studied [21].

2.5.3 OCR systems for Handwritten Documents

Nigussie [52] made the first study among the recognition of Amharic handwritten characters on bank check amounts. He applied underline removal, slant normalization and character size normalization without applying noise removal methods. He adopts a stage by stage segmentation algorithm which was used by previous studies of Amharic OCR and originally suggested by Pal and Chaudhuri [16] and feeds the segmented and normalized character for ANN to extract the unique features as well as classify characters. However, the result obtained from his study was unsatisfactory [52].

Mesay [42] also made an attempt to recognize handwritten characters present in postal address labels. Line fitting algorithm is used by applying a simple geometric calculation to determine features which could represent and describe the character as uniquely and precisely as possible. He normalized characters using 32x32 pixels which are divided into 16 smaller squares of 8x8 pixels. Then the least square technique was applied to fit a linear model to the distribution of foreground pixels and three features were extracted from each smaller square. For training and classification purpose, he used ANN using back propagation algorithm through cross validation technique and reported a recognition rate rages between 2.3% - 34.9% in test sets. As a future research direction, he suggests studies on recognition of noisy images [42].

The other attempt made by Wondwossen [65] that use ANN classifiers to recognize a specialized type of handwriting "Yekum Tsifet" achieved recognition accuracy between 16.18% - 31.53% on test sets. He didn't use pre-processing techniques that he finally suggested thinning, noise removal and slant correction to be considered in order to enhance recognition performance.

Some other studies were made on recognition of handwritten such as; Yaregal and Bigun [78] studies on the recognition of Amharic word in unconstrained handwritten text using HMMs and they obtained promising results and also another study of Yaregal and Bigun [74] on writer independent offline recognition of handwritten Ethiopic characters.

2.5.4 OCR systems for Real Life Documents

Million and Jawahar [46] tried to study the recognition of real life Amharic documents such as newspaper, magazine and books; and to recognize printing font variations. They have used binarization, noise removal and skew correction techniques. Projection profile technique is used for correcting the skewness of the document image and they applied horizontal and vertical projection profiles method for segmentation after document preprocessing is done [46].

Principal component analysis (PCA) for dimensionality reduction and linear discriminant analysis (LDA) were consecutively used to extract features and the results are fed into multi-class Support Vector Machine (SVM) for training and classification. High recognition rate was achieved for both printed and real-life documents but they faced misclassification of characters due to the artifacts such as large ink-blobs joining disjoint characters or components, and cuts of characters at arbitrary direction due to paper quality or foreign material [46].

Abay [1] also tried recently the recognition of real life documents by taking his data from holy bible, popular old Ethiopian fiction 'Fiker Eskemekabir', 'Addis Zemen' newspaper and 'Federal Negarit Gazette'. He had tested three (linear, median and adaptive) filtering algorithms and adaptive works well for the real life Amharic documents. Stage by stage segmentation which failed to segment Amharic characters having disconnected strokes and also normalization and thinning techniques are used. Finally, ANN is used as both feature extractor and classifier. Abay reported poor recognition rate for the degraded real life documents for the holy bible, newspaper and 'Federal Negarit Gazette'. He recommended further studies must be made on feature extraction and noise removal algorithms [1].

Recently, Michael [43] developed recognition of real life documents by applying different preprocessing techniques. He tested Median and Weiner filtering algorithm and he selected Weiner algorithm because it performs better. Also for image thresholding, he tested Otsu and Sauvola algorithms and Sauvola performed better. For segmenting lines, words and characters, he developed a visual C# program which modified projection profile that works by calculating the sum of black pixels in horizontal axis for line and vertical for segmenting words and characters. From the result it is observed that characters which have discontinuity in their body, over lapping pixels, touched pixels, and adjacent pixels with neighboring characters are failed to segment successfully.

Michael also applied the underline removal and normalization methods. Modified zoning technique is employed for feature extraction which first segments a character image into abstract zones according to the number of dimensions and calculates the average vector distance for every cell by taking the ratio of black pixels to all pixels that the cell spans by considering the lower left corner as an absolute origin. For classification purpose, multiclass SVM is employed and he stated that it is low generalization error and computationally inexpensive. Better results were achieved and he recommended that better feature extraction techniques must be studied to deal with the characters similarity and also better segmentation algorithm should be explored in order to increase the recognition rate. He also mentioned that better noise and underlines detection and removal, thresholding, skew detection and correction, and automatic page segmentation algorithms should be explored in future studies [43].

Based on the result and recommendation of the recent study made by Michael [43], there was segmentation errors that decreases the performance of the recognition. He recommended the need to explore better segmentation algorithm for the enhancement of Amharic document image recognition. Yaregal [43] and Yaregal and Bigun [68]-[79] also suggests the need for future investigation on exploring effective preprocessing, segmentation and primitive extraction technique on real life documents with different levels of degradations for the improvement of the effectiveness of recognition systems.

2.5.5 Word level Page Segmentation

Gedion [26] studied a word level page segmentation technique on real life Amharic document image collections for his study of Document Image Retrieval System (DIRS) without recognition. He explored various page segmentation techniques and tested them to segment the text area from the graphic section of document image and he successfully separated the text area from figures and tables [26].

Based on his discussion, there are limited numbers of shapes of text characters but the shapes of non-text elements are unlimited. Therefore OCR engines treat both text and non-text components differently, such that they only recognize text components and then arrange recognized text and also the graphics of non-text components in an output document using layout information.

Gedion used MATLAB built in functions such as Watershed, connected component (cc) analysis, Horizontal Run Length Smoothing (HRLS), Hough Transform, Dilation and Bounding Box Approach to separate the text/graphic area and also individual word images from the document. He finally proposed a good combination of cc, HRLS, Hough Transform and Dilation for successful word level page segmentation and he get a remarkable result. However, he observed the proposed system it removes text with larger font sizes and also recognizes some text parts as a line and tackling those problems are forwarded as a future research directions.

Based on the above literature review, we can note that the major challenges in Amharic OCR development are degradation of real life documents and detection of different layouts from documents such as column blocks, graphics, logos, table lines and other shapes. Hence, the current work tries to explore an effective page and text segmentation that can detect column block, table lines, graphics and text at character level. By doing so, the work tries to add knowledge in the area of Amharic OCR by experimenting techniques that can improve the applicability of Amharic OCR for real life documents by testing image preprocessing techniques such as skew detection, noise removal, binarization, page and text segmentation algorithms and select the best combination to be integrated with previously developed recognition system. Also, exploring an effective character segmentation technique that can detect and split connected characters is the main focus of the study.

CHAPTER THREE
IMAGE PREPROCESSING AND SEGMENTATION TECHNIQUES

From the various challenges and problems confronted in the development of OCR systems, the major one is the degradation and poor quality of document images. Most of real life document images contain different level of degradations that makes preprocessing phase one of the main tasks applied on a document image for the purpose of efficient recognition. To fix various deficiencies and degradations of document images, several processes such as correcting skew angle, detecting image layout and text area, detecting underline and removing it, noise removal, thresholding and some other processes are suggested [13]. The type of preprocessing that is going to be applied in OCR systems may vary due to the type of script, degradation level and type, and also the nature of deficiencies found on it [43].

The other image processing techniques that is applied before the feature extraction and classification stages of OCR system is known as segmentation. It is grouped into two levels as text/graphic segmentation and text segmentation. Text/graphic segmentation is a process of dividing the document image into homogeneous zones, each consisting of only one physical layout structure (text, tables, pictures, etc.) and most of the times it is applied before even preprocessing techniques takes place whereas text segmentation techniques help to segment the text part of the image to lines, words and characters [26].

This study focuses on integration of better preprocessing and segmentation algorithms to the recognition techniques developed so far in order to enhance the accuracy of Amharic character recognition for real life documents with varying noise levels. This chapter explores image preprocessing and segmentation techniques that are going to be applied before the recognition process is performed.

3.1 Architecture of the Amharic OCR

Every OCR system passes through some processes before fully recognizing the text on document image and the general architecture of the proposed OCR System for Amharic language is

presented in figure 3.1. It starts by scanning the physical document to a digital format known as image acquisition and the document image passes through some preprocessing steps to improve its appearance and quality by removing and/or minimizing degradations occurred from scanning, printers, document age, etc. Some of the operations involved here are skew detection and correction, page segmentation, noise removal and binarization.

Page segmentation is applied in order to identify tables and other lines, graphics/text, and columns whereas text segmentation is performed in order to segment text lines, words and characters in the document image. The segmented characters are normalized in order to have common size and they are fed into feature extraction phase to extract unique features of character images in the form of vector. Then the classification stage trains the classifier for enabling it to predict the future Unicode/ASCII values of characters. The prediction for unseen features is made by consulting the model that is prepared during the training part of classification and the final result is constructed into structured text and displayed to end users using word processing applications [58].

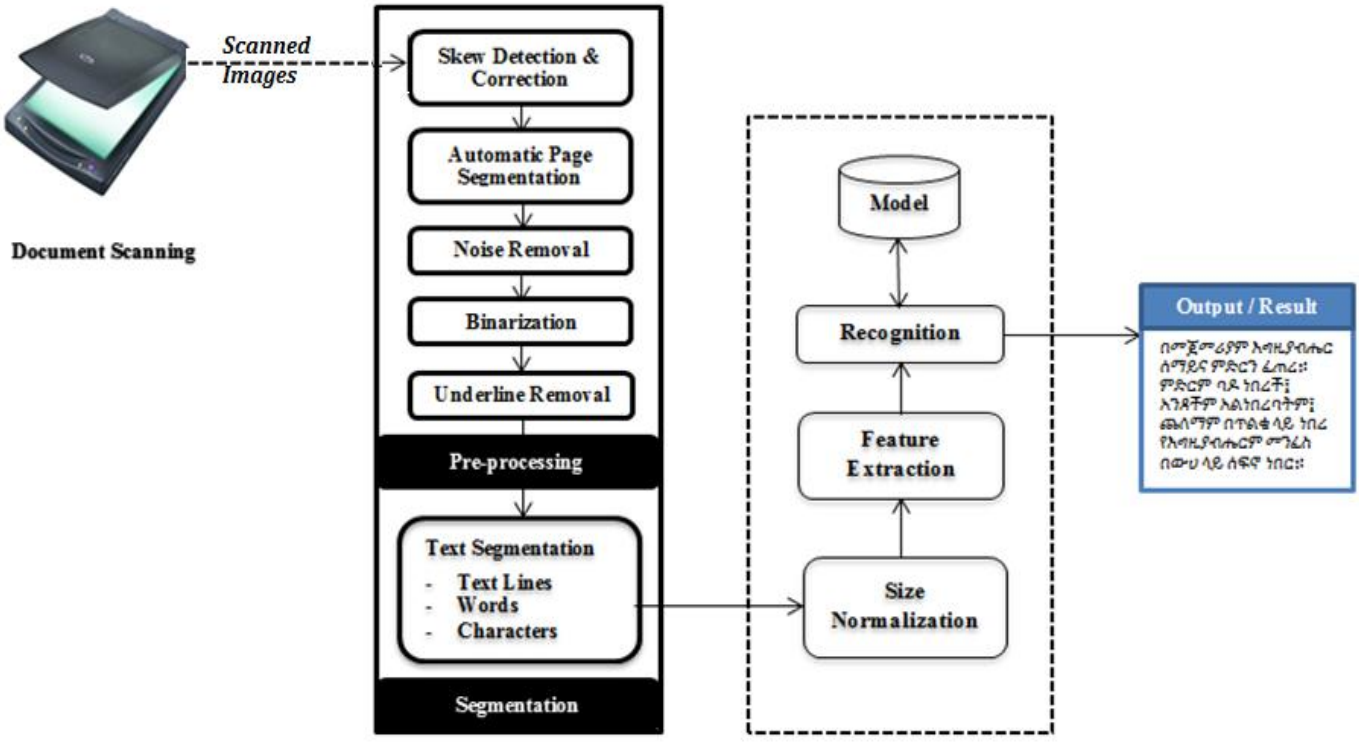


Figure 3.1 Architecture of the proposed Amharic OCR System

Rectangles in bolded line represent the focus of the present work.

The effectiveness and accuracy of previously developed Amharic OCR systems by Abay [1], Michael [43], Yaregal [69] and most previous studies were affected by the degradations that occur in real life documents.

3.2 Preprocessing Techniques

Preprocessing is an essential stage prior to segmentation and other remaining phases of OCR systems. It has a direct impact on the accuracy of the successive stages for recognition process. The stages in pattern recognition system are like a pipeline; meaning that each stage depends on the success of the previous stage in order to produce optimal/valid results [6] [7].

Real life document images have a number of limitations such as geometrical distortions, low resolution, various types and levels of degradations, and some other deficiencies [12]. Preprocessing describes any type of processing performed on raw data to prepare it for another phase. It is the maiden and key step in image processing for the enhancement of those mentioned limitations on real life document images. The preprocessing steps used in this study are discussed below.

3.2.1 Skew Detection and Correction

Text line is a group of adjacent characters, symbols and words in document images in such a way that horizontal straight line can be drawn [13]. The deviation of the baseline of the text from horizontal direction is called skew and the dominant orientation of the text lines in a document page determines the skew angle of that page. An example of skewed document image is displayed in figure 3.2 below.

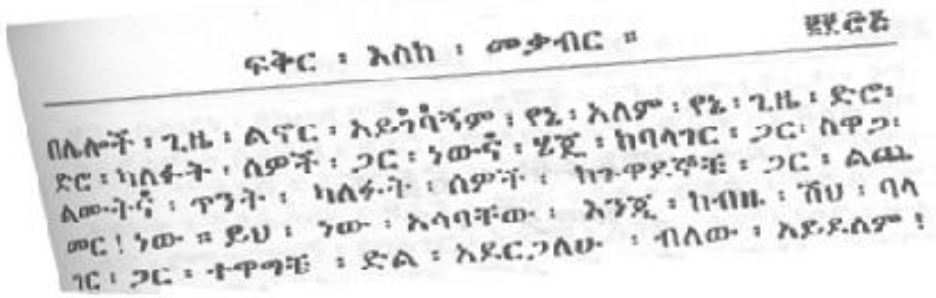


Figure 3.2: Skewed real life Amharic document image

Skewed document images must be preprocessed on the stage called *skew detection and correction*. It is an important image preprocessing steps that must be performed always before other stages of OCR system because of the possibility of rotation of the scanned document image (e.g. figure 3.2) and dependency of other phases on the rotation angle of the image. Therefore, document skew angle should be detected and corrected first before other preprocessing task.

According to the study made by Alginahi [6], skew detection techniques can be roughly classified as analysis of projection profile, Hough transform, connected components, Clustering and Correlation between lines techniques. He also stated that there are more than twenty five different methods of skew detection/correction techniques [6] [51]. Those various techniques perform two stage processes; the first is determining rotation angle ɵ and the second is rotating image by the angle -ɵ.

In this study, the adopted skew checker class implements document skew checking algorithm which is based on Hough line transformation. The algorithm is based on searching for text base black lines of text bottoms followed by white line below on the gray scale document image input which supposes that a white-background document is provided with black letters on a text document images. This technique effectively detects the skew angle caused due to some errors made in the process of document image acquisition or scanning or may be by other reasons and make corrections [31]. The Hough Transform algorithm is presented in section 3.2.4.

3.2.2 Noise Removal

Real life document images are composed of various type and level of noises due to many reasons. Noises can be introduced by optical scanning device resolution quality, paper quality, fax, photocopying, and the lack of efficiency in writing or printing instrument. The age of document can be another reason for the degradations due to the fact that historic documents face more dusts, cuts and breaks.

Noises can occur either or both in the foreground or background of document image. Therefore, the noisy dataset collection of Amharic document images used in OCR studies needs to be refined or preprocessed before text segmentation and other stages of OCR system are performed. Because noises/degradations found on those document images can cause disconnected lines, connected characters, large gaps between the lines, background may considered as foreground,

loss of information from the document, etc. So it is very essential to remove all of those noises and enhance the document image in order to improve the performance of recognition result for real life documents [13] [66].

The datasets used in this study are taken from real life Amharic document images with four levels and types of noises: *Low Level, Medium Level, High Level* and *Very High Level Noises* as collected and presented by Biniyam [13]. Because of the presence of degradations in these digitized documents from magazines, books, newspaper, bible, etc., there is a need to apply noise filters so as to reduce the effect of degradation before other recognition process are applied [48].

Million [45] categorizes degradations that are commonly observed in printed real life documents as salt and pepper, cuts, blobs and erosion of boundary pixels. Salt and pepper noise are the most prevalent one that is found on the Amharic real life document images [45]. Most of the natural images are assumed to have additive random noise which is modeled as a Gaussian [51]. Image denoising becomes a challenge for researchers because noise removal itself introduces some artifacts and blurring that can causes a loss of vital information from the document image.

Various noise detection and removal techniques are available and this study tests some linear and non-linear image denoising methods to see their effect on the degraded real life document images and also to find out the best combinations.

3.2.2.1 Mean Filtering

Mean (Average) filtering is a method of smoothing images by reducing the amount of intensity variation between neighboring pixels. It is the most popular and simple low pass filter that improves noisy images, flattens local differences and reduces sharpness by replacing any pixel value by the democratic vote of its mxn rectangular neighborhood [62].

Mean filter is the simplest linear filter implemented by local averaging operation (see equation 3.1) where the value of each pixel is replaced by the average of all the values in the local neighborhood:

$$\hat{\mathbf{f}}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{1}}{\mathbf{M}} \sum_{(\mathbf{i},\mathbf{j}) \in \mathbf{N}} \mathbf{g}\,(\mathbf{s},\mathbf{t}) \qquad \textbf{(3.1)}$$

Where, M is the total number of pixels in neighborhood (N) [13].

There are four algorithms available; arithmetic, geometric, harmonic and contra harmonic mean filter. Arithmetic mean filter is the simplest mean filter. From equation 3.2, S_{xy} represents the set of coordinates in a rectangular sub image window of size $m \times n$ centered at point (x, y). The mean filtering process computes the average value of the corrupted image g(x, y) in the area defined by S_{xy}. The value of the restored image *f* at any point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by S_{xy}. Mean filter simply smooth local variations in an image and noise is reduced as a result of blurring [28].

$$\hat{\mathbf{f}}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{1}}{\mathbf{mn}} \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{Sxy}} \mathbf{g}(\mathbf{s},\mathbf{t}) \qquad \mathbf{(3.2)}$$

Biniyam [13] noted that arithmetic and geometric mean filters are well suited for random noise like Gaussian and uniform noise whereas salt noises.

3.2.2.2 Median Filtering

For the elimination of the imperfections available on document images, median filter is efficient as compared to other non-linear filters. It is commonly used non-linear operator with a special type of low-pass filter which is able to remove noise and replace the bad pixels with reasonable values by substituting the image pixel values with the median of gray values in the local neighborhood of that pixel [13] [43].

For the pixel value *(x, y)* in an image and a given window size of *(mxn)*, the algorithm sorts the intensity values of pixels surrounding that pixel according to the window size. Once their values are sorted in increasing order by their values, the algorithm takes the median value as a new value for pixel *(x, y)*. If the number of pixels is even, the algorithm takes the two middle values to compute arithmetic mean of them and if the number of pixels is odd, the algorithm simply selects the mid value as a new value [6] [43].

The definition of median filter on [4] and [14] is presented for the image {I (x, y)} and the mean value *m (k, l)* with the point with coordinates *(x, y)* in the window size of *(mxn)*. If we assume that *m* and *n* to be odd, and if u (n) denotes the sorted sequence (u (n) ≥ u (n-1)) obtained from an array {I (x, y)} where x € {(k – (m - 1) / 2), … ,k + (m-1)} and y € {(l – (n - 1) / 2), … , l + (n-1)}, we have:

$$M(k, l) = (u(m\,n + 1) / 2)\,pk \qquad (3.3)$$

In median filtering, the input pixel is replaced by the median of the pixels contained in the neighborhood [62]. This is presented as equation 3.4 where W is a suitably chosen neighborhood:

$$u(m, n) = \text{median}\{y(m - k), (n - 1), (k - 1) \in W\} \qquad (3.4)$$

Generally, median filter is a nonlinear filter that is useful in removing isolated lines or pixels while preserving spatial resolution. It is found that median filter works well on binary noise but not so well when the noise is Gaussian. Moreover, its performance is poor when the number of noisy pixels is greater than or equal to half the number of pixels in the neighborhood [4].

As it is presented in algorithm 3.1, median filtering is done by replacing the value of each element by the median found in a window around the element. Thus, the median will in general replace a noisy value with one closer to its surroundings [13].

123	127	150	120	100
119	115	134	121	120
111	120	122	125	180
111	119	145	100	200
110	120	120	130	150

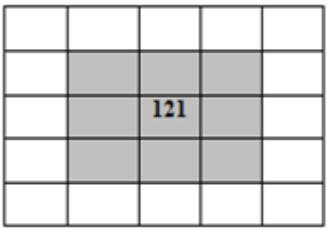


Figure 3.3 Illustration of median filter (a) Input image (b) Filtered image using median filter showing only the center pixel

The sorted pixel values of the shaded area are: (100, 115, 119, 120, 121, 122, 125, 134 and 145), providing a median value of 121 in the output image.

Algorithm 3.1: Median Filter algorithm

INPUT: *Pixel for gray scale image with a window size (mxn)*

OUTPUT: *a real number value for gray scale level*

STEPS:

1. *Get value for pixel (x, y) and neighbor pixels by mxn window size*
2. *Sort the luminance values of the pixels in ascending order*
3. *Count the number of pixels*
4. *IF count % 2 == 0*

   THEN

   take the middle values for arithmetic mean

   ELSE

   take the middle value
5. *Return the modified pixel value*

3.2.2.3 Weiner Filtering

Weiner filter has a long history that goes back to the Wiener-Hopf equations derived by Norbert Weiner and Eberhard Hopf in 1930's. It is a powerful linear filtering algorithm to remove salt-and-pepper and also other types of noises form the document images. The technique is also used for removal of blur type of noises in images. For example, noises caused by linear motion or unfocused optics [19].

The main goal of wiener filtering method is to design an input for some noisy data and minimize the effect of noise at the output of next phases according to some statistical data. The effect of noise is measured using the MSE and PSNR. A useful approach is then to minimize the MSE which is defined as the difference between the desired response and the actual filter output. It results a simple local smoothing whenever the variance in an image is large and it gives an improved local smoothing when the variance is small. Due to its selectiveness in preserving edges and other high frequency parts of an image wiener filtering method produces better results than previous linear filtering methods [13] [41] [43].

From Motwani's [51] discussion:

> *"Wiener filter is a filter used to produce an estimate of a desired or target random process by linear time-invariant filtering of noises by minimizing the mean square error (MSE) between the estimated random process and the desired process. The algorithm requires the information about the spectra of the noise and the original signal. It implements spatial smoothing and its model complexity control corresponds to choosing the window size. This method works well only if the underlying signal is smooth".*

The wiener filtering starts by calculating the mean (equation 3.5) and variance (equation 3.6) of neighboring pixels specified by window size of *mxn*. The algorithm also expects the variance for noise in order to estimate the desired response. In cases where it is impossible to determine the noise variance, it uses the average of all the local estimated variances. Once these values are identified, pixel wise Wiener filter using these estimates is calculated (equation 3.7).

$$\mu = \frac{1}{\mathbf{m} * \mathbf{n}} \sum_{\mathbf{x},\mathbf{y} \in \mathbf{n}} \mathbf{I}(\mathbf{x},\ \mathbf{y}) \qquad (3.5)$$

$$\boldsymbol{\sigma}^{\mathbf{2}} = \frac{1}{\mathbf{m} * \mathbf{n}} \sum_{\mathbf{x},\mathbf{y} \in \mathbf{n}} \mathbf{I}^{\mathbf{2}}(\mathbf{x},\ \mathbf{y}) - \mu^{\mathbf{2}} \qquad (3.6)$$

$$\mathbf{I}'(\mathbf{x},\mathbf{y}) = \mu + \frac{\boldsymbol{\sigma}^{\mathbf{2}} + \mathbf{v}^{\mathbf{2}}}{\boldsymbol{\sigma}^{\mathbf{2}}} (\mathbf{I}(\mathbf{x},\mathbf{y}) - \mu) \qquad (3.7)$$

Where m and n are local neighborhood of each pixel in an image *I*, $and\ v^2$ are the noise variance [41].

To get the restored version *R* of a given degraded image *I'* of some original image *I*, the output *R* must be close as possible to the "correct" image, *I* in order measure the restoration and to decide if it was a good job or not [13].

3.2.3 Binarization (Thresholding)

This is one of the mandatory steps in an OCR system that is responsible for the conversion of grayscale document image into bi-level (black and white) representation. It is very important to separate the foreground characters from their background. Applying binarization in document image removes some of its noises but it will depend on the level of degradations that an image contains and there is a need to apply noise filters beforehand to reduce their corresponding

effects in subsequent recognition processes [46]. Because of other subsequent algorithms such as underline detection and removal, segmentation, and feature extraction algorithms takes a bi-level input, binarization function needs to be effective.

The group of pixels representing objects of interest is called foreground pixels and the rest are called background pixels. Therefore the objective of binarization (thresholding) is to automatically choose a threshold that separates the foreground region with a single intensity (ON) and background region with a different intensity (OFF). After that, it reduces a gray-scale or color image to binary image, i.e., 1(ON) and 0(OFF) [5]. The algorithm is performed by selecting optimal threshold value on a grayscale image. Any pixel having intensity value less than the threshold is assigned to black (0) and above assigned to white (255) [13]. The concept of binarization can be applied using the algorithm 3.2 [13].

Algorithm 3.2: Binarization (Thresholding) algorithm

INPUT: *Filtered gray scale image*

OUTPUT: *Binarized image(0 and 1)*

STEPS:

If a[m, n] ≤ T

a [m, n] = Object(Foreground) = 0

Else

a [m, n] = background = 1

In order to select a threshold value T, there is no universal rule but some techniques have been proposed by various scholars and they are categorized in to two groups such as *global* and *local thresholding techniques*.

Global thresholding methods computes a single threshold value for binarizing the whole image and each pixel is going to be compared to this value for the decision making of background and foreground. However, they give better result for the documents that have uniform illumination documents and poor results are achieved for degraded documents. From this group, the better performing algorithm is Otsu's thresholding [13].

Local thresholding methods, also named as adaptive thresholding, compute a threshold value for each pixel individually using the information from neighborhood pixels. This works better than the global thresholding for the degraded real life documents with varying illumination across the document. Sauvola's binarization method is the best performing than other member of this class [13] [43].

3.2.3.1 Otsu's Thresholding Method

Otsu's thresholding method is one of the earliest and famous global thresholding methods suggested by Nobuyuki Otsu. It is based on the idea that involves evaluating the threshold which minimizes the weighted variance within a class and maximization of the variance between classes. It works directly on the grey level plot of histogram, so the method evaluates faster once the histogram is computed. It assumes the input image contains two types of pixels (i.e. foreground and background) and evaluates the optimal threshold distinguishing those two classes [60].

The algorithm uses a single threshold value for all the image pixels and it performs better when there is a consistency in luminance over the entire image. However, for degraded document images, adaptive thresholding which uses different threshold values for different local neighbors gives better result.

The following algorithm 3.3 shows the general step by step procedure that Otsu's method follows [43].

Algorithm 3.3: Otsu's Thresholding Method

INPUT: *Gray scale image (Filtered Image)*

OUTPUT: *Threshold /T/ (Real Number Value)*

PROCESS:

1. *Compute the histogram and probabilities of each intensity level*
2. *Initialize weight and mean*
3. *Step through all threshold values T=1,… to maximum intensity level*
   - *Update weight and mean*
   - *Compute the maximum class variance (corresponds to the desired threshold)*
4. *Return T*

3.2.3.2 Sauvola's Thresholding Method

Sauvola's thresholding method is a commonly used local thresholding method that consults neighboring pixels to decide the threshold value for a given pixel. It gives improved performance on documents in which the background contains light texture, big variations and uneven illumination. The algorithm computes a threshold value of a pixel using mean and standard deviation of pixels for a given window size [60].

The following algorithm 3.4 shows the general step of sauvola's method [43].

Algorithm 3.4: Sauvola's Thresholding Method

INPUT: *A pixel of gray scale image (Filtered Image), and window size of mxn*

OUTPUT: *Threshold /T/ (Real Number Value)*

PROCESS:

1. *Get the illumination of pixel (x, y)*
2. *Calculate mean and standard deviation*
3. *Calculate the threshold using the mean and standard deviation*
4. *Return T*

The above algorithm will be executed for each of the pixels found in the grayscale image and it takes a considerable time to binarize a given image. However, it gives good results in non-uniformly illuminated images and even in severely degraded documents.

3.2.4 Underline Detection and Removal

Just like other writing systems, Amharic also uses underlines usually for the identification of main ideas, topics and other ideas that we want to note. They are drawn horizontally below the specific text. Document images also contain lines other than underlines like over line, vertical line and some other different lines that are used for different purpose.

Underlines in document images can occur untouched with the text line they are emphasizing or touched with the lower parts of some characters in the text lines. They can also be found fragmented (or disconnected) and slightly curved [43].

Ermias [24] is the first to apply underline detection and removal from Amharic document images and he discussed underlines as they do not belong to the original image. This is due to the observed effect of underlines on other steps of OCR systems e.g. on Michael [43] investigation, where the result shows segmentation errors due to the existence of underline on the image. Therefore, underlines on Amharic document images needed be considered as noise and discarded from a document image [24].

Ermias [24] and Michael [43] adopted the algorithm suggested by Pal and Chaudhuri [16] for the removal of the top line from the Bangla script by modifying the algorithm to remove the underline from the Amharic script. The algorithm first assumes the lower zone as a region of interest (ROI), since underlines are normally found at the bottom of texts lines. If this region contains maximum horizontal projection (sum of black pixels along a row), the value is checked with a certain threshold value [24] [43].

In this study, another method is introduced and applied that is based on *Hough Transform* and the algorithm is modified to detect, plot and remove underlines and also other possible lines from the document images.

The Hough Transform method was introduced, in its most elementary form, by P.V.C. Hough in 1962, in the form of a patent. It is a technique which can be used to isolate features of a particular shape within an image. Its intended application was in particle physics, for detection of lines and arcs in photographs obtained at cloud chambers. Many elaborations and refinements of this method have been investigated [8].

Hough transform is a transform used to detect straight lines. To apply the transform, first some preprocessing including edge detection is desirable [29]. It is very helpful to detect not only the underlines, but also every line in the document image weather it is on the underline or over line, tables and frame borders can be removed using this method.

In Hough transform, a line in the image space can be expressed with two variables. For example: *Cartesian coordinate system*, parameters: (m, b) and *Polar coordinate system*: parameters: (r, ɵ) are considered as a line as shown in figure 3.4. [29].

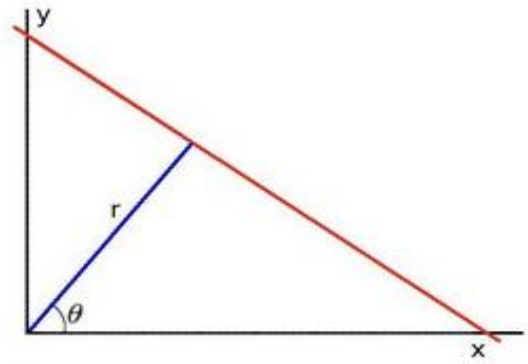


Figure 3.4 Polar coordinate System

For Hough Transforms, we will express lines in the *Polar system*. Hence, a line equation can be written as [29]:

$$\boldsymbol{y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(-\frac{r}{\sin\theta}\right)} \qquad (3.8)$$

By arranging the terms it gives: $\boldsymbol{r = x\cos\theta + y\sin\theta}$ and generally [64]:

For each point $\boldsymbol{(x_o, y_o)}$, we can define the family of lines that goes through that point as: $\boldsymbol{r_\theta = x_o.\cos\theta + y\sin\theta}$ and it means that each pair $\boldsymbol{(r_\theta, \theta)}$ represents each line that passes by $\boldsymbol{(x_o, y_o)}$. The other point here is for a given line $\boldsymbol{(x_o, y_o)}$, we plot the family of lines that goes through it, and we get a sinusoid. For instance, for $\boldsymbol{x_o = 8}$ and $\boldsymbol{y_o = 6}$, we will get the following plot of a $\boldsymbol{(\theta - r)}$ plane [29].

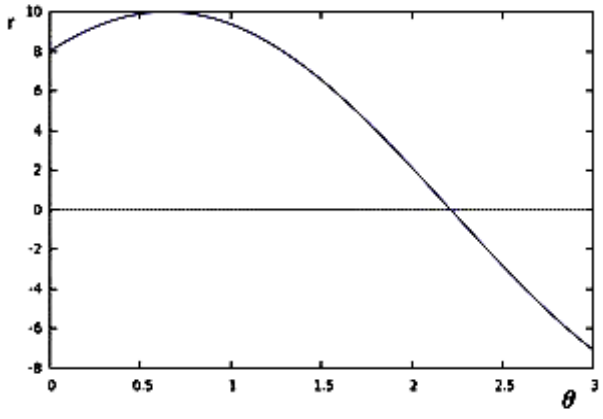


Figure 3.5 a (θ −r) plane

We can do the same operation above for all the points in an image. If the curves of different points intersect in the plane **(ɵ −r)** that means that both points belong to a same line. The following plotting example shows the plot for two more points. The a line can be detected by finding the number of intersections between curves and when more curves intersects, that means the line represented by that intersection have more points and we can define a threshold of the minimum number of intersections needed to detect a line [29].

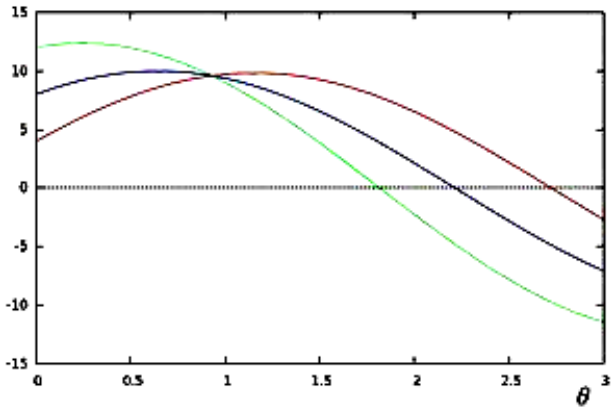


Figure 3.6 a (ɵ −r) plane with more than one point and the intersections represent the line

This is what the Hough Line Transform does. It keeps track of the intersection between curves of every point in the image. If the number of intersections is above some *threshold*, then it declares it as a line [29].

3.3 Segmentation Techniques

Document images from real life contains different components such as text, graphics, tables, logos, columns, etc. To recognize these document images, there is a need to detect and segment text/graphics, columns, tables, lines, word and characters accurately to identify components that potentially represents the document image to make a recognition performance better [26].

Segmentation is the inherent part of an OCR system which is all about segmenting the text/graphic, tables, column regions, text lines, words and ultimately the characters in document images properly for better recognition performance because the performance of remaining other steps of OCR systems relies on this stage [59]. The objective of segmentation is to identify the

meaningful constituent regions which can be a text, graphic, halftone or continuous pictures, etc. [20] [60].

There are a lot of problems faced among the studies of Amharic OCR systems. The major problems for low in accuracy of text segmentation techniques as mentioned by different researchers can be language dependence or vice versa. For example; the overlapping nature of Amharic scripts, broken characters due to the degradation of printed real-life documents, connected characters, different noises such as ink-bleeding and erroneous results of preprocessing methods can also create those difficulties, for example the error in underline detection and removal can also cause segmentation error [26] [43] [45].

As it has been discussed so far, segmentation of document image occurs at two levels. On the first level, text blocks, columns, pictures, tables, logo and other parts are separated which is named as *page segmentation* and the second level involves text lines, words and characters segmentation or extraction from the segmented text block is performed which is named as *text segmentation*. One of the benefits of applying page segmentation on document image is the preservation of small parts of documents so that we can produce smaller blocks of documents without horizontal scrolling. Further, this small block can be sent for further processing instead of the whole document [36].

The task of page segmentation is to divide the document image into homogenous zones, each consisting of the only one physical layout structure (text, graphics, pictures, tables, etc). Therefore, the performance of OCR systems depends heavily on the page segmentation techniques used. To this end, several algorithms have been proposed [36].

The segmentation techniques that are explored and tested for the purpose of both page and text segmentation are *Hough transform, connected component labeling or analysis, projection profile, Morphological dilation* and they all are experimented in different combinations on real life document images.

3.3.1 Hough Transform

In digital image processing for OCR, one of the major problems is detecting the simple shapes like straight line, circle or ellipse. Most of the previous studies used edge detection for the

detection algorithm to detect graphical items considering noises and shapes are less strong than that of individual scripts or text symbols. However, there are limitations on the edge detection methods that cause isolation or disjoint pixels on the desired curves of the ideal shapes [25].

Therefore, Hough transform i.e. presented in section 3.2.4; address such kinds 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 [8].

In previous section 3.2.4, Hough transform algorithm has been discussed in detail for the purpose of detecting continuous black pixels on a binarized image and removing them for the effective removal of underlines form the document images. The same algorithm is modified and applied here to detect table lines and other possible lines in the document image.

3.3.2 Connected Component Analysis

Another very important technique for document image segmentation is connected component analysis. It is applied in document images with high dimensionality to detect connected regions in binary images [41]. Connected component labeling that is alternatively named as connected component analysis, blob extraction, region labeling or extraction; is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic [30].

Connected components labeling scans all the pixels of document image and groups them into components based on pixel connectivity, i.e. all pixels in the connected component shares similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a gray-level or color labeling according to the component it was assigned to. Extracting and labeling of various disjoint and connected components in an image is central to many automated image analysis applications such as OCR systems [30].

There are two types of connected component labeling; one pass and two pass. The 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 and the two pass scans the image two times. The first pass goes through each pixel and checks each pixel and using these pixel labels,

it assigns a label to the current pixel and the second pass cleans up any mess it might have created. One pass labeling takes high processing time and memory space than two pass [26].

After scanning the image pixel by pixel, in order to identify connected pixels which share similar set of intensity values *V* (*i.e. V = {1} for binary images and range of values for gray level images, for example: V = {51, 52, 53, ..., 77, 78, 79, 80}*.); the labeling operator scans the image by moving along a row until it comes to a point *p* (where *p* denotes the pixel to be labeled at any stage in the scanning process) for which *V={1}*. When this is true, it examines the four neighbors of *p* which have already been encountered in the scan (*i.e.* the neighbors to the left of *p*, above it, and the two upper diagonal terms) [30].

The algorithm below (Algorithm 3.5.) presents the one pass connected component labeling algorithm [64].

Algorithm 3.5: One Pass Connected Component Labeling

INPUT: *Binary or Gray Scale Document Image*

OUTPUT: *Labeled Connected Component*

PROCESS:

1. *Labeling operator scans the image until it comes to point p and If V = {1}, label* ***p***
2. *If pixel p is labeled; four neighborhoods will be examined*
3. *If all four neighbors are 0, assign a new label to* ***p****,*
   else
4. *If only one neighbor has V={1}, assign its label to p,*
   else
5. *If more than one of the neighbors have V = {1}, assign one of the labels to p and make a note of the equivalences.*

Two pass labeling scans the image two times as it has been mentioned earlier and algorithm 3.6 presents the two pass connected component labeling algorithm [64].

Algorithm 3.6: Two Pass Connected Component Labeling

INPUT: *Binary or Gray Scale Document Image*

OUTPUT: *Labeled Connected Component*

PROCESS:

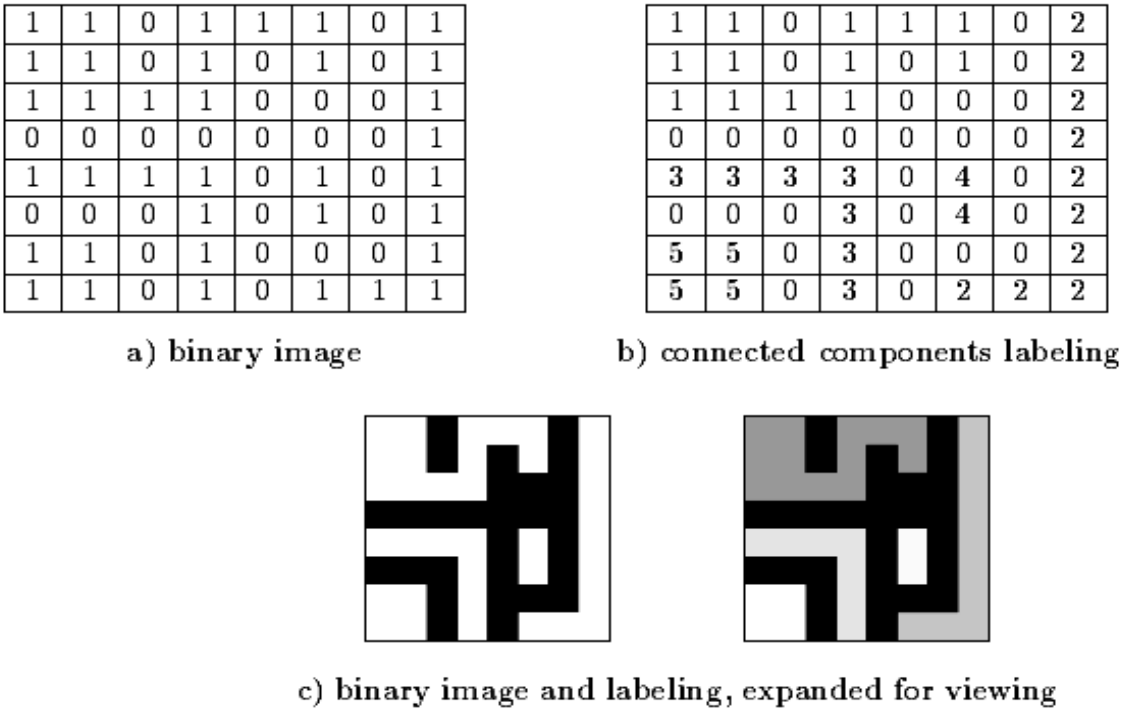
First Pass:

- *Scan the image pixel by pixel*
- *If the pixel is not a background*
  - ✓ *Check neighbors*
    - ❖ *If neighbors already labeled*
      - o *Assign neighbors parent label to the main label*
    - ❖ *If None of neighbors labeled*
      - o *Assign new label to the pixel*

Second Pass:

- *Scan the image pixel by pixel*
- *If the pixel labeled*
  - ✓ *Get labels parent*
    - ❖ *If parent pixel is in patterns list*
      - o *Add to existing list*
    - ❖ *Else*
      - o *Add to new list*

The following figure 3.7 presents an example of the connected component labeling applied on binary image.



1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

a) binary image

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	**3**	**3**	**3**	0	**4**	0	2
0	0	0	**3**	0	**4**	0	2
5	**5**	0	**3**	0	0	0	2
5	**5**	0	**3**	0	2	2	2

b) connected components labeling

c) binary image and labeling, expanded for viewing

Figure 3.7: A binary image with five connected components of value v = {1} [30]

3.3.3 Morphological Dilation

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image [41].

One of the most basic morphological operations is *Dilation;* which adds pixels to the boundaries of objects in an image, while the opposite process *Erosion* removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation operation, the state of any given pixel in the output image is determined by applying a rule to the

corresponding pixel and its neighbors in the input image. The dilation rule used to process the pixels is; the value of output pixel is the *maximum value* of all pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1 [41].

Figure 3.8 gives an example on how the dilation algorithm works in binary image. Note how the structuring element defines the neighborhood of the pixel of interest, which is circled. The dilation function applies the appropriate rule to the pixels in the neighborhood and assigns a value to the corresponding pixel in the output image. In the figure, the morphological dilation function sets the value of the output pixel to 1 because one of the elements in the neighborhood defined by the structuring element is on. Structuring element is an essential part of the dilation operation which is used to probe the input image. A structuring element is a matrix consisting of only 0's and 1's that can have any arbitrary shape and size. The pixels with values of 1 define the neighborhood.

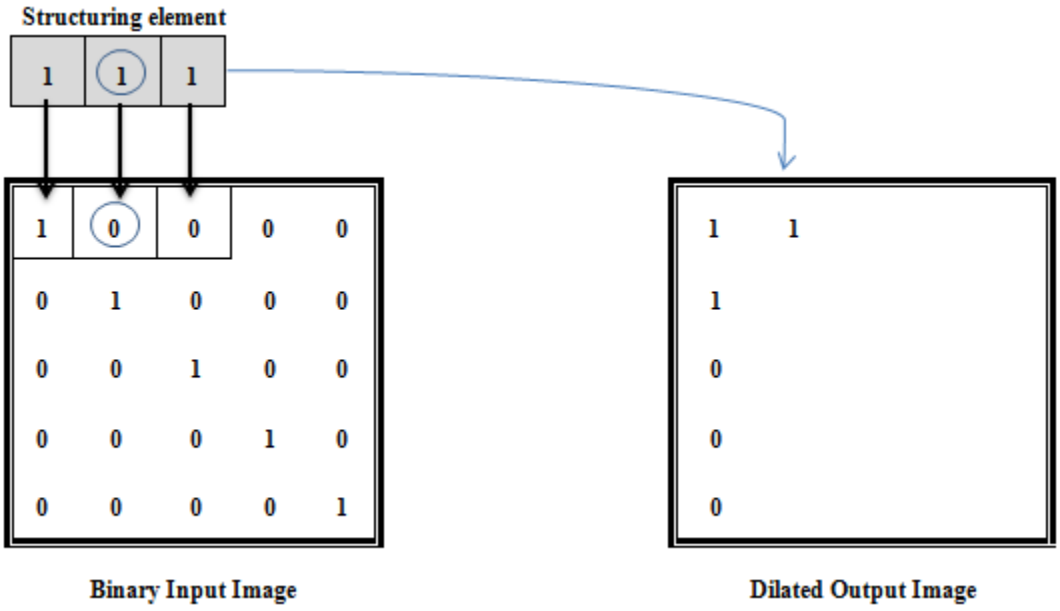


Figure 3.8: Morphological Dilation of a Binary Image

3.3.4 Text Segmentation

For the segmentation of text lines, words and characters form the document image that is the most critical component and an input to the next phases of OCR system, different types of

methods have been used so far by various scholars. Text segmentation extracts lines, words and finally into characters from the text document images [20].

Most of the Amharic OCR studies including Worku [66], Dereje [21], Million [44] and Michael [43] applied stage by stage segmentation technique which uses projection profile to isolate lines, words and characters from the document image. Text line segmentation is an essential stage because inaccurate segmentation will cause errors in recognition stage.

Projection profile method applies projected histograms drawn according to the frequency (count) of black pixels in the document image. It operates by summing the pixel values along the horizontal and vertical direction of the document in which gaps between the text lines and text scripts are analyzed through searching the projection valleys [9].

There are two main advantages for this approach in context of historical document. First it doesn't require binarization of image which makes it directly applicable on the gray scale images and the second is it is very robust to noise and other degradations. However, it is very sensitive to line skewness. Hence skew correction must be performed as a preprocessing stage before line segmentation [9].

3.3.4.1 Line Segmentation

Line detection operation on the detected text block can be performed using the horizontal projection which counts and sum horizontal black pixels. When the frequency of black pixels in a row reaches maximum, the projection histogram creates peaks in the histogram. On the other hand, when their count is low a valley will be created which denotes a boundary between two consecutive lines (See figure 3.9) [20].

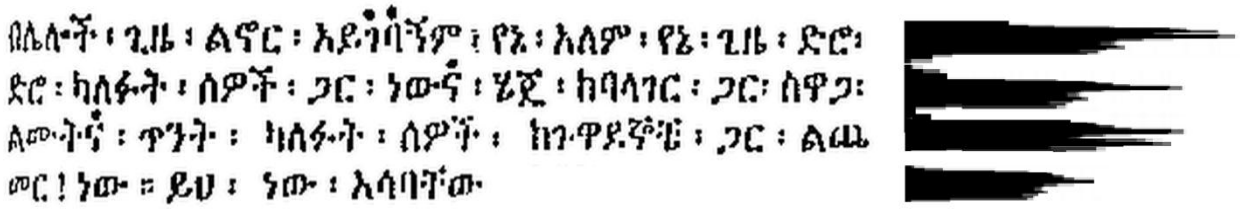


Figure 3.9: Example of Horizontal projection profile on a sample text image

Another method that is tested in this study is morphological dilation in horizontal direction for the segmentation of text lines. As discussed in section 3.3.3, dilation performs connection of black pixels in a direction based on the window size provided.

3.3.4.2 Word and Character Segmentation

Vertical projection profile method is applied after the horizontal projection result that is segmented text line. Each line will be vertically projection profiled to spot the space between individual characters or words [20] [40]. Another modified version of this technique is recursive projection profiling [40]. The following figure 3.10 presents an example of vertical projection.

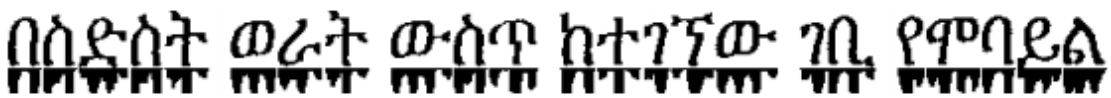


Figure 3.10: Example of Vertical projection profile on a sample text line taken from [43]

After the counting result of vertical and horizontal projection, the white space analysis must be made to decide text line position and vertical projection performs space analysis to detect words and characters from the detected line. The biggest task is deciding a threshold value to white spaces [40].

For the segmentation of word images, another method that is tested in this study is morphological dilation as discussed in section 3.3.3 by connecting black pixels in a direction based on the threshold provided and CC analysis finds the connected sections to determine words. It only differs with the line segmentation application of dilation by its threshold value.

3.4 Performance Evaluation

The performance of noise removal techniques is measured using Mean Squared Error (MSE) and Peak Signal to noise ratio (PSNR). *MSE* is a measure of an average of the squares of the errors, which is the difference between the estimator and what is estimated. *PSNR* is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [64].

For measuring the performance of segmentation techniques, various techniques are available. For this study, the performance is measured by manually counting the expected correct segmentation and also the erred to calculate the Accuracy percentage; as used by Michael [43].

The accuracy of the OCR system is measured by common and widely used methods such as recognition rate, error rate and rejection rate of test results from both testing and training sets. Recognition rate measures the proportion of correctly classified characters from the total characters and conversely, error rate measures the proportion of characters erroneously classified. Rejection rate measures the proportion of characters which the system was unable to recognize [23].

CHAPTER FOUR
EXPIRMENTATION

The main aim of this study is to experiment different image preprocessing, page and text segmentation techniques on real life Amharic document images with different levels of degradation. The selected techniques are finally integrated with previously developed Amharic OCR system to evaluate the change in performance.

For the experimentation purpose, TOSHIBA Intel(R) Core(TM) i3 CPU M380 @ 2.53GHz (2 CPUs), 4GB RAM and Windows 7 Ultimate operating system were used. MATLAB™ image processing toolbox R2013 and C# programming language using Microsoft Visual Studio 2013 tool are used for developing prototype and integration.

4.1 Dataset Collection and Image Acquisition

The goal of this research is to explore better image preprocessing and page segmentation techniques for real life Amharic document images to segment the tables, columns, graphics, logos, and text areas from those document images. The segmented text area are preprocessed and fed into the text segmentation method for the extraction of text lines, words and characters that can be input for the remaining stages of the OCR system. Therefore, Amharic document images with different level of degradation, containing graphics, pictures, tables, columns are collected and merged with the dataset prepared by Biniyam [13]. This stage is the first stage in OCR systems concerned with the preparation of sample training and testing in real life documents.

The dataset collected by Biniyam [13] contains real life document images with different levels of noises from sources such as newspapers, magazines, historic documents and religious books. It enables to evaluate if the proposed preprocessing and segmentation techniques are insensitive to noise and the variety of writing styles, sizes and fonts which is important for the problem domain. These documents are also believed to have real-life features that occur in most Amharic documents. The dataset doesn't contain handwritten and typewritten document images; rather it only contains printed documents. The following Table 4.1 summarizes document image collections used in this study.

Documents Collected By	Type and size of documents		
Biniyam [13]	**Noise Level**	**Document Type**	**Pages Extracted**
	Low	Magazine	2
		Qidassie Mariam	5
	Medium	Book	3
		Qidassie Yohannes	2
	High	Book	5
		Newspaper	1
	Very High	Book	6
		Qidassie Yohannes	2
		Sum	**26**
New added by the researcher	**Documents Contain**	**Document Type**	**Pages Extracted**
	Tables	Newspaper	7
	Pictures	Newspaper	6
	Columns	Newspaper	10
	Titles	Newspaper	5
	Skewed	Fiction	10
		Sum	**33**
		Total	**59**

Table 4.1: Summary of datasets used in the study

For the conversion of the newly collected manual documents into their digital format, flatbed scanner i.e. HP Scanjet G4050 device with windows 7 Fax and Scanning software is used. The documents are scanned in grayscale level with zero brightness and contrast levels having a resolution of 300 dpi. This resolution is selected because such value is optimal for keeping text textures of font sizes. The scanned images are stored as BMP image format (See Annex II for sample scanned document images).

4.2 Preprocessing

Preprocessing is a vital stage in OCR systems which have a direct impact on the accuracy of the remaining successive stages of recognition process. Many individual preprocessing algorithms are available and they are applied based on the problem at hand. For this study, the preprocessing steps such as skew detection and correction, page segmentation, noise detection, binarization and underline removal are explored to see their effect on real-life Amharic documents.

4.2.1 Skew Detection and Correction

Skew angle of document image is deviation of the baseline of text line from horizontal direction. The dominant orientation of the text lines in a document page determines the skew angle of that page. In document images, a horizontal straight line can be drawn through a group of characters, symbols and words that are adjacent and relatively close to each other. Skew detection and correction is an important image preprocessing steps in character recognition that must be performed before other stages of OCR system because rotation of document images had an impact on remaining phases.

In this work, the skew checker class that implements document skew checking based on Hough line transformation by searching for text base black lines of text bottoms followed by white line below on the gray scale document image input is adopted. The algorithm supposes that a white-background document is provided with black letters on a text document images.

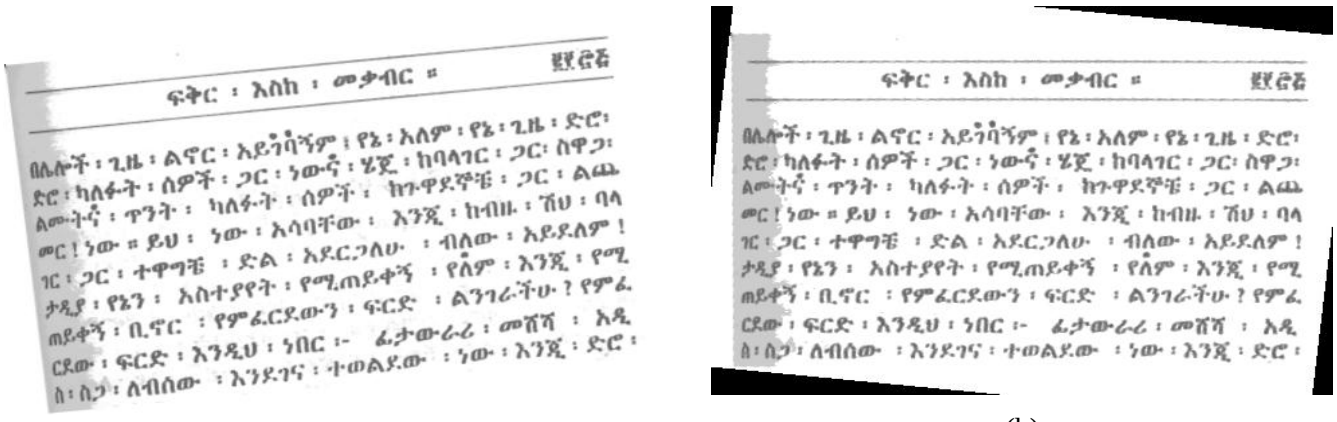
To implement the algorithm Microsoft Visual Studio 2013 tool is used by applying Visual C# programming language. The `AForge.Imaging` and `AForge.Imaging.Filters` super class of `DocumentSkewChecker()` class is imported to apply `GetSkewAngle()` for the detection of rotation angle ɵ. And if the angle is below 90^0, it will correct it by rotating it by -ɵ using `RotateBilinear()` class. The code used to implement skew detection and correction is given in Snippet 4.1.

Snippet 4.1: Implementation DocumentSkewChecker and RotateBilinear

```
// Detect image the skew angle
DocumentSkewChecker skewChecker = new DocumentSkewChecker();
double angle = skewChecker.GetSkewAngle(imageData);
// create rotation filter and rotate image
RotateBilinear rotationFilter = new RotateBilinear(-angle);
rotationFilter.FillColor = Color.White;
image.UnlockBits(imageData);
```

The proposed skew detection and correction algortihm first converts the image to 2D color space that is grayscale. The aim of this technique is to correct rotation occurred during image acquisition process and due to this fact, skew angle cannot be greater than 90^0.

Using this method, all the documents that are scanned are processed and checked. From the result, document images that are correctly scanned are returened as they are 0^0 skewed and the proposed techniqe return the image itself. For skewed images, it perfectly returned the corrected document image. Figure 4.1, shows the result of `DocumentSkewChecker` class.



(a) (b)

Figure 4.1: Result of DocumentSkewChecker

(a) Skewed original document image *(b) De-skewed (processed) document image*

The algorithm works well in correcting the skew angles of document images with a white background. However, the result shows that there are some limitations on correctly detecting the skew angle of document images with a colored background.

4.2.2 Automatic Page Segmentation

In this study, before applying denoising, binarization and other processing methods over the whole image, page segmentation techniques are applied. This is due to the success of OCR and other systems that needs document image analysis depends on the proper segmentation of texts, graphics, headings, tables, columns and text area.

Page segmentation is performed first to detect text region from non-text region so that the subsequent image processing are only applied over the text area. This will help other remaining stages of OCR systems in several ways such as removing noises from the border of text images by excluding the non text area.

Some MATLAB Built-in methods are integrated with Visual C# classes and libraries to develop an algorithm using various techniques such as *Hough transform, morphological dilation, connected component (CC) analysis* and *CC width, height and analysis.* The developed page segmentation techniques are used for segmenting tables, lines, graphics, text areas, column blocks, and titles from the document images.

4.2.2.1 Table Segmentation

A MATLAB function that uses Hough transform method is developed and integrated with visual C# for the detection and removal of table lines from the given document image. Table segmentation is essential because it has a direct impact on the results of an OCR systems.

Hough transform is also used for the detection and removal of underlines, overlines or any other possible lines. This is because the main concern of recognition is on the text regions rather than non text regions of the image. Therefore, we need to identify the table, label the table lines and remove them so that the document contains only text.

Table line detection is performed by the following MATLAB function and the sample code is given in Snippet 4.2 (see Annex III).

Snippet 4.2: Implementation Hough Transform for Table and Lines Segmentation

```
function [ image ] = tableLines(image, cutoff, line_thresh)
    %% receive and image, binerize it and Transform
    Binary_image = ~im2bw(image);
    BW = imcomplement(Binary_image);
   [H,T,R] = hough(BW);
    xlabel('\theta'), ylabel('\rho');
    %% setting a peak value for lines
    P = houghpeaks(H,100,'threshold',ceil(0.3*max(H(:))));
    x = T(P(:,2)); y = R(P(:,1));
    %% set lines by filling little gaps
    lines = houghlines(BW,T,R,P,'FillGap',1,'MinLength',cutoff);
end
```

The developed MATLAB function "`tableLines(image, cutoff, line_thresh)`" receives the rgb or grayscale image to convert it into the binary image using the function `hough()`. The function is used to detect line objects found on the binary image by testing diverse peak (P) values. In previous study made by Gedion [26], some text parts with condensed foreground pixels were considered and detected as a horizontal line (see figure 4.3).

This problem is enhanced by modifying the algorithm by adding two additional properties of `houghlines();`the `'FillGap'` and `'MinLength'`. The `'MinLength'` property enables the algorithm to set the minimum gap between broken lines and through expirment the `'Cutt_off'` value is decided through the iterative expirment and 35 is identified as a better threshold. The `'FillGap'` property checks the the gap between characters and fills those gaps using the threshold that is 1 pixel based on the expirment. Also better results were registered by adjusting peak value to exclude the text parts from being considered as lines. However, it missed

some lines from the document image. Figure 4.2 presents results of expirmentation of Hough transform on noisy images that contain tables and other lines.

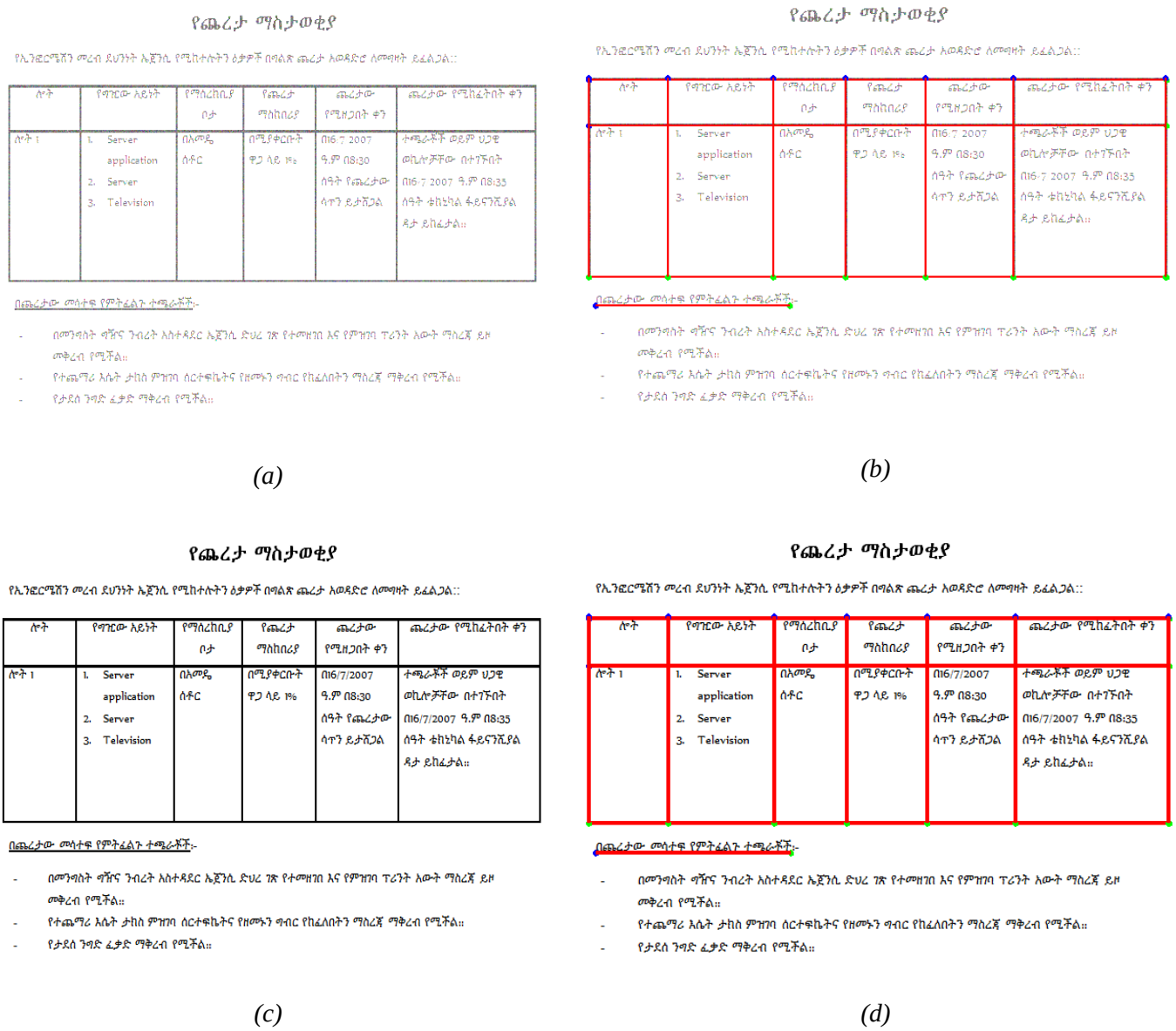


Figure 4.2: Implementation of Hough transform on images with different types of noises

(a) Original image with blurred noise *(b) Detected table and other lines of image (a)* *(c) Original image with Gaussian Noise* *(d) Detected table and other lines from image (c)*

For comparing the previous algorithm used by Gedion [26] to detect tables and lines in his study on Amharic DIR system, the following result were obtained (see figure 4.3). The expirment shows; his algorithm detects the part of the text as a line and also when the threshold is fixed it missed some of the lines.

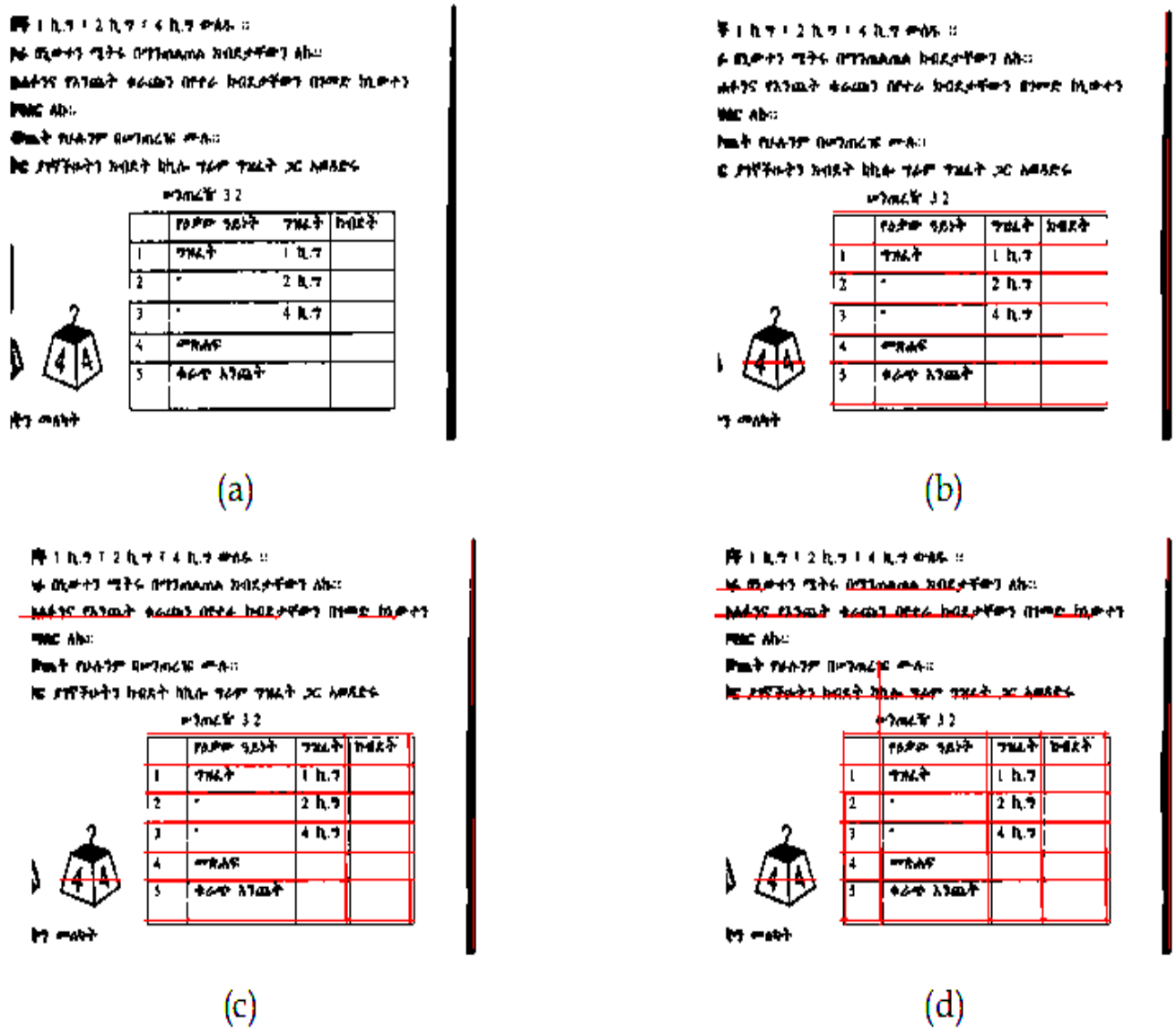
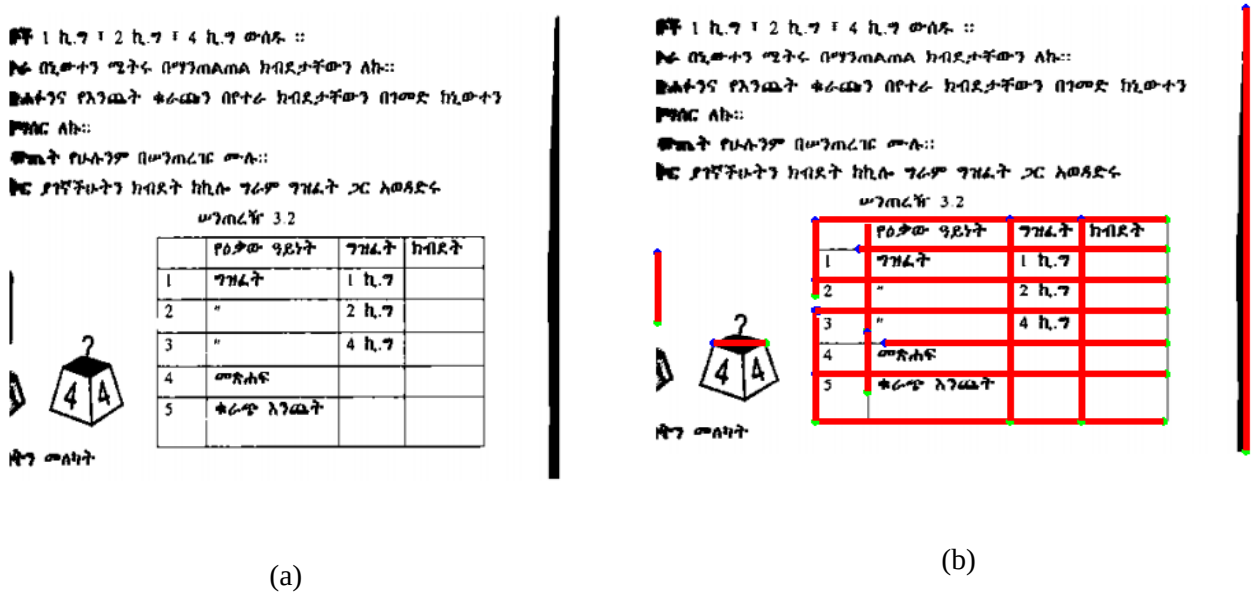


Figure 4.3 – Gedion's [26] result on the implementation of Hough Transform with Different Thresholds to detect and delete lines from image:

(a) Original image, *(b) Identify some part of tables only,* *(c) Some text parts are detected as lines* *(d) Detect all lines and some more text areas as lines*

The proposed method registered better performance. However, some of the lines with shorter length and broken lines are missed by the algortihm. Therefore, it needs further investigation to detect all the lines of those types. The experimental result for the same document image used by Gedion [26] is presented in the following figure 4.4.



(a) (b)

Figure 4.4 – Implementation of Proposed Table Line Segmentation on Gedion's [26] document image

(a) Original image that contains table *(b) Result of table and other possible lines detection*

4.2.2.2 Text/Graphic Segmentation

After table segmentation stage is done, the next page segmentation technique is Text/Graphic Segmentation. It is another essential procedure that must be applied over the image before other stages of OCR system [45]. Text and graphics are the major constituents of any document image. Segmentation of graphics is essential for better OCR performance and it is particularly difficult in the context of graphics made of small components (dashed or dotted lines etc.) having features similar to texts. This study proposes a robust technique for segmenting graphics from Amharic document images.

The text/graphics segmentation algorithm is developed by applying Morphological Dilation and Connected Component (CC) analysis technique for the separation of text and non-text components. The algorithm takes advantage of the fact that characters have limited numbers and sizes whereas the shapes of non-text elements are unlimited.

CC labeling or analysis is done by the MATLAB built-in function or method named as `bwconncomp()`that is used to identify the connected regions or components in a binary input image. Snippet 4.3 presents the MATLAB code used for CC labeling. Moreover, `bwlabel()` is used to label the connected components of the given binary image.

Snippet 4.3: Implementation of Connected Component Labeling

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

Both of the `bwconncomp()` and `bwlabel()` extract the connected components from binary image regions and assign it to the `cc` variable and the number of detected connected components on the image is assigned to `num` variable. On this study, both 4 and 8 connectivity of pixels are used in order to find the CC of the given image depending on its purpose.

The difference between 4 and 8 CC connectivity labeling is how the algorithm defines connected pixels. For example, for the pixel P, 4 connectivity only checks the four neighbors, called *direct-neighbors* i.e. NORTH, SOUTH, EAST and WEST neighbors of P whereas 8 connectivity is known as *indirect-neighbor*s checks all the surrounding pixels around P including diagonal pixels and figure 4.5 demonstrates their effect on a given binary pixels. The labeled pixels represent pixels that are considered as connected to the central pixel in both approaches.

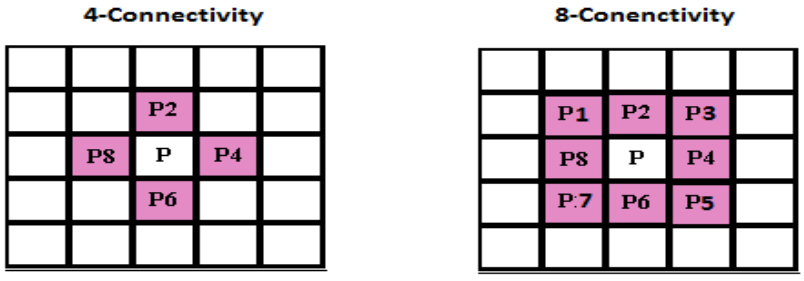
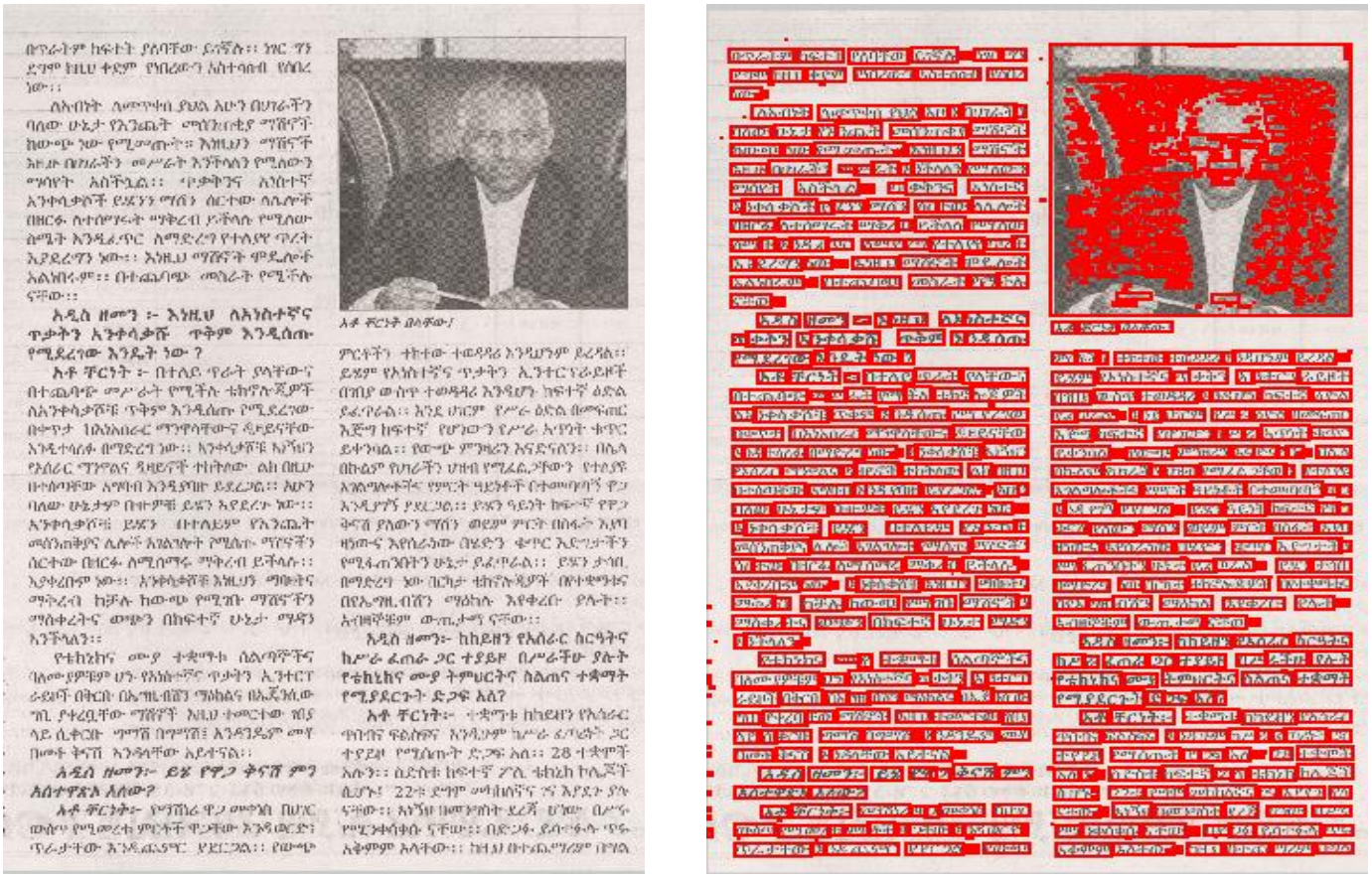


Figure 4.5 – Checking mechanism of 4 and 8 CC Connectivity Labeling

The proposed Text/graphic segmentation technique is based on the CC analysis with a few amount of dilation (both in a vertical and all directional); are used for connecting the broken pixels of the components of the document image. Figure 4.6 shows the result of CC labeling on the document image that contains both text and graphics.



(a) (b)

Figure 4.6 – The result of CC labeling on

(a) The Original text/graphic document image *(b) Result of CC labeling on image (a)*

As we can see from the above CC labeling result, all of the components in the document images are labeled. Dilation (see figure 4.6 (b)) connects the characters and it forms words and other tiny and large objects of the graphics. The next step performed on CC labeled document image is height, width and area analysis of component to decide threshold value. The threshold is used to separate the text or graphics or other non-text elements of the image such as tables, lines and so

on. However, the experimental threshold decided is based on the aim of segmenting texts from graphics.

In order to decide whether the component is the tiny or large object, CC height and width analysis is used. In this study, dilation connects the characters and disconnected sections of the graphics and the values of the area is computed for each component and saved using an array 'size_info [height, width, area]'. The average area of every component is also computed to decide the threshold. 5000 is found to be a better threshold based on the experiment conducted. The following Snippet 4.4 shows the MATLAB code for thresholding and labeling technique to separate a text from the image sections.

Snippet 4.4: Implementation Text\Graphics CC Labeling

```
% storing height, width and area of each component label on array
and labeling the
 size_info = [];
 cc = 1;
 for cnt = 1:num
     x = Ibox(:,cnt);
     size_info (cc,1) = x(3,:,1);
     size_info (cc,2) = x(4,:,1);
     size_info (cc,3) = x(3,:,1) * x(4,:,1);
     cc = cc + 1;
     if (size_info(cnt,3) > 5000)
        rectangle('position',Ibox(:,cnt),'edgecolor','r');
     end
 end
```

However, characters with larger font sizes such as drop caps and smaller graphics having smaller sizes are considered as graphics. It needs further investigation on the area to solve such problems. The result of the experiment on a sample images from the dataset is presented on figure 4.7 (i) and (ii).

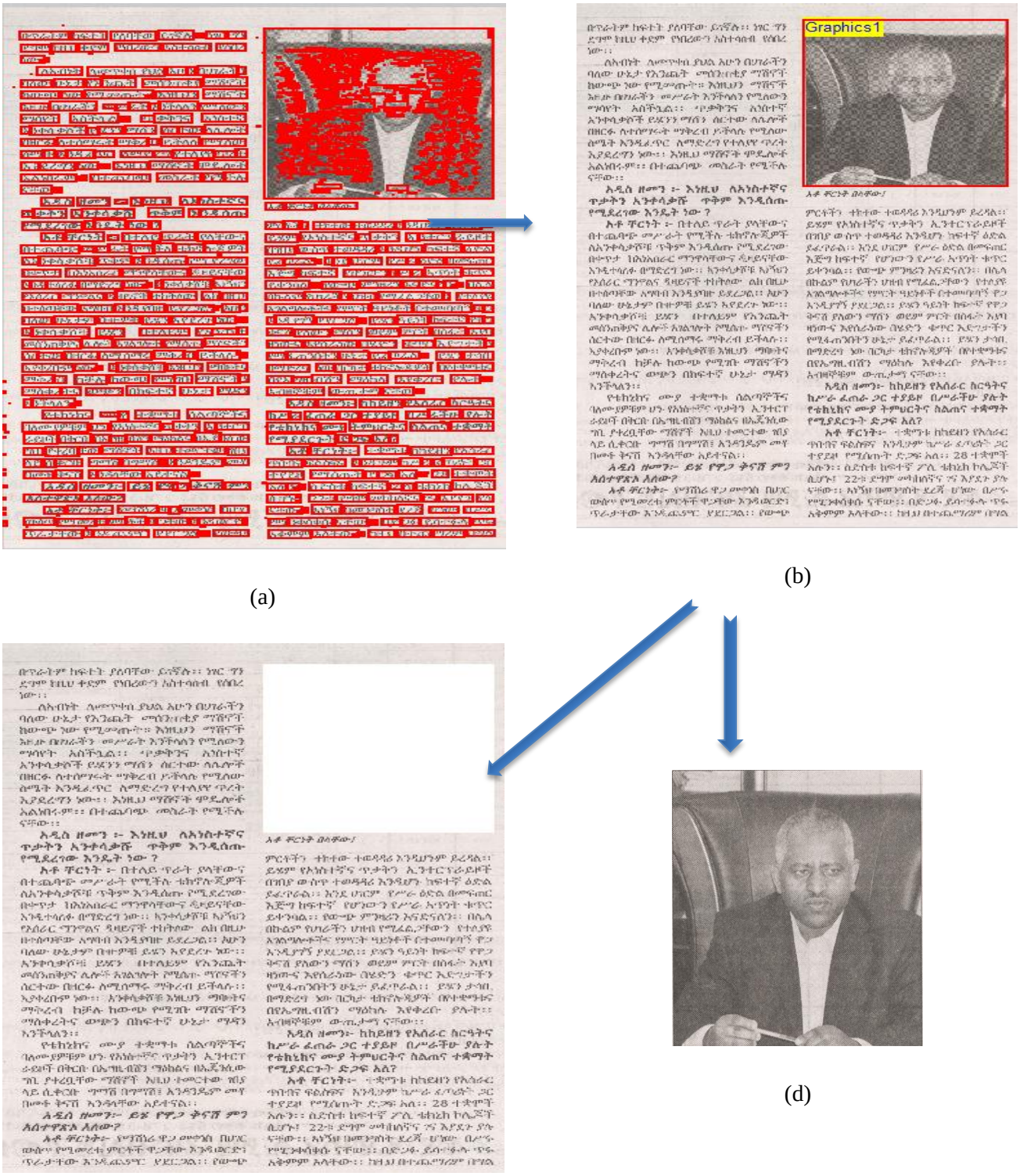
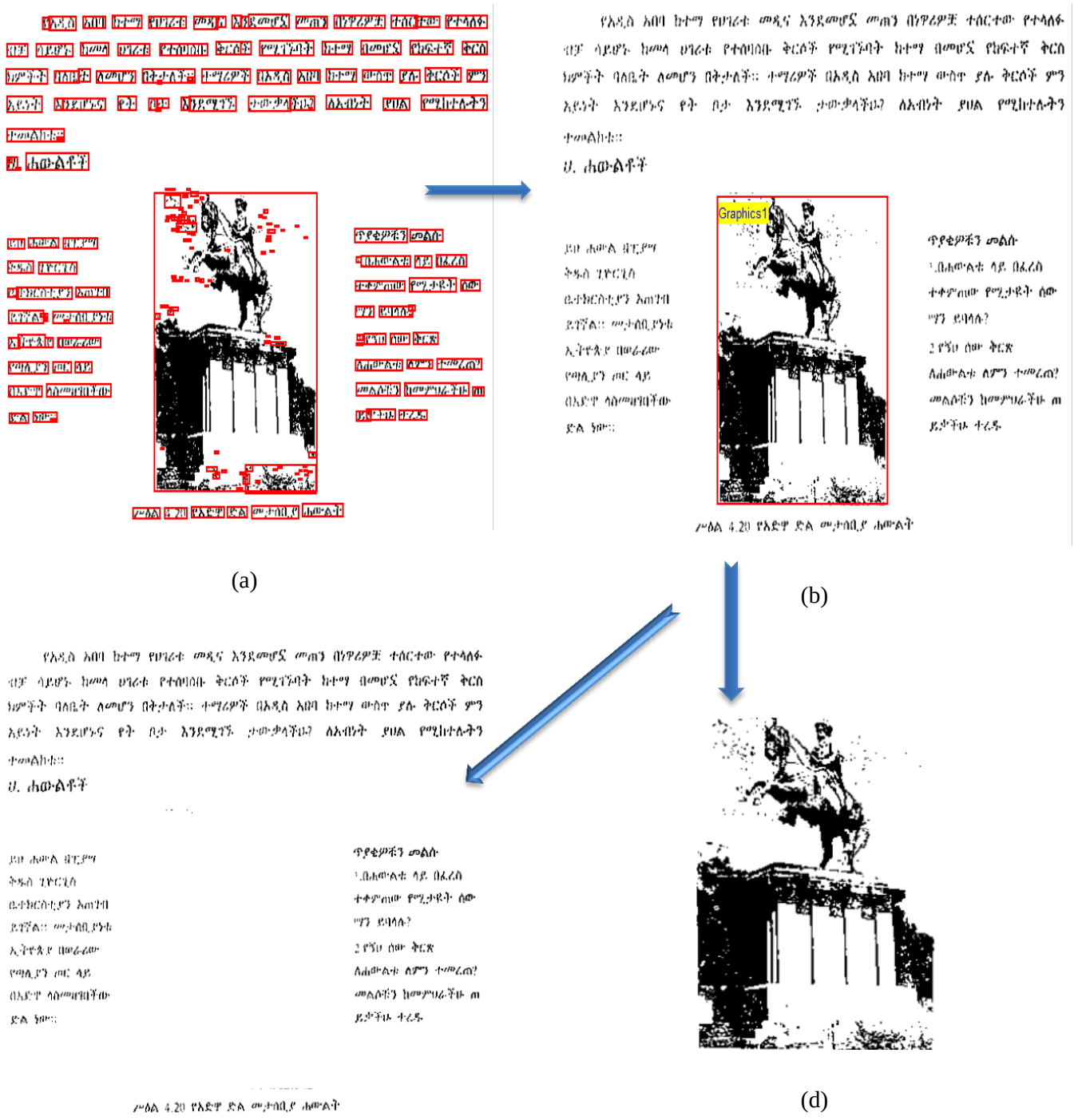


Figure 4.7 (i) – Experimental result of the CC Labeling and Thresholding

(a) CC Labeled Image, (b) Image after CC Label Thresholding (c) Segmented Text area of the image (d) Segmented Graphics from the image



(a) (b) (c) (d)

Figure 4.7 (ii) – Experimental result of the CC Labeling and Thresholding

(a) CC Labeled Image, (b) Image after CC Label Thresholding (c) Segmented Text area of the image (d) Segmented Graphics from the image

4.2.2.3 Column Block and Title Segmentation

The previous text/graphics segmentation has produced two types of document images; text area and graphics image. However, for the document images that might contain multi columns such as newspapers, books, etc.; it is essential to detect and segment those regions for benefits such as layout formation for the final output formatting and for removing noises that occurs in the borders of real life document images.

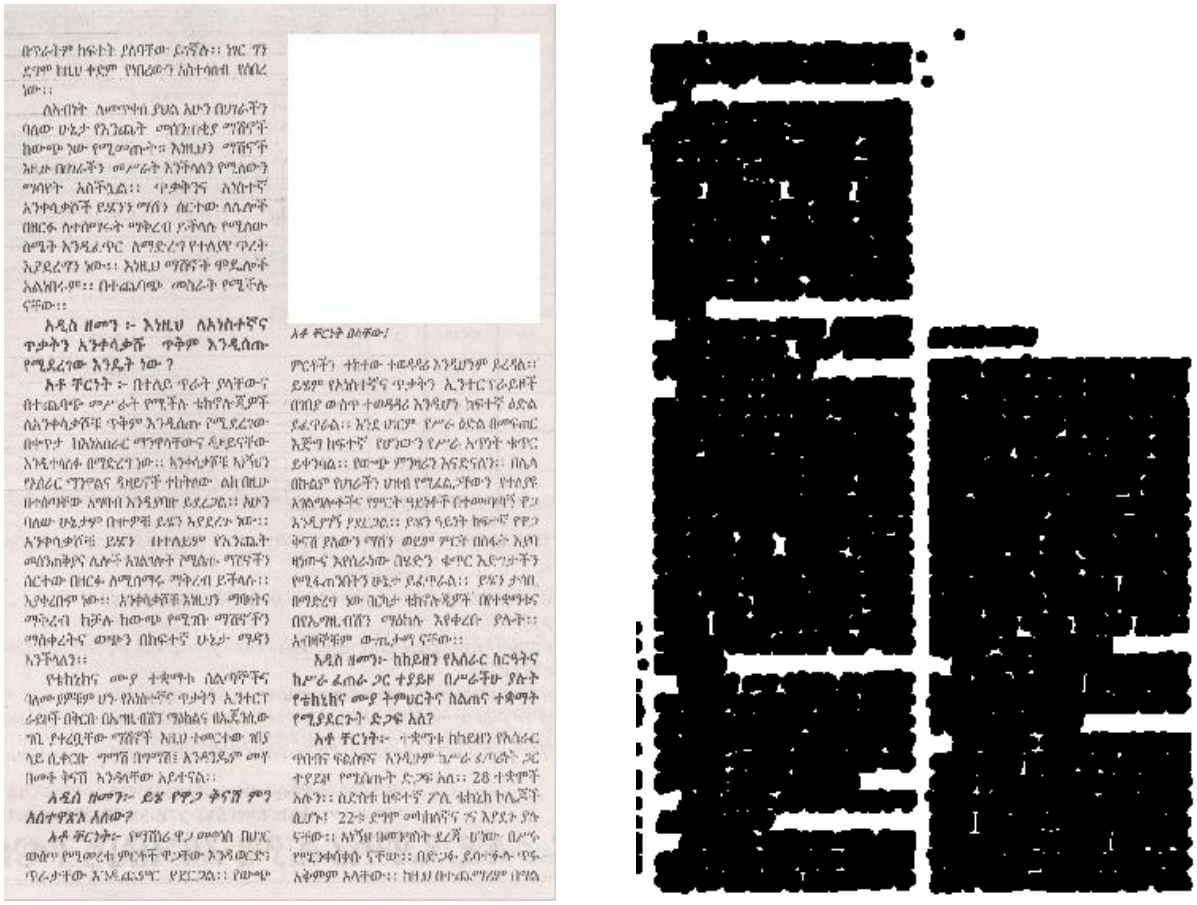
As it was discussed in section 3.3.3, this study developed a technique based on morphological dilation and CC analysis to identify columned blocks from text document images. Both dilating image in all direction (multi-directional) and also in a vertical direction is tested.

The implementation of multi-directional dilation is performed by the following MATLAB function given in Snippet 4.5.

Snippet 4.5: Implementation of Dilation (multi-directional)

```
function [dilatedImage] = dialate(binary_image, dilation_thresh)
    dilatedImage = bwdist(~binary_image) >= dilation_thresh;
end
```

The function accepts two parametrs from the visual C# functions; the binary columned image and the decided threshold value. It returns the dilated binary image based on the input provided. The result shows connected pixels due to the dilation algorithm that connects characters, words, text lines, figures, etc. The algorithm is helpful to separate page headings, columns, and other parts of the image using white space analysis and connected component height, width and area analysis. Figure 4.8 presents the result of dilation in all direction and connected component labeing on the segmentation of columns in columned text images.

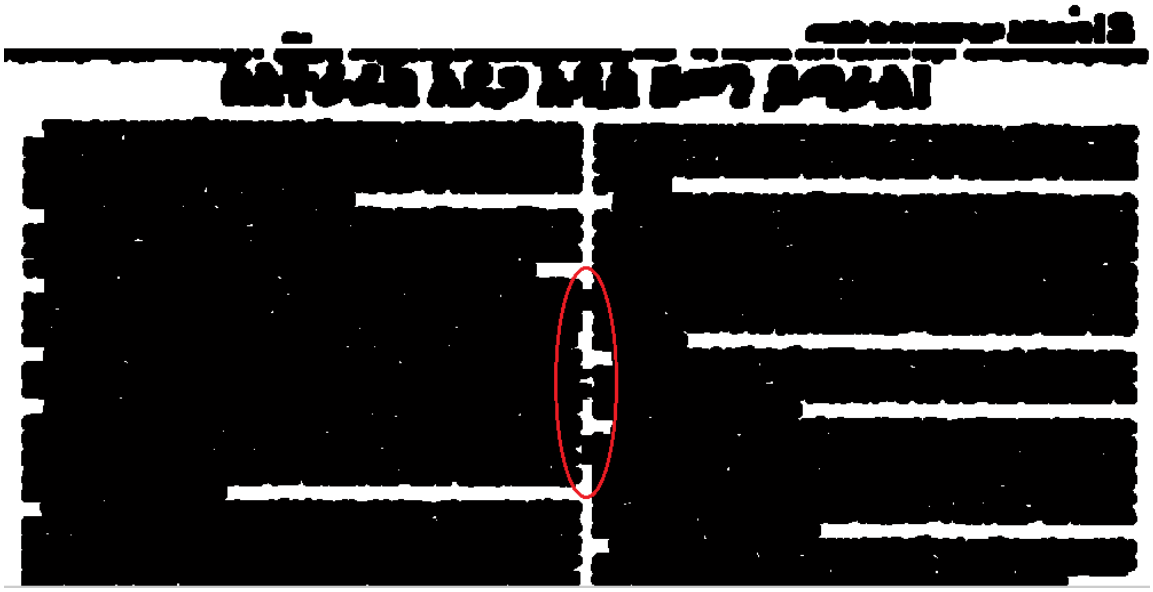


(a) (b)

Figure 4.8 – The Experimental Result of Multi-Directional Dilation on Columned Document Image with Medium Font Size.

(a) Original binarized image, (b) Multi-directional dilation using threshold value of 8

As we can see from the above sample expirment, the pixels of text lines are successfully connected by dilation using the threshold value of 8 in all directions. However, for the columened documents that has smaller font size, the above algorithm connects all the pixels including the white spaces between the columns. Figure 4.9. shows the result of the multi-directional dilation over smaller font sized textul images.



(a) result of multi-directional dilation over the small font sized image

(b) *result of multi-directional dilation over the original image (a)*

Figure 4.9 – The Experimental Result Multi-Directional Dilation Over Columned Document Image with Smaller Font Size.

The red ellipse in the above figure 4.9 (a) presents the unnecessary connections made by multi-directional dilation which causes a problem of connecting the two columns. For solving such issues; this study applied a vertical dilation technique that is used to connect the space between characters, words, text lines, grapics, and other elements that are found in the document image in a vertical direction with a few amount of multi-directional dilation at threshold 1. It is the same

as multi-directional dilation except it is only increasing pixel values in a vertical direction. Based on an iterative expirment, *4x16* window size is selected for dialating the document image vertically. The MATLAB function that is used for the implementation of vertical dilation is given in Snippet 4.6

Snippet 4.6: Implementation of Vertical Dilation

```
function [dilatedImage] = dialateVertically(binary_image)
    se = [0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0;
          0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0;
          0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0;
          0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0; 0 1 1 0];
    dilatedImage = imdilate(~binary_image, se);
end
```

The above function recives a single binary image parameter and returns vertically dilated image. Expirmental result of the algorithm over the original image of figure 4.9 is presented in figure 4.10.

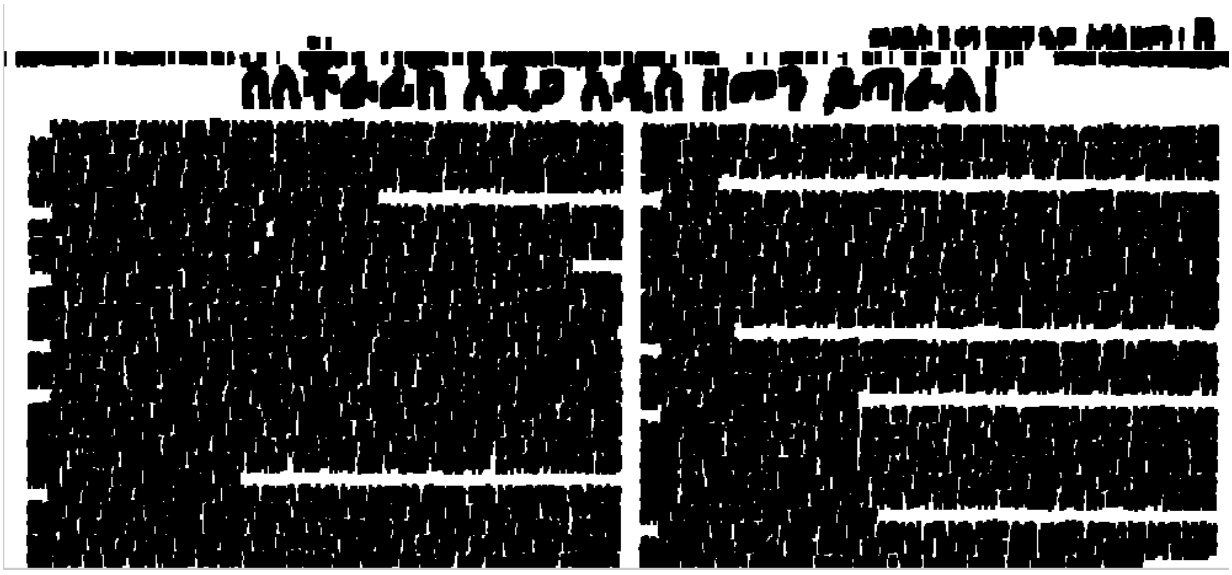


Figure 4.10 – The Experimental Result of Vertical Dilation Over Columned Document Image with Smaller Font Size.

From the above expirment, the proposed dilation gives better result by keeping the space between column blocks. The next step after dilation is performed over the above image inorder

to find the connected components. The height, width and area analysis is made for deciding the thresholding for column block identification using the code shown in Snippet 4.7.

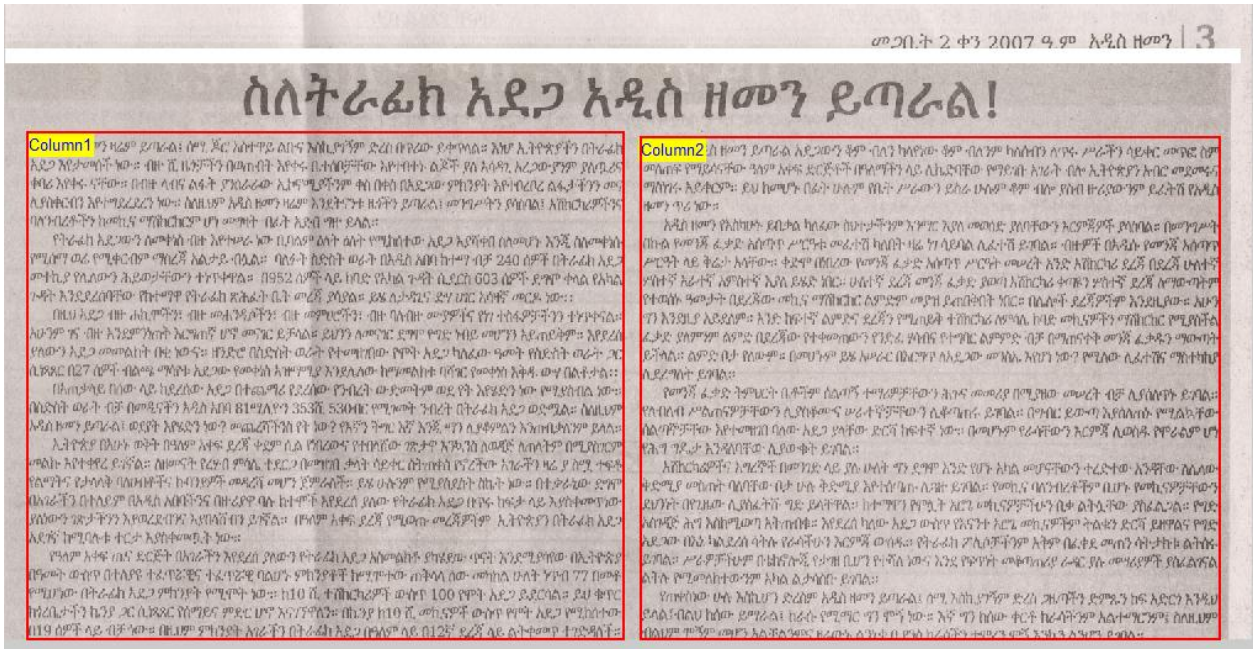
Snippet 4.7: Implementation of Automatic Thresholding for Column Block Identification

```
size_info = [];
sumArea = 0;
for cnt = 1:num
     component_area = component_width * component_height;
     size_info (cnt,1) = component_width;
     size_info (cnt,2) = component_height;
     size_info (cnt,3) = component_area;
     sumArea = sumArea + component_area;
end

maxArea = max(size_info);
for cnt = 1:num
     x = Ibox(:,cnt);
     if (size_info (cnt,2) > maxArea(1,2)/4 && . . .
        size_info (cnt,1) > maxArea(1,1)/4)
        rectangle('position',Ibox(:,cnt),'edgecolor','r');
     end
end
```

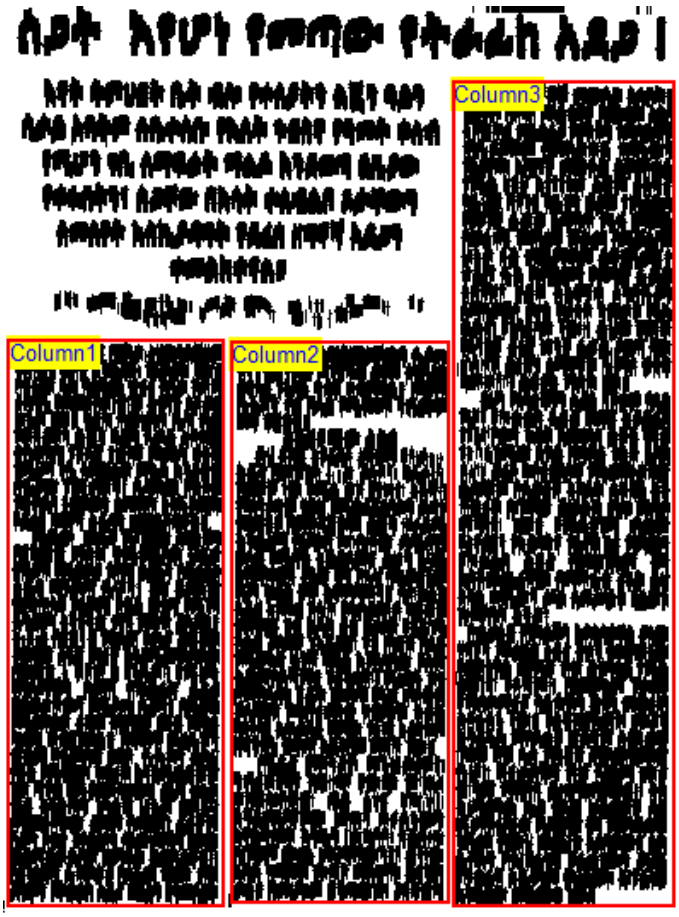
Based on the above automatic thresholding calculation, the area of each connected components is computed and stored in the array with the width and height of the component. The algorithm select the maximum area by assuming the tallest and widest components are column blocks. It is assumed that the shortest height and width of the column block is one-fourth (¼) of the maximum height and width. The experimental results of column block detection is presented in figure 4.11(i) and (ii).

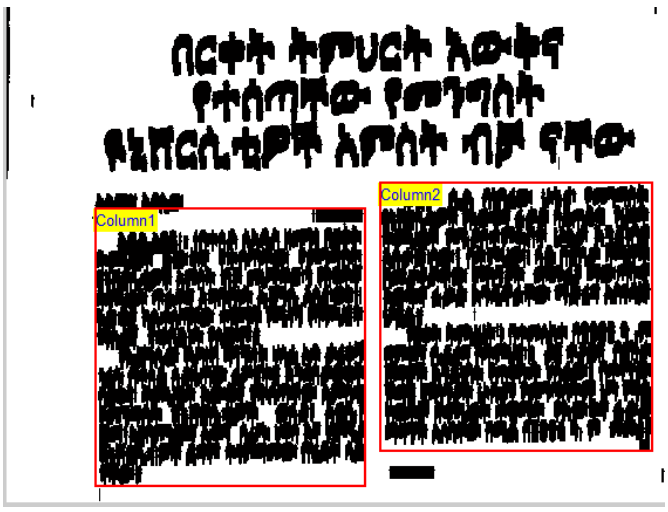
(a) Result of automatic thresholding over the CC labeled dilated image



(b) Result of application of automatic thresholding over the original image

Figure 4.11 (i) – The Experimental Result of the Proposed Column Block Segmentation Algorithm For Two Columned Document Image



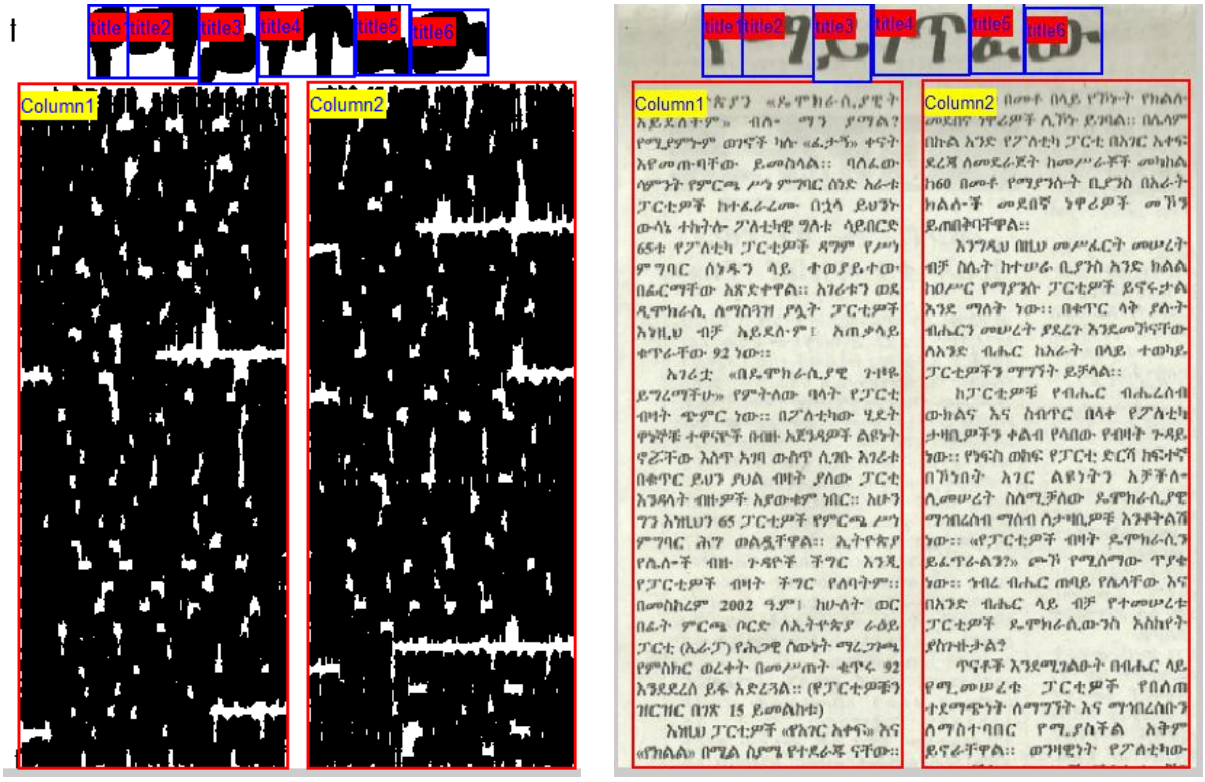


(a) Result of automatic thresholding over the labeled CC

(b) *Result of automatic thresholding over the original image*

Figure 4.11 (ii) – The Experimental Result of the Proposed Column Block Segmentation Algorithm For Three Columned Document Image

As we can see from the above results, the height and width of component is important to decide weather the component is a column, header/title. Noises and other tiny components are are successfully removed and only the text area is segmented. Figure 4.12 presents the result of deciding weather the object is a title or noise.



(c) (d)

Figure 4.12 – Implementation of Dilation and CC analysis over columned images with titled image

(a) Original binarized image, *(b) Result of dilation of the input image* *(c) CC labeling result of dilated image* *(d) Application of CC labeling over the original image*

The assumption made in this study is, the title's used in real life documents especially on newspapers, books and magazines are bolder than other parts of the text and their positions are assumed to be the top of the text page. The thresholding that is used for separating the heading section is computed by taking the minimum area of the title as 1000 and the remaining as a noise. However, the proposed algorithm has a limitation of identifying titles in the middle of the text, and titles that have the same font size with the text block.

4.2.2.4 The Proposed Automatic Page Segmentation Technique

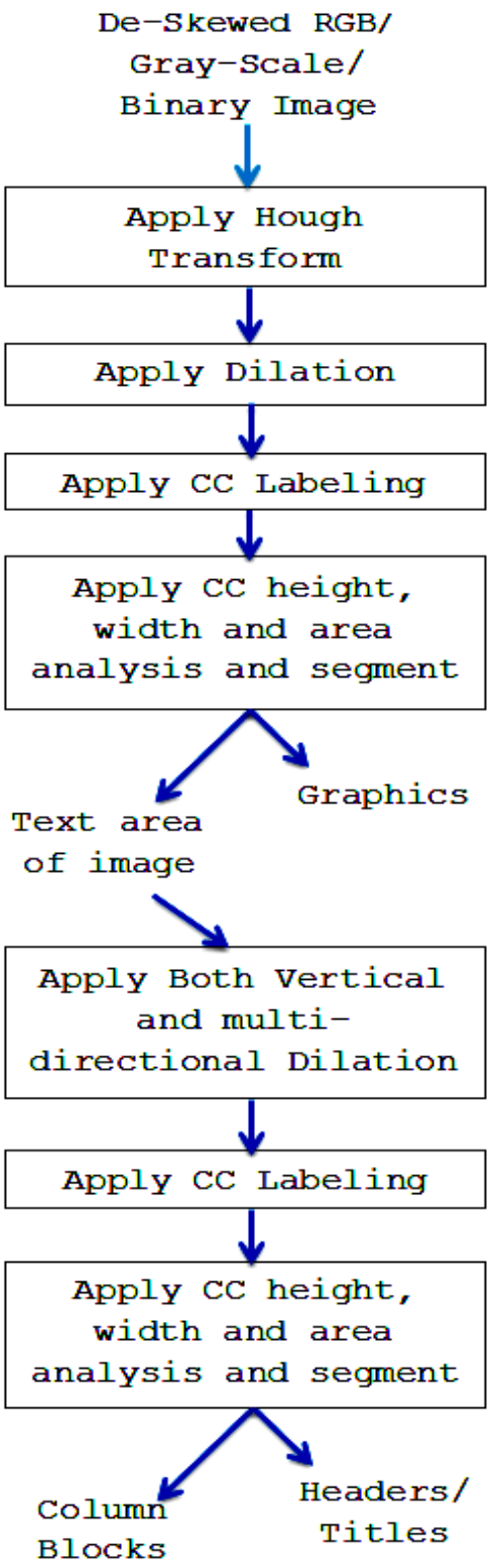
After all the above implementations and test results, the researcher proposed an automatic page segmentation method that integrates some MATLAB and Visual C# functions which are based on Hough Transform, Multi-directional and Vertical dilation, CC Labeling, and CC height, width and area analysis algorithms. The procedure of the proposed techniques is presented in figure 4.13.

As it have been discussed in section 4.2.1, the first processing that must be performed over the document images before any other stages of the OCR system is skew detection and correction due to the dependency of the next stages on the correct skewness of the document image. The next stage is named as automatic page segmentation that performs a separation of text area from the graphical regions, column blocks, tables and other objects. It removes all noises found on the border and provides an advantage for other stages to be applied only over the text image.

The input for the proposed automatic segmentation technique is the de-skewed document images that can be in a gray scale/RGB/Binary format. The proposed combined page segmentation technique first applies Hough Transform to label each table and other possible lines over the image.

The next stage applies dilation over the image to connect the words and gaps that exists between graphics and CC is applied over the dilated image. The analysis of height, width and area analysis is done to set a threshold for separating texts from graphics.

After the text/graphics segmentation is performed, the algorithm searches for column blocks and headers by applying a vertical dilation and connecting pixels vertically without losing the space between columns. CC labeling is performed next over the dilated image to label the connected components. Calculating the threshold using the height, width and area analysis is the last stage that automatically segments columns block and page headings or titles.



Input

Hough transform is applied over the image to segment tables and other lines.

Apply dilation to connect characters and fill gaps in graphics/pictures.

To identify all connected components

To set better threshold to identify the text and graphics sizes; and segment the input image.

Text/Graphics segmentation will end here and output is text area and graphics. For text areas that have columns, they will be used as input.

To connect the pixels vertically without losing the space between columns.

To label all the connected components over the dilated image.

To set threshold automatically for columns block and page headings or titles segmentation.

Output

Figure 4.13 – The proposed Automatic Page Segmentation Technique

4.2.2.5 Performance Result

The researcher tested the proposed page segmentation techniques over the datasets that are collected from real life documents presented in section 4.1. For making the decision of whether the proposed techniques perform better on the document image collections and also to select better combination of algorithms to form best performing automatic page segmentation, experiments are conducted on Amharic document images that contain tables, lines, pictures, columns and headers.

Experimental Results:

For measuring the performance of page segmentation techniques used this study uses the counting of the expected correct segmentation, the erred segmentation made and calculating the segmentation Accuracy percentage. The expected correct segmentation represents the expected number of lines, pictures, column blocks and title characters for each page segmentation methods (i.e. table, text/graphics, column block and title) individually.

Document Type		Experiment Result of Page Segmentation			
		Expected Result	**Correctly Segmented**	**Erroneously Segmented**	**Accuracy (%)**
(i)	Documents containing Tables, Lines, etc.	84	76	8	90.47 %
(ii)	Documents containing Images/Graphics	13	12	1	92.31 %
(iii)	Documents containing Column Blocks	30	29	1	96.67 %
(iv)	Documents containing Titles	7	5	2	71.43 %

Table 4.2: Experimentation result that shows the performance of the proposed page segmentation techniques

The result indicates that the proposed page segmentation technique that is based on the Hough transform, dilation, CC labeling and CC height, width and area analysis works better on real life document images with different noises. Nevertheless, the proposed technique for detecting title is tested over the document images that contain titles in the middle of text that are common on document images from newspapers. These titles cannot be detected and they can also be the

reason for poor segmentation of column blocks if they cover the white space that separate column blocks.

4.2.3 Noise Removal

In real life document images, noises can occur either in the foreground or background of document image. Therefore, noise removal is necessary to clean noisy dataset collections of Amharic document images specially the ones which are used in OCR studies. This is because noises can be the reason for disconnected lines, connected characters, large gaps between the lines, loss of information, etc. Therefore, removing errors and enhancing the document image is crucial to improve the performance of recognition result.

After skew detection and correction, and Automatic Page Segmentation are done; the forthcoming techniques are applied only the text area of document image. As it has been discussed on section 4.2.2.3, one of the advantage of page segmentation technique i.e. column block segmentation is; noises that exist outside the text area are removed successfully. Figure 4.14 shows an example of document image that contain border noises and the result of column segmentation.

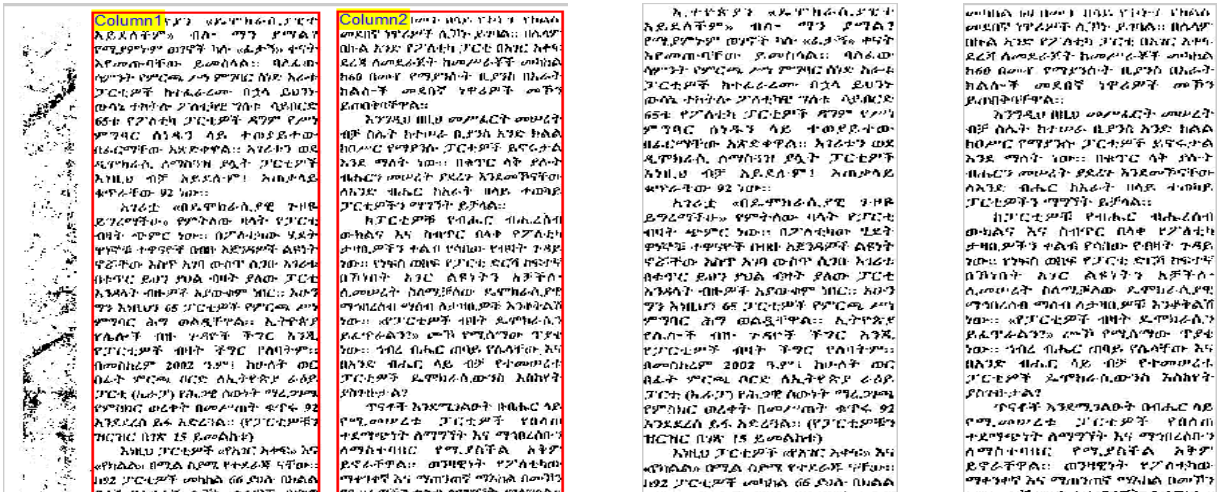


Figure 4.13 – The removal of border noises after column block segmentation

In this work, three noise filtering algorithms (Average/Mean, Median and Wiener) are tested to clean the text area of document image. Mean (Average) filtering is a method of smoothing images by reducing the amount of intensity variation between neighboring pixels. It is the most popular

and simple low pass filter that improves noisy images. Median filter is the simplest and efficient as compared to other non-linear filters. It is a commonly used special type of low-pass filter which is able to remove noise and replace the bad pixels with reasonable values by substituting the image pixel values with the median of gray values in the local neighborhood of that pixel. Weiner filtering is a powerful linear filtering algorithm to remove salt-and-pepper and also other types of noises form the document images. The technique is also the most important technique for removal of blur type of noises in images. Image denoising algorithms are implemented using MATLAB R2013a Image processing toolbox and Microsoft Visual C#.

To experiment the average filtering, a MATLAB function is developed to calculate the mean values of each pixels and compute and set the average pixel values of neighbors based on the window size *mxn*. The built-in functions of MATLAB `medfilt2(image, [m n])` and `wiener2(image, [m n])` are used for the application of median and wiener filtering in image using window size of *mxn*. The functions will accept its two parameters from Visual C# function; the first is the two-dimensional (2D) input image. Three dimensional input images must be converted into gray scale image using `rgb2gray()` function of MATLAB and the second parameter is the window size *(mxn)* in the form of array i.e. [*m n*].

As it was discussed in section 3.4, the experimented three denoising methods are integrated with Visual C# interface and MSE and PSNR are used to measure their performance by computing the amount of disturbance that a filtering algorithm introduces into the image.

As it has been discussed, one of the arguments needed to perform those denoising algorithms is window size. All the filtering algorithms are tested through different window sizes to compare their performance. From the experiments made, it is observed that window size increment have a direct impact on the quality of the image. Table 4.3 presents the effect of three denoising techniques by different window size on the same document image. As we can see from the result, the distortion over the images increases as window size increases.

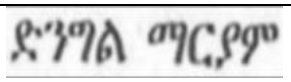
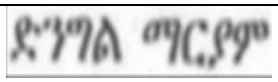
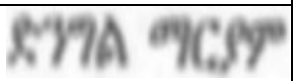
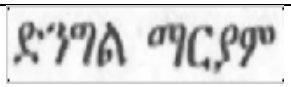
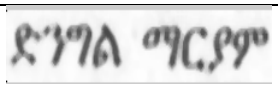
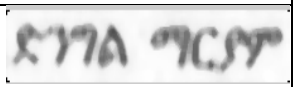
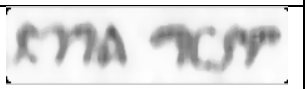
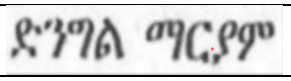
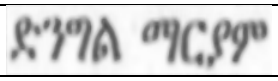
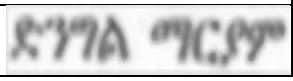
ድንግል ማርያም *(Original Image)*	Window Size (mxn)			
	3x3	5x5	7x7	9x9
Average Filtering				
Median Filtering				
Wiener Filtering				

Table 4.3: The Experimentation result of Noise Filtering Algorithms using 3x3, 5x5, 7x7 and 9x9

The noise filtering algorithms are applied on real life document images that contain various types and levels of noises. Image quality measures used in this study; MSE and PSNR are computed to pick best performing algorithm. *MSE* is a measure of an average of the squares of the errors and *PSNR* is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Both are computed using a MATLAB function `[psnr, mse] = measerr (originalImage, filteredImage)` that returns both PSNR and MSE. The following Table 4.4, Table 4.5 and Table 4.6 presents the MSE and PSNR measurement result for mean, median and wiener filtering techniques respectively.

Noise Level	Document Type	No of Pages	Mean /Average/ Filtering							
			3 x 3		5 x 5		7 x 7		9 x 9	
			MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Low	Magazine	2	248.40	24.17	694.34	19.72	1192.90	17.36	1556.60	16.21
	Qidassie Mariam	4	115.64	27.50	240.83	24.31	377.31	22.36	528.15	20.90
Medium	Book	3	116.26	27.48	256.97	24.03	386.42	22.26	463.51	21.47
	Qidassie Yohannes	2	39.82	32.13	99.11	28.17	168.04	25.88	243.91	24.26
High	Book	5	150.37	26.36	331.72	22.92	508.21	21.07	621.58	20.20
	Newspaper	1	712.41	19.60	1406.10	16.65	1989.30	15.14	2247.30	14.61
Very High	Book	6	69.33	29.72	169.39	25.84	281.81	23.63	407.70	22.03
	Qidassie Yohannes	2	36.77	32.48	95.36	28.34	163.46	25.99	239.37	24.34

Table 4.4: The Experimental Result of Image Quality Measure (MSE and PSNR) of Average/Mean Filtering Algorithm Using Different Window Sizes

Noise Level	Document Type	No of Pages	Median Filtering							
			3 x 3		5 x 5		7 x 7		9 x 9	
			MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Low	Magazine	2	136.89	26.77	461.02	21.53	1021.12	16.62	1600.50	16.05
	Qidassie Mariam	4	96.03	63.55	139.40	26.82	251.82	24.17	421.82	21.89
Medium	Book	3	66.97	30.27	177.88	26.26	352.00	23.53	447.05	22.28
	Qidassie Yohannes	2	15.02	36.39	42.04	31.90	84.36	28.86	149.81	26.38
High	Book	5	71.59	29.79	164.14	25.69	365.27	22.78	500.89	21.25
	Newspaper	1	453.81	21.56	1257.00	17.13	2248.00	14.61	2622.30	1394.00
Very High	Book	6	45.69	32.43	99.11	28.65	162.20	26.34	250.51	24.36
	Qidassie Yohannes	2	15.06	36.36	41.49	31.95	81.17	29.04	141.91	26.62

Table 4.5: The Experimental Result of Image Quality Measure (MSE and PSNR) of Median Filtering Algorithm Using Different Window Sizes

Noise Level	Document Type	No of Pages	Wiener Filtering							
			3 x 3		5 x 5		7 x 7		9 x 9	
			MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Low	Magazine	2	34.68	32.78	65.23	30.14	122.66	27.39	198.72	25.29
	Qidassie Mariam	4	16.03	37.48	28.93	35.26	52.70	33.11	89.04	31.06
Medium	Book	3	33.52	33.04	53.86	30.91	71.58	29.71	92.64	28.63
	Qidassie Yohannes	2	8.69	38.75	16.28	36.02	22.22	34.66	28.03	33.66
High	Book	5	33.05	33.12	58.70	30.55	82.30	29.08	108.08	27.89
	Newspaper	1	147.99	26.43	297.73	23.39	542.36	20.79	768.98	19.27
Very High	Book	6	32.08	33.83	63.71	30.60	89.15	29.04	115.02	27.88
	Qidassie Yohannes	2	9.08	38.56	17.51	35.70	23.38	34.45	28.87	33.54

Table 4.6: The Experimental Result of Image Quality Measure (MSE and PSNR) of Wiener Filtering Algorithm Using Different Window Sizes

Based on the results of MSE and PSNR, in each of the algorithms the lower values of MSE and higher values of PSNR are registered in the smaller window size of *3 x 3*. For example, for the medium level noise document ("Qidassie Yohannes") an MSE of 39.82, 15.02 and 8.69, and also PNSR of 32.13, 36.39 and 38.75 are registered for average, median and wiener filtering algorithms respectively. The result indicates that wiener filtering method is able to smooth out noises with a few disturbances than average and median filtering methods. Therefore, all the test document images are filtered using wiener filtering algorithm using window size of 3 x 3 before it is binarized.

4.2.4 Binarization (Thresholding)

As it was discussed in section 3.2.3, binarization is a mandatory preprocessing step in an OCR system that is responsible for the conversion of grayscale document image into bi-level (black and white) representation. Binarization is important to separate the foreground characters from their background and it also removes some of the noises.

During the conversion of document image into binary, there is a need to set a threshold value that can be used for comparing it with each pixel values. Pixel intensity levels less than a threshold value are assigned to black pixel and higher values are assigned to white pixel. There is no universal rule to perform these but some techniques have been proposed by various scholars and they are categorized into two groups as global and local thresholding techniques.

Global thresholding methods compute a single threshold value for binarizing the while image and each pixel is going to be compared to this value to decide whether a pixel is a background or foreground. Local thresholding computes threshold value for each pixel individually using the information from neighborhood pixels.

In this work, *Otsu thresholding* and *Sauvola thresholding* algorithms are tested with different combinations of image denoising techniques. MATLAB R2013a is used to implement thresholding techniques.

Ostu thresholding is implemented by using a Matlab built-in function `graythresh(image)` which computes a single global threshold for the given image so that it can be applied over the image using the function `im2bw()`. On the other hand, sauvola thresholding is implemented by

adopting a Matlab function that was directly used in Michael's [43] study and it is tested in different window size (*mxn).* (See Annex III).

Figure 4.14 displays the effect of global and local thresholding over document image from real life which has been filtered using the selected denoising technique; i.e. wiener filtering technique by window size of *3 x 3*.

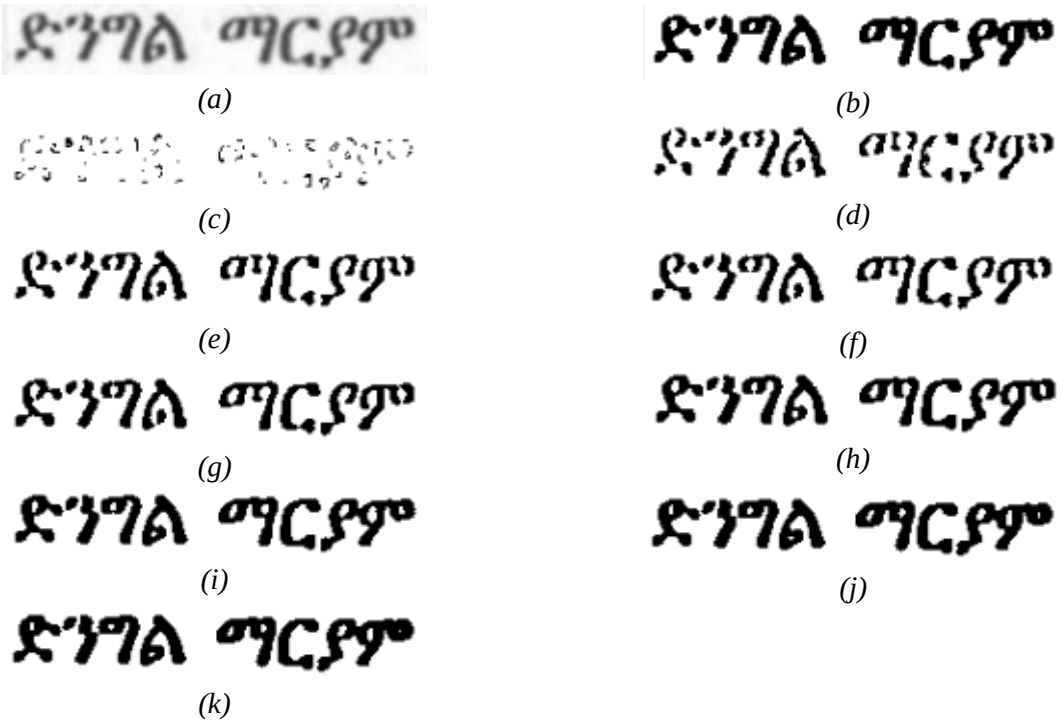


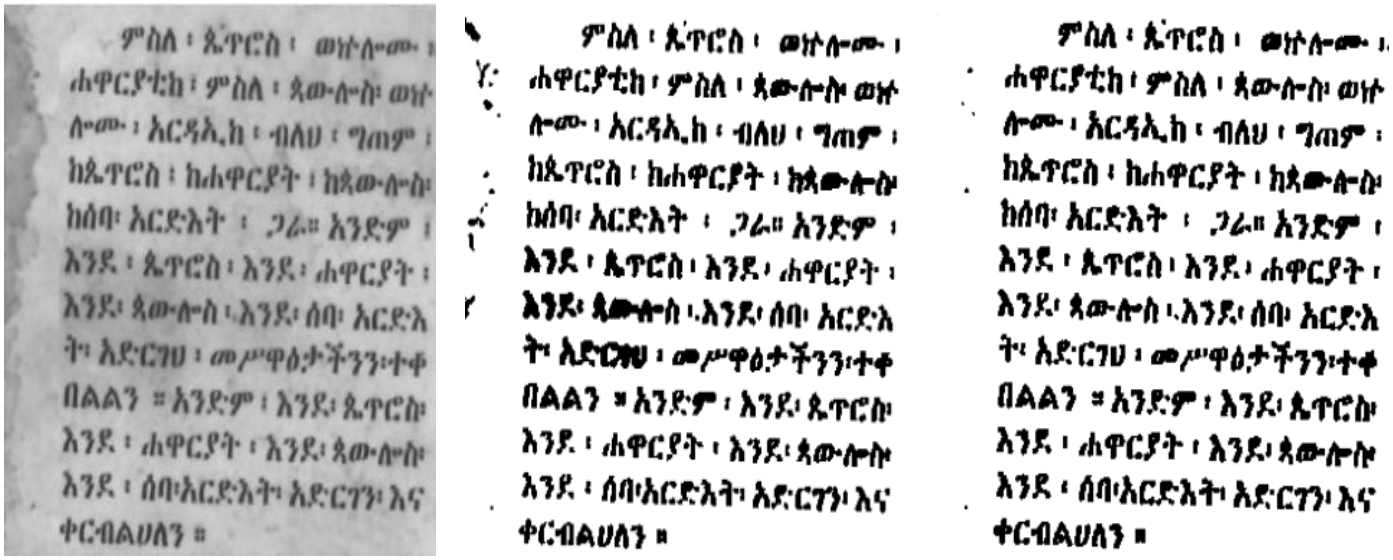
Figure 4.14 – The Experimental Result of Otsu and Sauvola Thresholding Techniques on Wiener Filtered Images

(a) Wiener Filtered Image (b) otsu thresholding (c) sauvola using window size of 3x3 (d) sauvola using window size of 5x5 (e) sauvola using window size of 7x7 (f) sauvola using window size of 9x9 (g) sauvola using window size of 11x11 (h) sauvola using window size of 15x15 (i) sauvola using window size of 20x20 (j) sauvola using window size of 25x25 (k) sauvola using window size of 30x30

From the above experiment, we can see the effect of binarization over the de-noised document image. We can observe that some of the blurring noises created by the noise filtering stage are also removed by binarization process. For measuring the quality of binarized document and select the better performing technique, a human /subjective/ evaluation is made. It is observed

that global Otsu thresholding erases some parts of the textual information and creates some unwanted foreground noises.

However, sauvola local thresholding technique uses different thresholds for each pixel by computing threshold value for each local pixel based on the neighboring pixels within a window size *mxn*. Based on the experiment, lower window size results a loss of pixel values and gives poor result whereas larger window size scores better result. It is discovered that window size larger than *11x11* works better and for this research, window size of *20x20* and *25x25* are found as best performing window sizes. Finally, 20X20 is selected as better aiming to minimize the connection of characters during binarization. Figure 4.15 explains why the Sauvola thresholding is selected over Otsu thresholding method.



(a) *(b)* *(c)*

Figure 4.15 – The Experimental Result of Binarization of degraded Document image using Otsu and Sauvola Thresholding Method

(a) Weiner filtered degraded document image

(b) Otsu thresholding method

(c) Sauvola thresholding method using 20x20 window size

As we can see from the above result Otsu global thresholding algorithm results in more foreground noises than the Sauvola thresholding that have huge impact on a recognition result. Therefore Sauvola is voted as best performing algorithm.

However, sauvola thresholding didn't clean the image at all i.e. there are also some noisy pixels. Another MATLAB function `bwareaopen(image, pixel_size)` is used to remove pixels that are less smaller than a given area i.e. `pixel_size` (see Annex III).

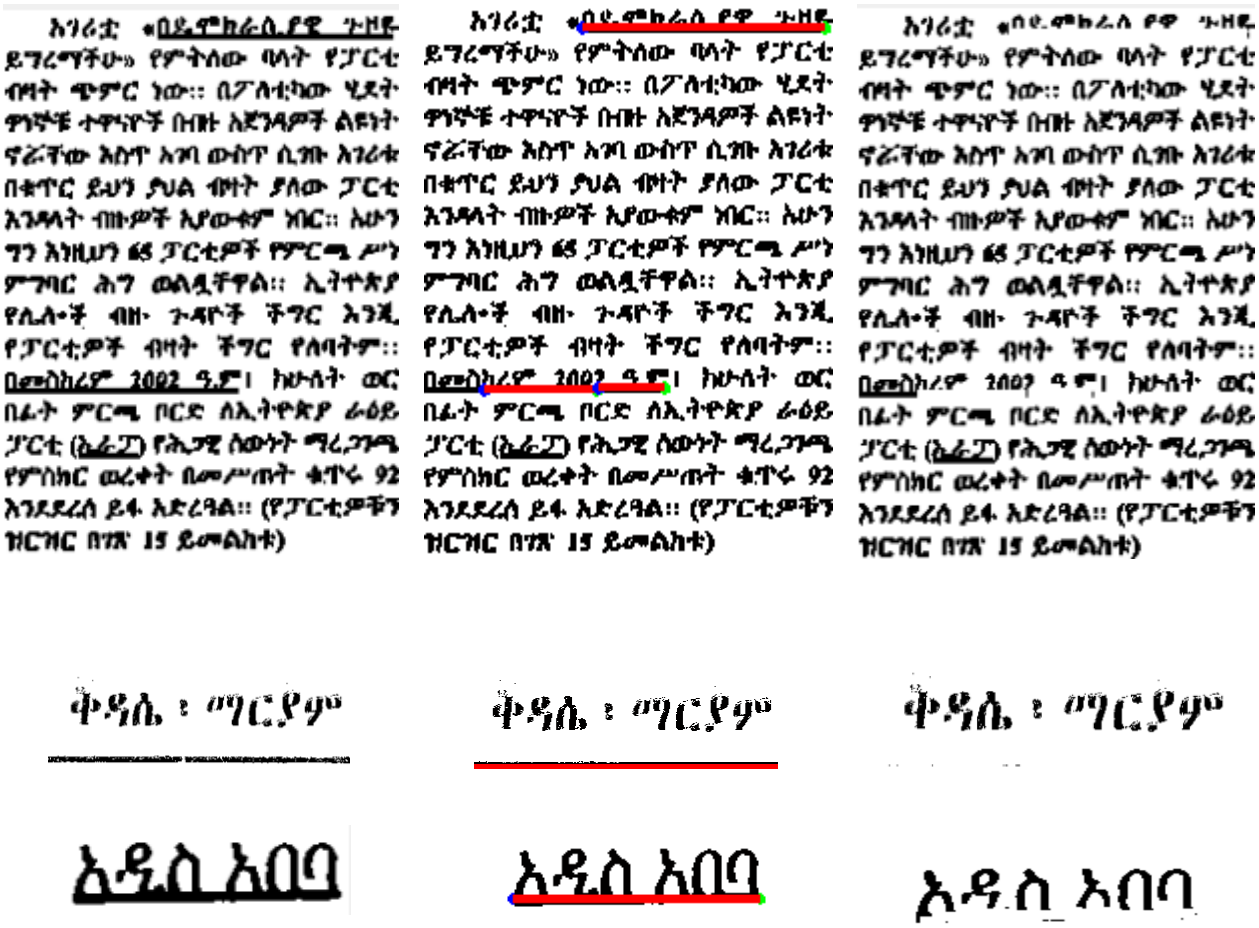
4.2.5 Underline Removal

Document images contain different types of lines such as overlines, underlines, table lines, picture lines, vertical lines, page boarders, etc. Underlines are used in document images for different reasons such as the identification of main ideas, topics or titles. They are drawn horizontally below the specific text and may or may not touch the lower parts of the text. They can also be found disconnected (as dotted lines) and slightly curved.

The page segmentation stage that deletes tables from the document image also removes underlines and many possible vertical and horizontal lines. However, there might be some missing underlines that may not have been detected in the table lines segmentation technique. Therefore, this stage is used as a purifying stage to minimize the recognition errors caused due to the undetected underlines.

The method applied here is the same as the technique used for automatic table segmentation i.e. *Hough Transform* that uses the two functions; `hough()` and `houghlines()`. Most of previous studies used an algorithm that was originally suggested by Pal and Chanduri. The algorithm can only be applied after the text line segmentation due to the fact it assumes the bottom of the text image as underline if the count of the black pixel becomes above some threshold.

The Hough transform based underline removal method can be applied either after text line segmentation or before text segmentation is performed. However, it shows some limitations to detect broken and short lines. The following figure presents the experimental result of Hough transform to detect underlines and its failures.



(a) *(b)* *(c)*

Figure 4.16 – Experimental Result of Hough Transform for Removing Underlines

(a) Underlined image *(b) Plotted image* *(c) Underline removed image*

4.3 Text Segmentation

Based on the proposed OCR system architecture that was presented in section 3.1, the next stage after image preprocessing completion is text segmentation. It is the stage that focus on the textual section of document image extracted from page segmentation and noise removal and binarization stages of preprocessings. This stage identifies text lines, words and characters from a binarized textual image and recongition accuracy depends on these phase.

Among the availiable segmentation techniques from different literatures; Projection Profile, Morphological Dilation and CC Analysis are tested. Projection profile technique calculates the sum of black pixels found in horizontal or vertical axis to locate positions where text lines, words

and characters are found in the image. Morphological dilation is used to connect characters and words by increasing the size of character image pixels. CC analysis is used to identify, label and separate the connected parts from binarized document image.

Text segmentation can be started in differnt order. However, for better reconstruction of the result and also for makinf the decomposition more easier, this study segments lines, words and characters respectively.

4.3.1 Line Segmentation

Horizontal projection profile is a line segmentation technique that is used in most of amharic OCR studies and it is expirmented in this particular study. A Visual C# method adopted from Michael [43] which accepts a parameter of binary image and threshold value and returns `List<int>` of coordinates of segmentation points. Snippet 4.7 presents the algorithm used to count black pixels in row and computes the higher and lower values of the black pixels to track the segmentation points using the threshold 7 decided as the best on his expirment.

Snippet 4.7: Implementation Horizontal Projection profile [43]

```
//Horizontal projection
for (int y = 0; y < img.Height; y++)
{
    for (int x = 0; x < img.Width; x++)
    {
      //Get the color value of current pixel and check if it is black
       pixelColor = img.GetPixel(x, y);
       if (pixelColor.R == 0 && pixelColor.G == 0 &&pixelColor.B == 0)
            pixles++;
     }
      //Add the sum in a row and restart the sum for the next row
      hrt.Enqueue(pixles);
      pixles = 0;
}
```

However, the algorithm has a problem of segmenting some characters from the training sets such as **“ኸ”** due to the disconnected feature found at the top of the character (see figure 4.17). There

are also other Amharic characters having the same features as Geez numbers. The algorithm segmented the upper appendage and the body part as individual lines.

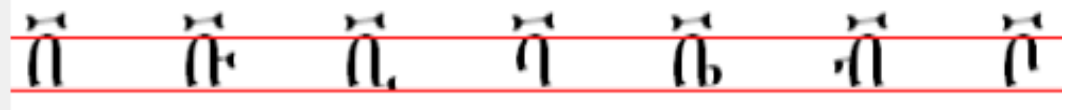


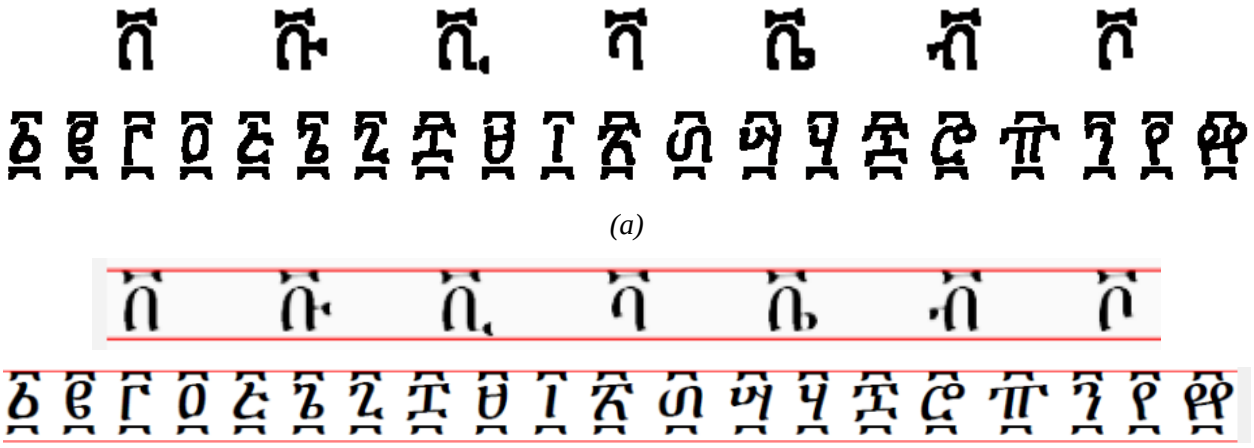
Figure 4.17 – Experimental Result of Projection Profile as Tested by Michael [43]

In order to solve such issues, this study first applies morphological dilation in vertical direction using the MATLAB code presented in Snippet 4.8. The algorithm uses *4x4* window size to dilate the image in a vertical direction.

Snippet 4.8: Implementation of Morphological Dilation (Vertical) for connecting Character Body and Appendages

```
function [verticallyDilatedImage] = vertical_dil(dilatedImage)
      %% setting strel for making vertical dilation
      se = [0 0 1 0 0; 0 0 1 0 0; 0 0 1 0 0; 0 0 1 0 0];
      verticallyDilatedImage = imdilate(dilatedImage,se);
end
```

The above method fills or close the gap between the upper or lower appendages and the body part of a given character image. Then projection profile technique will be applied over the dilated image and the coordinate points collected in the `arrayList<>` are used to segment the original input image. Figure 4.18 presents the expirmental result of vertical dilation and their effect on horizontal projection result.



(a)

(b)

Figure 4.18 – The Experimental Result that shows the effect of Vertical Dilation on the Result of Horizontal Projection Profile

(a) Vertically dilated image *(b) Horizontal projection profile*

The above expirment shows the combination of vertical dilation with horizontal projection profile shows better results in segmenting characters with appendages than the previous method. The proposed method is tested on the test sets presented in section 4.1 and the performance evaluation is presented in section 4.3.4. . Figure 4.19 shows the sample expirmental results of the proposed line segmentation method over real life dataset.

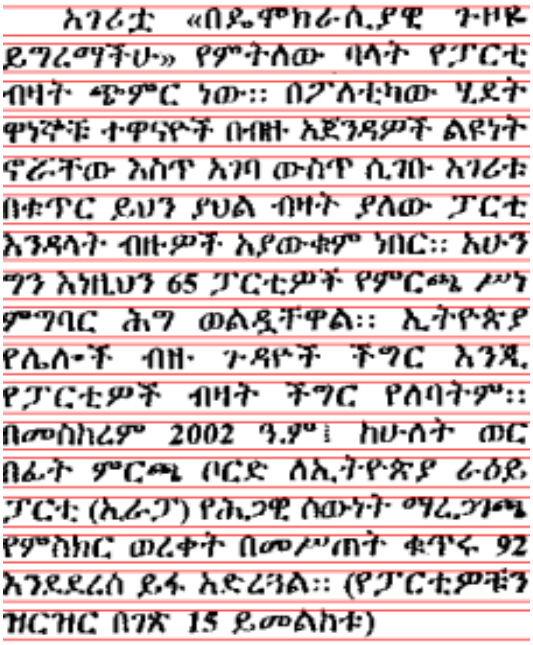


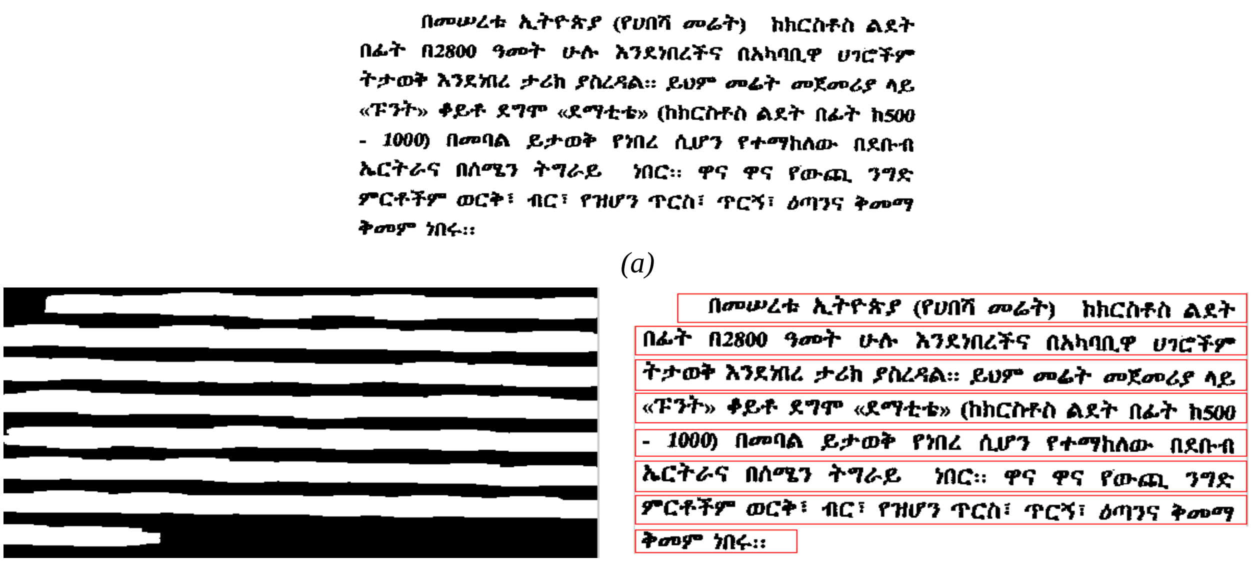
Figure 4.19 – Experimental Result of the Proposed Technique On Real Life Document Image

Another method that is expirmented to segment text line is a morphological dilation technique which connects binary image pixles in a horizontal direction. To decide the better horizontal dilation window size, an iterative test is made and it is observed that using the window size of *29x9* shows better results. Figure 4.20 displays the developed window for image dialation.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.20 – The Proposed Window for Dilation of Image Horizontally to Connect Character in a Row

Using the suggested window, the input image is dilated by the MATLAB code that performs horizontal dilation. Finally, white space analysis is made using CC Analysis to segment the text line. Figure 4.21 presents the effect of horizontal dilation over the image and the result of text line segmentation.



(a)

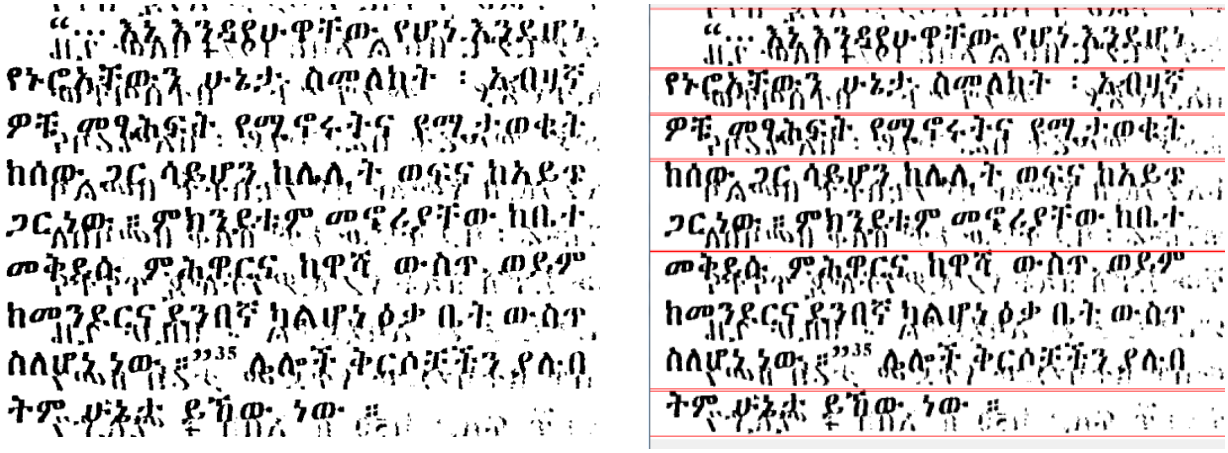
(b) *(c)*

Figure 4.21 – Experimental Result of Horizontal Dilation and CC Analysis for line segmentation

(a) Binarized Image *(b) Horizontal Dilation* *(c) CC Analysis*

Based on the expirmental results, both of the proposed algortihms performs better to segment text lines from the binarized document images. Nevertheless, dilation based method has a limitation of segmenting training set ‘Fidel’ because of the large space between characters that cannot be connected. Consequently, for both training and testing sets text line segmentation, horizontal projection profile with a vertical dilation is prefered.

However, both of the proposed methods failed to segment document images having an ink-bleeding noise at their background. Figure 4.22 shows an example of ink-bleeded document image and the result of the proposed projection profile based method that causes an incorrect extraction of text lines which has a direct impact on the performance of recognition.



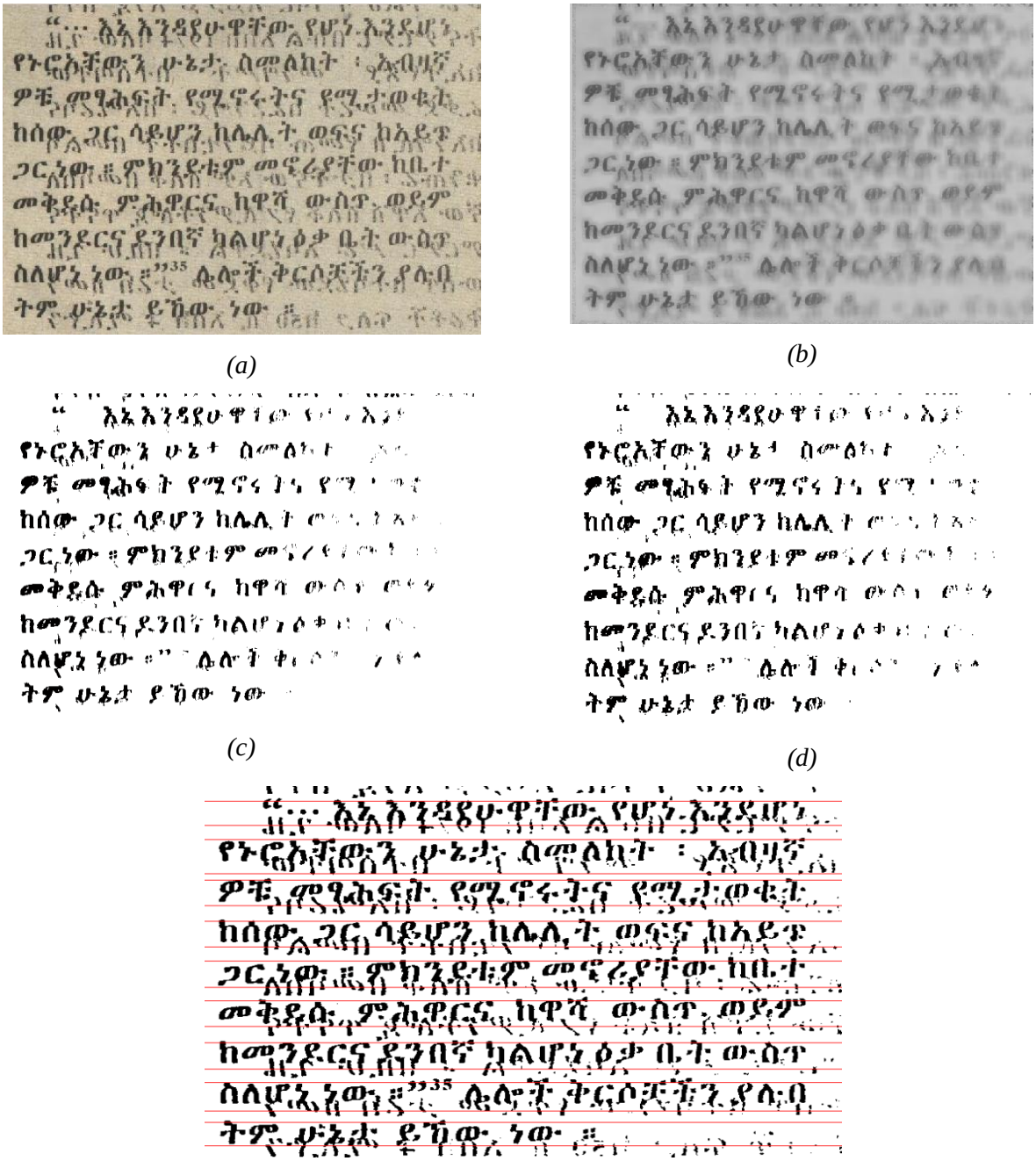
(a) *(b)*

Figure 4.22 – Experimental Result of Horizontal Projection Profile on Ink-Bleeded Document Image

(a) Binarized ink-bleeded document image *(b) Horrizontal Projection Profile Result*

To alleviate this problems, a new method is proposed that first smooths the original image using ink-bleeding filtering function developed using wiener filtering and binarizes the document by sauvola thresholding.

The filtering performed using wiener filtering smooths or blurs the image using expirmentally decided *10x10* window size and sauvola local thresholding using *25x25* window size is applied to make the possible text lines visible. After thresholding, the proposed horizontal projection profile method is applied over the processed image. Lastly, the traced segmentation points are used to segment the original image. Figure 4.23 displays the experimental result of Ink-Bleeded document text line segmentation.



(a) *(b)* *(c)* *(d)* *(e)*

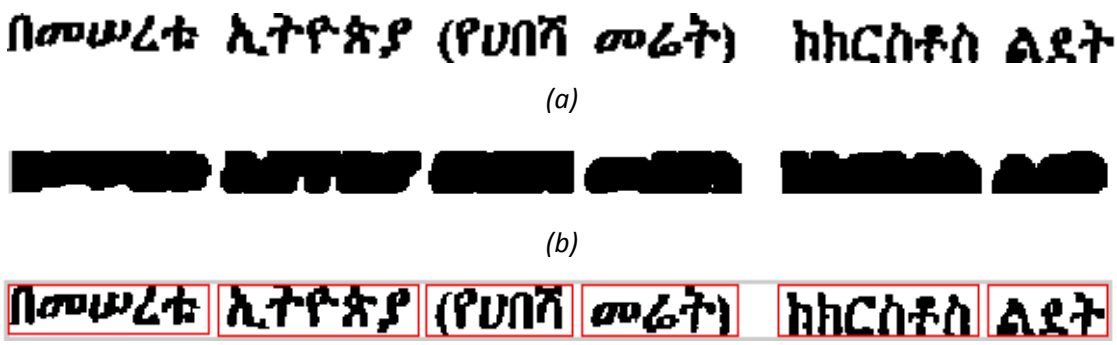
Figure 4.23 – Experimental Result of the proposed Segmentation Technique for an Ink-Bleeded Document Images

(a) Original ink-bleeded image (b) weiner filtering using 10x10 window size (c)Sauvola Thresholding using 25x25 window size (d) Dilation result (e) Horizontal Projection Profile Result

4.3.2 Word Segmentation

Word segmentation is performed over the extracted text lines to segment word images. For this purpose, this study first expirmented a technique based on dilation and CC analysis. It is implemented using a MATLAB function to dilate and increases the pixel of the characters in all direction and 5 is selected as best threshold value to dilate the image after an iterative experiment is made (see Annex III).

After the morphological dilation, CC Analysis is made to find the connected regions from the dilated image and the identified components are taken as a word. Figure 4.24 shows the expirmental results of dilation and CC analysis in segmentation of words.



(a)

(b)

(c)

Figure 4.24 –Experimental Result of Dilation Based Word Segmentation

(a) Binary Text Line Image (b) Dilated Image using threshold value of 5 (c) CC labled Image

The above segmentation method is adopted from Gedion's [26] study on DIR and it is also tested on historic documents. Most of historic document images separate words using a colon like punctuation mark known as "Two-Dots"/**:**/ that becomes a reason for poor word segmentation results. However, most recent documents separate words using white space and the proposed technique works very well on those types of documents. Figure 4.25 shows exprimental result over historic documents.

Figure 4.25 – Experimental Result Shows Errors of Dilation Based Word Segmentation on Two-Dot Separated Sentence

Another method that is mostly used by previous studies is a vertical projection profile method. In Michael's [43] study, it was observed that the algorithm was affected by non uniformity between the spaces found between words.

For the expirmentation of vertical projection profile, a Visual C# program that is adopted from Michael [43] and presented in Snippet 4.9 is used. The algorithm is also applied for the segmentation of characters by counting the black pixels vertically.

Snippet 4.9: Implementation of Vertical Projection Profile [43]

```
//Verical projection
for (int x = 0; x < img.Width; x++)
{
   for (int y = 0; y < img.Height; y++)
   {//Count black pixels vertically
      pixelColor = img.GetPixel(x, y);
      if (pixelColor.R == 255 && pixelColor.G == 255 &&...
                                    pixelColor.B == 255)
            pixles++;
      }
      vrt.Add(pixles);
      pixles = 0;
}
```

Figure 4.26 shows the expirment result of vertical projection profile on text line images that was segmented in previous stage.

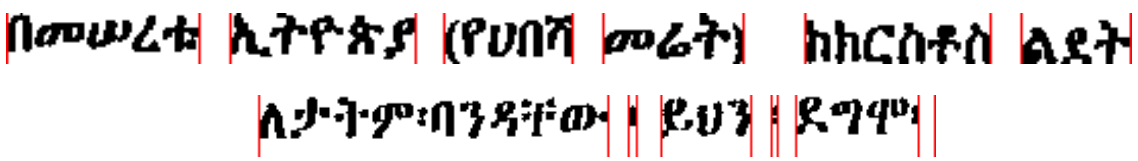


Figure 4.26 – Experimental Results of Vertical Projection Profile in Word Segmentation for both White Spaced and Two Doted Images Respectively

The proposed segmentation techniques are also tested over an ink-bleeded segmented line image. Segmenting word from ink-bleeded document is challenging and as we can see from the

expirmental result (see figure 4.27), the expirmented techniques failed to segment words from an ink-bleeded text lines.

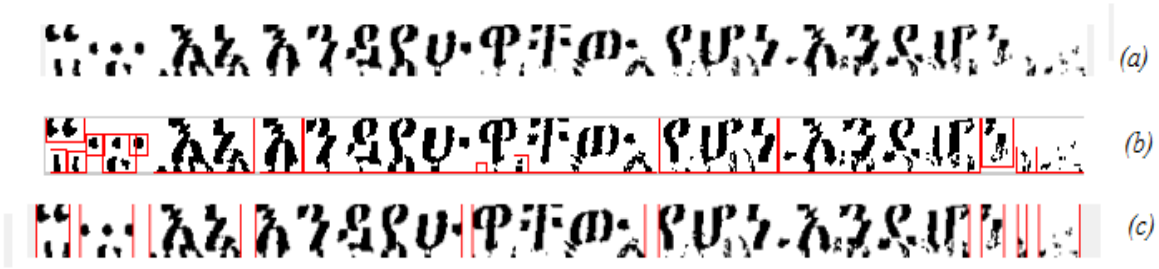


Figure 4.26 – Experimental Result Shows the Result of Dilation and Vertical Projection Profile Based Word Segmentation for ink-bleeded documents

(a) Original ink-bleeded line image (b) Morphological Dilation (c)Vertical Projection Profile

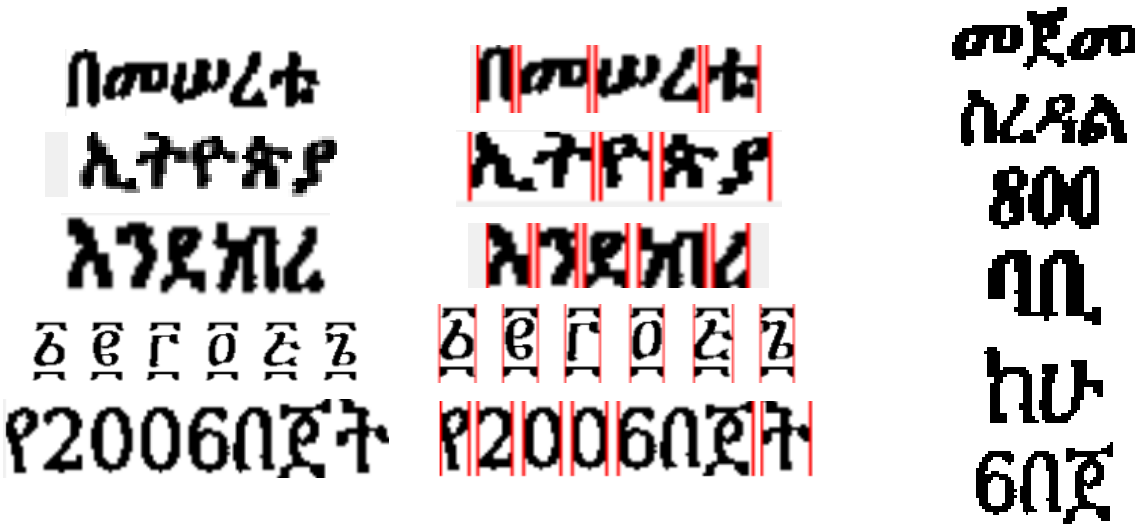
Based on the above expirment, both methods are tested over an ink-bleeded text line images and poor performance is registerd. Also for words separated by two dots, this stage can be skipped because words can be recognized using those separation points.

4.3.3 Character Segmentation

Character segmentation is the final text segmentation stage performed after word segmentation is completed. It is a process of identifying and segmenting individual characters from the given binary word or text line images. In this particular study, two character segmentation techniques are tested; *Vertical Projection Profile and CC Labeling or Analysis.*

To experiment vertical projection profile, the same algorithm applied in word segmentation is used. The thresholding formula applied in Michael's [43] study is adopted to find the space between characters automatically. It is done by an approximate pixel width calculation and the height of the image to estimate the average number of characters in an image that is used as a threshold value to find the spacing between characters.

Using this approach, it was observed that those set of rules failed to segment characters which are close to each other, overlapping, and connected. Figure 4.27 presents the experimental result of vertical projection profile to segment characters from the given word images.

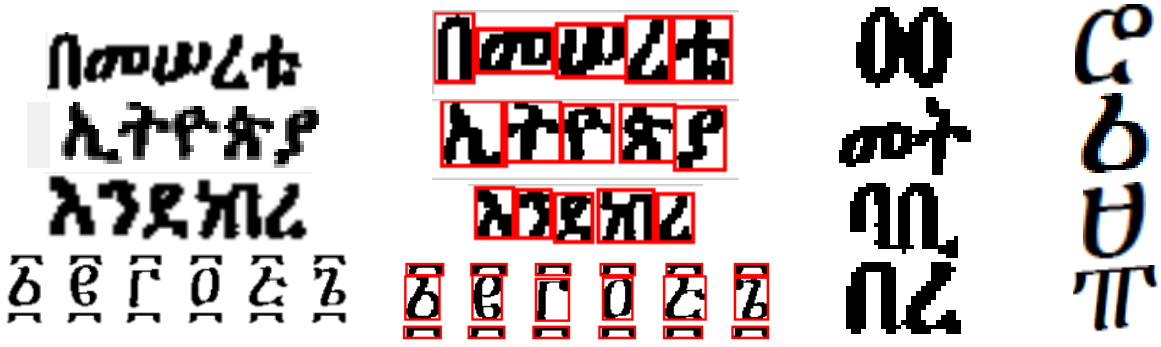


(a) *(b)* *(c)*

Figure 4.27 – The Experimental Result of Vertical Projection Profile Based Character Segmentation

(a) Binary Word Image *(b) Vertical Projection Profile* *(c) Erroneously Segmentned Characters*

The result from the experiment shows vertical projection profile failed to segment characters correctly even if they are not connected. This study tests another method i.e. CC Analysis which solved some of the above problems. The algorithm computes the connected components from a character image and crops the image using MATLAB built in function called `imcrop()`. However, this method has a problem of segmenting characters having disconnections in their body. Figure 4.28 shows the result of CC Analysis in character segmentation.



(a) *(b)* *(c)*

Figure 4.28 – Experimental Result Shows the Result of CC Analysis Based Character Segmentation

(a) Binary Word Image *(b) CC Analysis* *(c) Erroneously Segmentned Characters from their appendages*

In the above sample experimental results, it is observed that CC analysis is poor in segmenting characters with disconnected nature or disconnections that are caused by noises. Both of the tested methods have their own advantages and disadvantages on segmenting character. CC Analysis can successfully segment overlapping characters that are very close if they are not connected. However, characters with separated upper or lower appendages from character body and characters having disconnection in their body are erroneously segmented whereas vertical projection profile can segment those characters successfully. Therefore, there is a need to develop a robust technique that can merge the advantage of both algorithms.

In this particular study, CC analysis is modified and tested for character segmentation using MATLAB image processing toolbox. First, the algorithm performs CC Analysis over the word image, it search and labels connected pixels from a given binary image. Figure 4.29 shows an example of CC Labeling result.

Figure 4.29 – Sample CC Labeling Result

CC Labeling labels all connected regions including connected characters and characters with appendages. Then, the algorithm calculates and save automatically some data about each CC labels or bounding boxes in MATLAB array called `size_info`. The collected data has the following attributes presented in table 4.7.

CC Num	X(START)	X(END)	Y(START)	Y(END)	Width(W)	Height(H)	Area	Mid-Point of X	$\frac{W}{H}$	$\frac{H}{W}$
1	0.5000	5.5000	7.5000	17.5000	5.0000	10.0000	50.0000	3.0000	0.5000	2.0000
2	5.5000	9.5000	7.5000	17.5000	4.0000	10.0000	40.0000	7.5000	0.4000	2.5000
3	10.5000	26.5000	1.5000	17.5000	16.0000	16.0000	256.0000	18.5000	1.0000	1.0000
4	28.5000	55.5000	0.5000	17.5000	27.0000	17.0000	459.0000	42.0000	1.5887	0.6296
5	56.5000	61.5000	7.5000	17.5000	5.0000	10.0000	50.0000	59.0000	0.5000	2.0000
6	60.5000	64.5000	8.5000	17.5000	4.0000	9.0000	36.0000	62.5000	0.4444	2.2500

Table 4.7: The Sample Collected Data about CC

Using the collected information about each detected components of a given word image, first the algorithm detects the main body of the character and checks weather the character has appendages or other disconnected natures from either top or bottom.

An important assumption is made i.e. a character main body that cross the central height of word image is made to detect the main body of the character image. Therefore, half of the word image height `(height/2)`, maximum and minimum value of height and width is also calculated using MATLAB built-in function `max( )` and `min( )` from the collected `size_info` array. Figure 4.30 illustrates some of the calculations made.

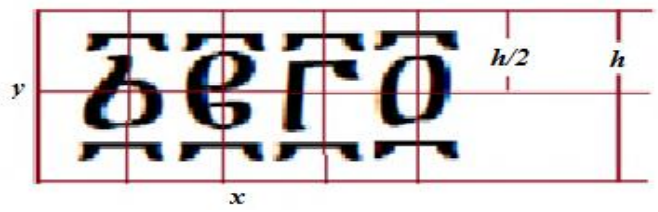


Figure 4.30 – Sample word Image Height and Mid-Point of X Calculation

After calculating the height(`h`), `h/2`, and `Mid-Point` of each component, the proposed method checks whether the given component shares the same vertical point to find the largest object in a vertical direction and it takes it as component's main body. Then, the decided main body of the character is labeled with a bounding box having a height the same as input word image. The following figure 4.31 shows the result of each step in the proposed method that is used to segment characters.

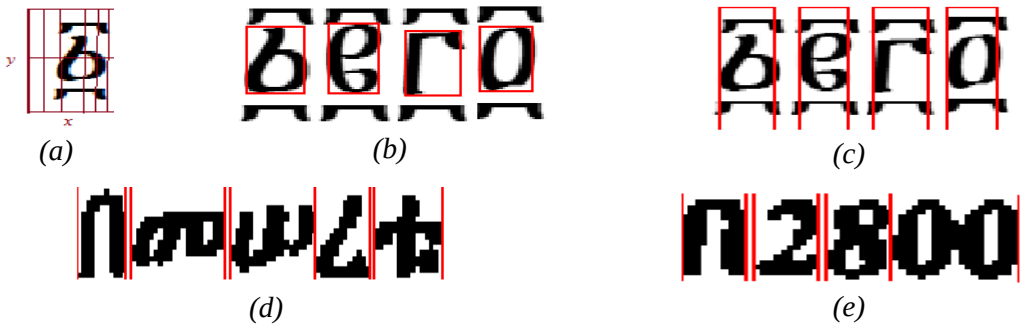


Figure 4.31 – The Experimental Results of the Developed CC Based Method

a) Finding Appendages and MainBodies (b) Detect Character Main bodies (c) Adjust bounding box window size to the height of the image (d) Result on Normal Text (e) Errors in segmenting conencted characters

The method successfully segmented characters that have appendages and also characters that have close pixel difference to another character. However, it failed to segment connected and broken characters. For this purpose, another rule is added that uses information stored in the previous array (table 4.7).

Estimation or detection of two or more connected characters in Amharic is a challenging task. This is because of the variations in character width. Characters such as '፣ ' have very small width whereas characters like '፴ and '፻' have larger width than its height.

The developed thresholding points are decided as the best points after an iterative test. The algorithm first compares weather the width of the character is greater than its height. In real life document images, it is assumed that, most connected characters have a greater width greater than its height.

The following rules and thresholding methods are developed and proposed to detect the connected characters. The developed method that leads us to connected characters is given in the following Snippet 4.10.

Snippet 4.10: The Developed Method for Identifying and Segmenting Connected Characters

```
for cnt = 1:len
%     Detecting connected characters
      if hei >=size_info(cnt,4) && hei <= size_info(cnt,5)
            if size_info(cnt,6) > size_info(cnt,7)
               if(((size_info(cnt,6)-minW)/10 >= 0.5))
                  if(maxH - size_info(cnt,7) <= 3)
                     if(size_info(cnt,10) >= 1.18 && . . .
                                 size_info(cnt,11) >= 0.5)
                       % Return the detected connected character
                     end
                  end
            end
       end
%     Return as non-connected Characters
end
```

The result of the above code only performs the detection of connected characters and the formula used was decided through an iterative experiment. It shows promising result on identifying or detecting connected characters. The following sample plotting images presented in figure 4.32 are some of the sample experimental results of the proposed method. The green line plot represents the connected characters whereas the red lines are normal characters.

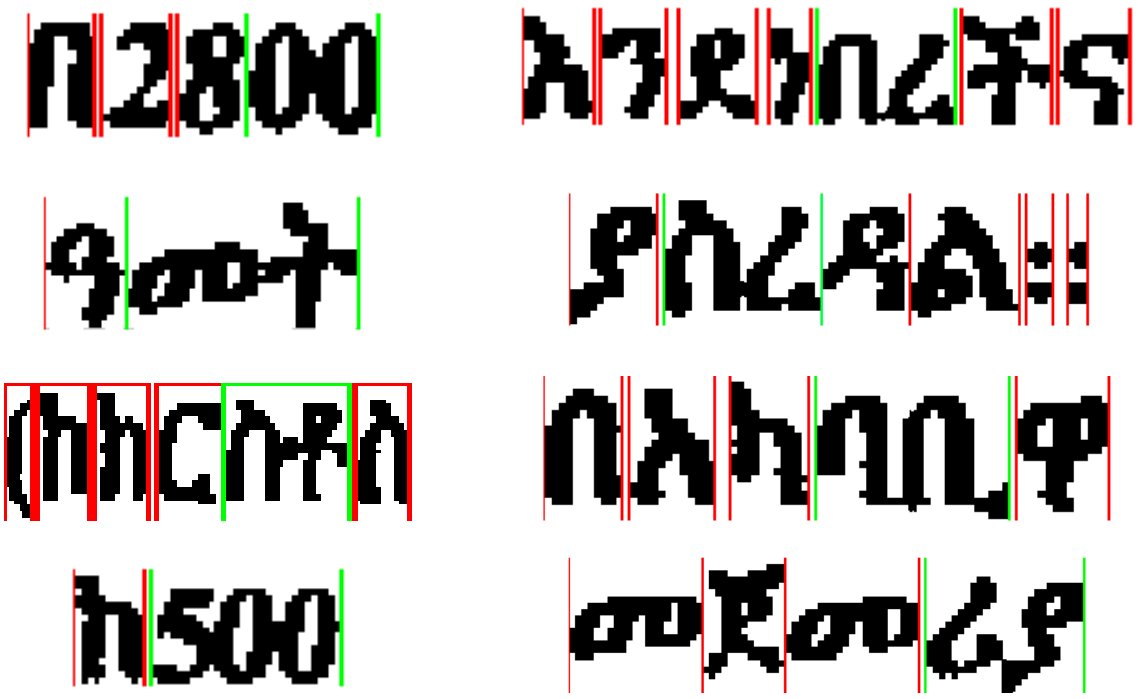


Figure 4.32 – The Experimental Results of Connected Characters Detection

The next step after identifying the connected characters is splitting them into individual characters. It is a very challenging process due to inconsistency of Amharic characters width. After some experiments, another rule is produced to split the connected characters that were detected in the previous stages.

The new rule also depend on the $\frac{W}{H}$ information stored on the size_info array which is used to checks how many times the image needs to be divided. The rule is presented in Snippet 4.11 and the full code is presented in Annex III.

Snippet 4.11: The Developed Method for Splitting Connected Characters

```
if size_info(cnt,6)/size_info(cnt,7) >= 2.5
      if size_info(cnt,6)/size_info(cnt,7) > 2.6
            if size_info(cnt,6)/size_info(cnt,7) >= 3
                  if size_info(cnt,6)/size_info(cnt,7) >= 4
                        % divide it into five equal parts
                  else
                        % divide it into four equal parts
                  end
            else
                  % divide it into three equal parts
            end
      else
            % divide it into two equal parts
      end
else
      % needs further checking of the width
end
```

The above developed Matlab code splits connected characters into a maximum of five equal parts based on the stored information i.e. $\frac{W}{H}$. Some of the sample results of splitting are presented in figure 4.33 which green plots represent splitting points.

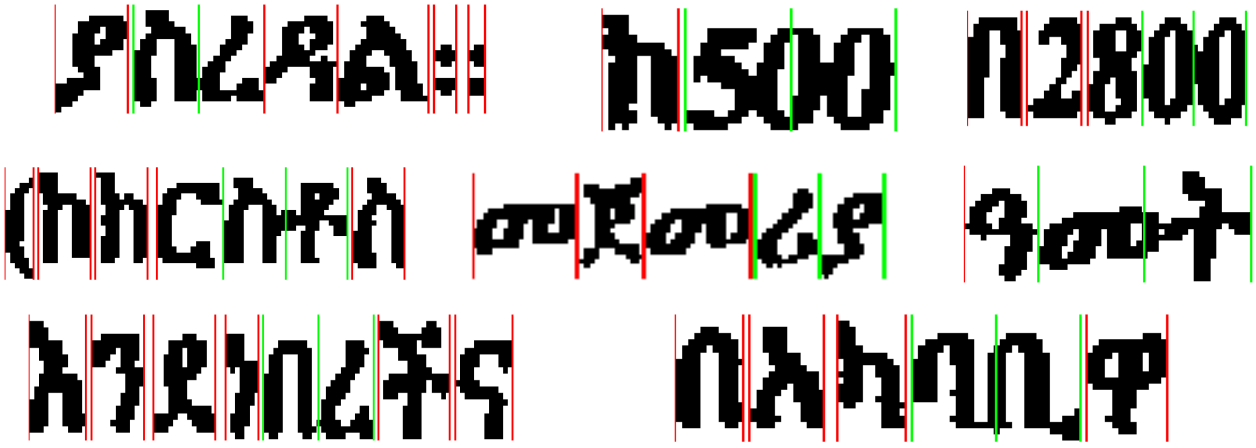


Figure 4.33 – The Experimental Results of Connected Characters Splitting

However, the proposed character detection and splitting techniques detect long characters like ‘ጬ’, ’ጯ’, ’ማ’, ’ሜ’, ’ም’, ’ም’ and ’ሟ’ as connected characters. To prevent an error made by a splitting method, some checking procedure is performed based on size analysis before splitting process. It compares the area of the regions that are selected to be divided with their original main area and using the following thresholding method presented in Snippet 4.12, it successfully eliminates some of those errors (see Annex III).

Snippet 4.12: Matlab Code for checking characters Area before splitting

```
% needs further checking (Continued from Snippet 4.11)
subImage1 = imcrop(image, . . . % the first image section
subImage2 = imcrop(image, . . . % the second image section
% Compute the CC and area for both images
[cc2, num2] = bwlabel(~subImage1);
Iprops2 = regionprops(cc2);
Ibox2 = [Iprops2.BoundingBox];
Ibox2 = reshape(Ibox2,[4 num2]);
[cc3, num3] = bwlabel(~subImage2);
Iprops3 = regionprops(cc3);
Ibox3 = [Iprops3.BoundingBox];
Ibox3 = reshape(Ibox3,[4 num3]);
y = Ibox2(:,1);
z = Ibox3(:,1);
if (((y(3,:,1) * y(4,:,1)) >= ((size_info(cnt,8)/4) + 20)) &&. . .
              ((z(3,:,1) * z(4,:,1)) >= ((size_info(cnt,8)/4)) + 20))
      % Split the image
else
      % Cancel the splitting process and return the image
end
```

The sample result of the above algorithm is presented in the following figure 4.34.

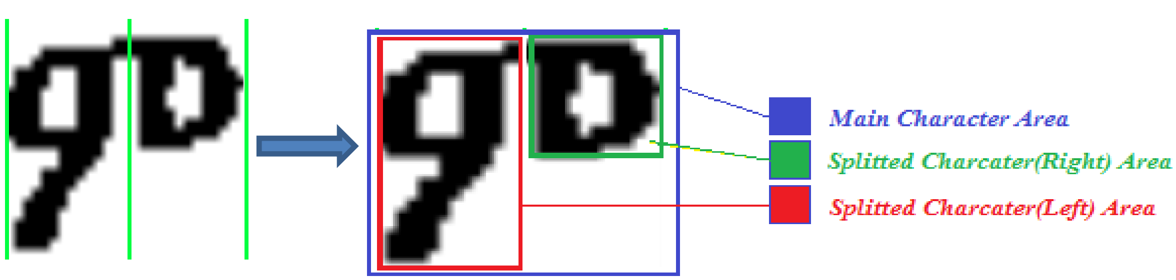


Figure 4.34 –Experimental Results of Checking before Splitting

However, the proposed character segmentation procedure registered low performance on the document images with very high noise levels by segmenting small disconnected components of broken characters as separate characters.

4.3.4 Performance Evaluation

The researcher tested the proposed text segmentation techniques on the datasets collected from real life documents. To decide which algorithm performs better, it's important to see the experimentation result obtained from the tests made on those datasets.

Experimental Results:

For measuring the performance of text segmentation, the same technique that is used for measuring the performance of automatic page segmentation method is also applied. Table 4.8 presents the experimental results for each text segmentation stages.

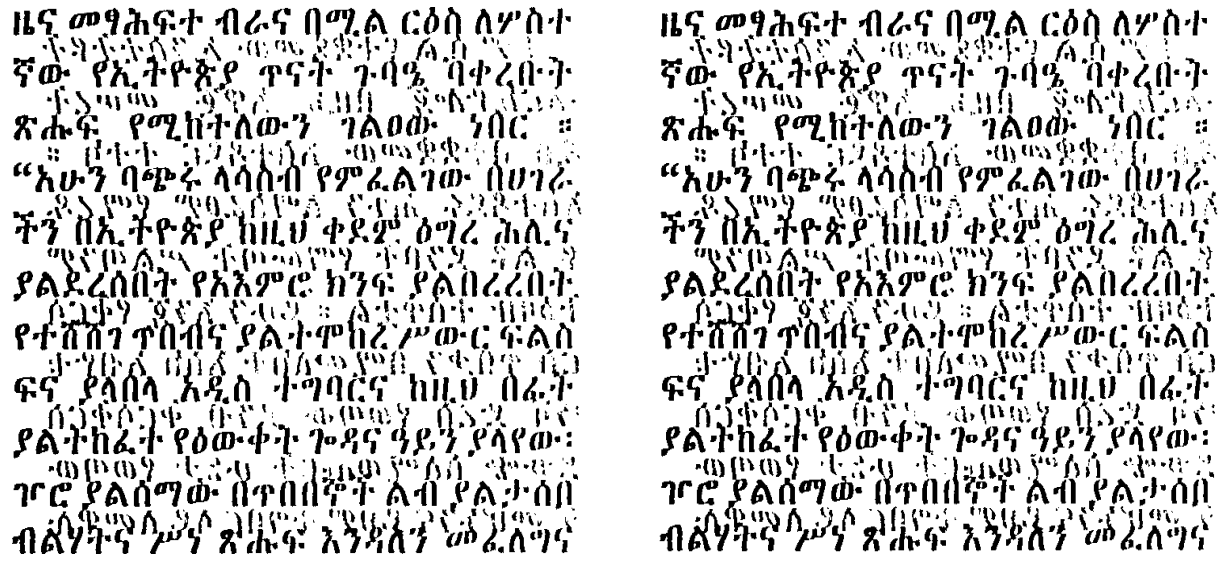
	Noise Level	Document type and no. of pages	Experimental Result of Text Line Segmentation						
			Expected Result	Correctly Segmented		Wrongly Segmented		Accuracy (%)	
				Horizontal Projection with Dilation	Horizontal Dilation with CC	Horizontal Projection with Dilation	Horizontal Dilation with CC	Horizontal Projection with Dilation	Horizontal Dilation with CC
Testing Dataset	Low	2 Magazine, 4 Qidassie Mariam	123	123	123	0	0	100 %	100 %
	Medium	3 Books, 2 Qidassie Yohannes	175	175	170	0	5	100 %	97.14 %
	High	5 Books, 1 Newspaper	222	177	126	45	96	79.73 %	56.76 %
	Very High	6 Books, 2 Qidassie Yohannes	285	215	160	70	125	75.44 %	56.14 %
		Sum	**805**	**690**	**579**	**115**	**226**	**85.71 %**	**71.93 %**
Training Dataset	PG-Unicode 3	Fidel (4 Copies)	164	163	0	1	0	99.39 %	0 %
	VG-Unicode	Fidel (4 Copies)	164	162	0	2	0	98.78 %	0 %
		Sum	**328**	**325**	**0**	**3**	**0**	**99.09 %**	**0 %**

Table 4.8: Experimental Results of Text Line Segmentation Methods

Based on the above result, both of the expirmented text line segmentation methods show better performance for testing datasets with low and medium noise levels. The horizontal projection profile with morphological dialation performs 100% accuracy for both low and medium level noisy document images. Also it segments text lines from the training dataset “Fidel” with an average accuracy of 98.78%.

The horizontal dilation and CC based line segmentation method segments 0% of the training set due to its dependecy on the space between characters. However, 100% and 97.14% accuracy is obtained for segmenting text lines of low and medium noise level images. Though, the proposed projection profile with dilation performs better by sucessfully extracting all text lines from all of the low and medium noise level document images.

For the document images with a high and very high noise level; performance of both methods shows poor accuracy rates. Horizontal dilation based method can only segment 56.45 % and horizontal projection profile segments 77.58 % accurately. This is because most of the document images at those noise level datasets contain an ink-bleeding. It is challenging to find the space between text line for those document images. The following sample result on figure 4.35 shows an erroneous text line segmented from an ink-bleeded document images.



(a) *(b)*

Figure 4.35 –Experimental Results of Text Line Segmentation on an ink-bleeded document image

(a) Image considered as text line using Horizontal Projection Profile

(b) Image considered as text line using Morphological Dilation

As it was discussed on section 4.3.1, new method based on wiener filtering, sauvola thresholding and horizontal projeciton profile is developed and expirmented on those ink-bleeded document images and the result is presented in table 4.9.

	Noise Level	Experimental Result of Text Line Segmentation (For ink-bleeding)			
		Expected Result	Correctly Segmented	Erroneously Segmented	Accuracy (%)
Ink-Bleeded Document Image	Very High	245	243	2	99.18 %
	Sum	**245**	**243**	**2**	**99.18 %**

Table 4.9: Accuracy of the Proposed Text Line Segmentation Method for Document Images with Ink-Bleeding Noise

As we can see from the expirment result, the proposed method has identified 99.18% of the lines from a high noise level document images whereas the previously used methods segment 56.45% and 77.58 % from those document images.

The next stage after text line segmentation is word and character segmentation. Table 4.10 and table 4.11 presents the results of both word and character segmentation for low, medium, high and very high level noisy datasets prepared by Biniyam [13].

	Noise Level	Document types	Experimental Result of word Segmentation						
			Expected Result	Correctly Segmented		Erroneously Segmented		Accuracy (%)	
				vertical Projection	Dilation with CC	vertical Projection	Dilation with CC	vertical Projection	Dilation with CC
Testing Dataset	Low	Magazine and Qidassie Mariam	783	777	775	6	8	99.23 %	98.97 %
	Medium	Book and Qidassie Yohannes	956	920	898	36	58	96.24 %	93.93 %
	High	Books and Newspaper	1025	893	824	132	201	87.12 %	80.39 %
	Very High	Book and Qidassie Yohannes	989	542	565	447	547	54.80 %	57.13 %
			3808	**2642**	**2542**	**1166**	**1267**	**69.38** %	**66.75** %

Table 4.10: Accuracy of the Proposed word Segmentation Method in four Noise Levels

The result in table 4.10 shows the accuracy of both Michael's [43] and Gedion's [26] word segmentation methods resulted in poor performance for document images with very high level of noise. Such result was obtained because most documents include ink-bleeding as well as the writing style of those documents uses two-dots to separate words.

Table 4.11 shows expirmental results obtained from the expirmented character segmentation methods on a sample taken from document images at each noise-level to measure the performance of character segmentation.

Experimental Result of Character Segmentation										
Document Noise Level	Expected	Correctly Segmented			Erroneously Segmented			Accuracy (%)		
		VP	CC	Modified CC	VP	CC	Modified CC	VP	CC	Modified CC
Low	1695	1413	1152	1485	282	543	210	83.36 %	67.96 %	87.61 %
Medium	1107	901	588	911	206	519	196	81.39 %	53.12 %	82.29 %
High and Very High	2036	1106	938	1031	930	1098	1005	54.32 %	46.07 %	50.64 %
Sum	4838	3857	2678	3427	1418	2160	1411	79.72 %	55.35 %	70.84 %

Table 4.11: Accuracy of the character Segmentation Methods in noisy real life document images

The results indicate that the proposed character segmentation performs better in low and medium level noisy document images and it detects most of the connected characters and split them. However, for the high and very high level document images, errenous segmentations happens mostly due to the degradations of the character.

The developed segmentation method results in character heights same as that of the word image. Therefore, it might introduce an extra white space in all directions of the character during the segmentation process. Therefore, there is a need to eliminate those spaces to minimize their effect on recognition. Figure 4.36 shows an example of extra unnecessary white spaces producing additional height and width to the character image.

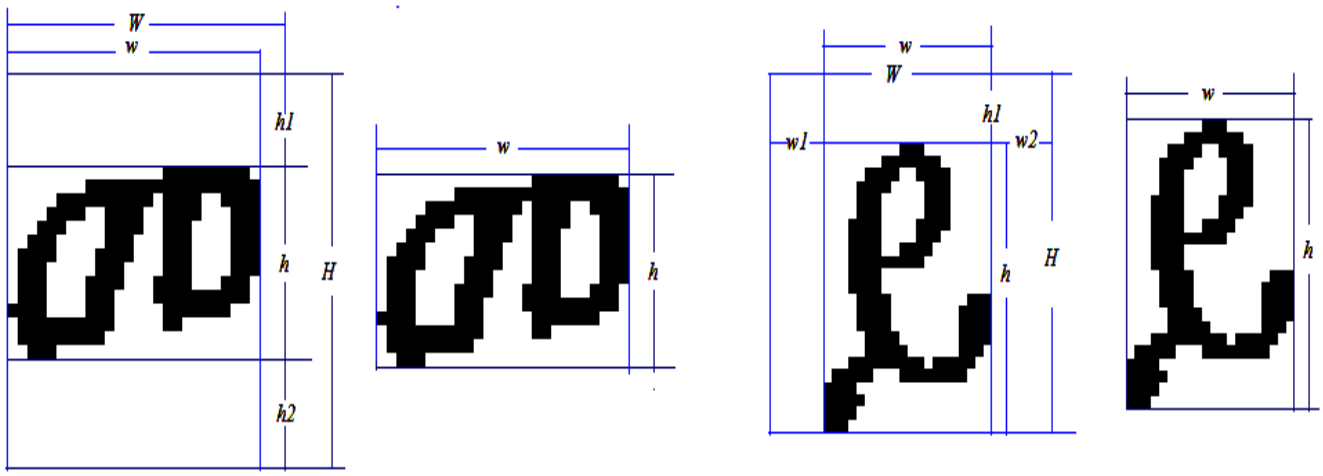


Figure 4.36 –Elimination of Extra width and height

From the above sample character image, the total segmented height *H* and width *W* might involves unnecessary vertical or horizontal white spaces indicated by *h1*, *h2*, *w1* and *w2*. They must be removed up to the exact size of the character image (*height* and *width*) becomes *h* and *w* respectively. This study applies vertical projection to eliminate those white spaces.

As compared to previous studies, low performance is obtained in character segmentation because of the document types used in this study are very noisy real life documents with different types of noises. The researcher tested the proposed segmentation technique on '*Addis zemen*' that was used in previous study and 98.67% of characters are segmented accurately and 1.33% are errenous segmentation that are caused by the horizontal disconnection of characters.

4.4 Integrating the proposed system with Amharic OCR System

The proposed skew correction, authomatic page segmentation, noise removal, binaraization and text segmentation methods; are developed using MATLAB R2013 Image Processing Toolbox and Microsoft Visual Studio Professional 2013. The C# programming language is used to develop user interface and some of its image processing libraries are used. The proposed system is integrated with the previously developed recognition system by Michael [43] which includes normalization, feature extraction, and classification stages of OCR system.

In order to see the impact of the proposed segmentation methods on recognition performance, their recogniton accuracy rate is computed for the sample document images from '*Addis Zemen Newspaper*' and *'Megazine'*. The selected sample test documents images containg different

elements with varying layout such as non-text elements i.e. graphics, table lines, and also have layout of single and two column blocks.

First, the document images are preprocessed by the selected skew correction, page segmentation, noise removal and binarization methods. Next, they are segmented using the proposed text line, and word segmentattion methods. Then, characters are segmented using both the developed CC based and vertical projection profile methods to see the recognition performance variation due to the proposed character segmentation methods. Then, each character is normalized by the function adopted from Michael's [43] study. The characters are normalized by *50x50* because of the heuristic to capture all essential information.

Michael [43] produced three models using Nyala, Visual Geez Unicode and combined model. From the developed models, the combined model performs better by scoring 98.94 % accuracy. Due to time limitation, no new model is produced for evaluating the recognition performance. Instead, the combined model used in his study is adopted to evaluate the impact of proposed method on recognition.

For the purpose of feature extraction and testing purpose, modified zoning method and linear multi-class SVM are also adopted from Michael [43]. The algorithm is developed using C# programming which makes it easily compatible for integration with with the proposed system. Table 4.12 presents the effect of the proposed segmentation methods on the performance of recognition result.

Testing Document Image	**Document Image Layout**	**Non-text Elements**	**No. of Characters**	**Recognition Rate**		**Error Rate**	
				Vertical Projection	**Proposed Method**	**Vertical Projection**	**Proposed Method**
Addis Zemen Newspaper	Single Columned	None	900	93.66 %	95.89 %	6.34 %	4.11 %
Sport Newspaper	Single Columned	None	385	82.34 %	85.97 %	17.66 %	14.03 %
Magazine	Single Columned	None	228	41.67 %	61.87 %	58.33 %	38.13 %
Addis Zemen Newspaper	Two-columned	Graphics	1432	59.92 %	75.63 %	40.08 %	24.37 %
Addis Zemen Newspaper	Single Columned	Table Lines	287	17.77 %	76.31 %	82.23 %	23.69 %
		Average	3232	59.07 %	79.13 %	40.93 %	20.87 %

Table 4.12: Summary of Recognition Results for the Sample Testing Document Images

As presented in the table 4.12, the integration of the proposed segmentation methods improves the recongition performance for the selected sample document images. The performance increase is due to the proposed character segmentation method that dealt with some of the weakness of vertical projection profile as presented in the previous section. The selected document contains pictures, tables, column blocks, overlapping and connected characters that cannot be segmented by the previous method. The following figure 4.37 presented sample result of columnized document image that contain graphics using both character segmentation methods after the page segmentation and preprocessing techniques are applied. The shaded characters represent correctly recognized characters whereas non-shaded characters are mis-classified characters.

በጥራትም ከፍተት ያለባቸው ይገኛሉ። ነገር ግን ደግሞ ከዚህ ቀደም የነበረውን አስተሳሰብ የሰበረ ነው።

ለአብነት ለመጥቀስ ያህል አሁን በሀገራችን ባለው ሁኔታ የእንጨት መሰንጠቂያ ማሽኖች ከውጭ ነው የሚመጡት። እነዚህን ማሽኖች እዚሁ በሀገራችን መሥራት እንችላለን የሚለውን ማሳየት አስችሏል። ጥቃቅንና አነስተኛ አንቀሳቃሾች ይሄንን ማሽን ሰርተው ለሌሎች በዘርፉ ለተሰማሩት ማቅረብ ይችላሉ የሚለው ስሜት እንዲፈጥር ለማድረግ የተለያየ ጥረት እያደረግን ነው። እነዚህ ማሽኖች ሞዴሎች አልነበሩም። በተጨባጭ መስራት የሚችሉ ናቸው።

አዲስ ዘመን ፦ እነዚህ ለአነስተኛና ጥቃቅን አንቀሳቃሹ ጥቅም እንዲሰጡ የሚደረገው እንዴት ነው ?

አቶ ቸርነት ፦ በተለይ ጥራት ያላቸውና በተጨባጭ መሥራት የሚችሉ ቴክኖሎጂዎች ለአንቀሳቃሾቹ ጥቅም እንዲሰጡ የሚደረገው በቀጥታ ከእነአሰራር ማንዋላቸውና ዲዛይናቸው እንዲተላለፉ በማድረግ ነው። አንቀሳቃሾቹ እነኝህን የአሰራር ማንዋልና ዲዛይኖች ተከትለው ልክ በዚሁ በተሰጣቸው አግባብ እንዲያባዙ ይደረጋል። አሁን ባለው ሁኔታም በብዙዎቹ ይሄን እያደረጉ ነው። አንቀሳቃሾቹ ይሄን በተለይም የእንጨት መሰንጠቅያና ሌሎች አገልግሎት የሚሰጡ ማሽኖችን ሰርተው በዘርፉ ለሚሰማሩ ማቅረብ ይችላሉ። እያቀረቡም ነው። አንቀሳቃሾቹ እነዚህን ማባዛትና ማቅረብ ከቻሉ ከውጭ የሚገቡ ማሽኖችን ማስቀረትና ወጭን በከፍተኛ ሁኔታ ማዳን እንችላለን።

የቴክኒክና ሙያ ተቋማቱ ሰልጣኞችና ባለሙያዎቹም ሆኑ የአነስተኛና ጥቃቅን ኢንተርፕራይዞች በቅርቡ በኤግዚብሽን ማዕከልና በኢጄንሲው ግቢ ያቀረቧቸው ማሽኖች እዚህ ተመርተው ገበያ ላይ ሲቀርቡ ማማሸግ በማማሸግ፣ አንዳንዴም መቶ በመቶ ቅናሽ እንዳላቸው አይተናል።

አዲስ ዘመን፦ ይሄ የዋጋ ቅናሽ ምን ለስተዋጽኦ አለው?

አቶ ቸርነት፦ የማሽነሪ ዋጋ መቀነስ በሀገር ውስጥ የሚመረቱ ምርቶች ዋጋቸው እንዲወርድ፣ ጥራታቸው እንዲጨምር ያደርጋል። የውጭ

ፎቶ - በሰመ ጎሳ

አቶ ቸርነት በላቸው፣

ምርቶችን ተክተው ተወዳዳሪ እንዲሆኑም ይረዳል። ይሄም የአነስተኛና ጥቃቅን ኢንተርፕራይዞች በገበያ ውስጥ ተወዳዳሪ እንዲሆኑ ከፍተኛ ዕድል ይፈጥራል። እንደ ሀገርም የሥራ ዕድል በመፍጠር እጅግ ከፍተኛ የሆነውን የሥራ አጥነት ቁጥር ይቀንሳል። የውጭ ምንዛሪን እናድናለን። በሌላ በኩልም የሀገራችን ህዝብ የሚፈልጋቸውን የተለያዩ አገልግሎቶችና የምርት ዓይነቶች በተመጣጣኝ ዋጋ እንዲያገኝ ያደርጋል። ይሄን ዓይነት ከፍተኛ የዋጋ ቅናሽ ያለውን ማሽን ወይም ምርት በስፋት አያባዛነውና እየሰራነው በሄድን ቁጥር እድገታችን የሚፋጠንበትን ሁኔታ ይፈጥራል። ይሄን ታሳቢ በማድረግ ነው በርካታ ቴክኖሎጂዎች በየተቋማቱና በየኤግዚብሽን ማዕከሉ እየቀረቡ ያሉት። አብዛኞቹም ውጤታማ ናቸው።

አዲስ ዘመን፦ ከከይዘን የአሰራር ስርዓትና ከሥራ ፈጠራ ጋር ተያይዞ በሥራችሁ ያሉት የቴክኒክና ሙያ ትምህርትና ስልጠና ተቋማት የሚያደርጉት ድጋፍ አለ?

አቶ ቸርነት፦ ተቋማቱ ከከይዘን የአሰራር ጥበብና ፍልስፍና እንዲሁም ከሥራ ፈጣሪነት ጋር ተያይዞ የሚሰጡት ድጋፍ አለ። 28 ተቋሞች አሉን። ስድስቱ ከፍተኛ ፖሊ ቴክኒክ ኮሌጆች ሲሆኑ፣ 22ቱ ደግሞ መካከለኛና ገና እያደጉ ያሉ ናቸው። እነኝህ በመንግስት ደረጃ ሆነው በሥሩ የሚንቀሳቀሱ ናቸው። በድጋፉ ይሳተፉሉ ጥሩ አቅምም አላቸው። ከዚህ በተጨማሪም በግል

(a) Sample original Image

በጫራትም ከፍተቅ ያለባቸው ይገኖሱ ነፃር ግን
ያ፟ጭሞ ከዚ(ቋ፟ም የነበረመኑገ አስተብሰብ የለበረ
ነፀጉ
ለአብነቅ ለቀሩቀሰ ያህዕ አከነ በሀገራቼን
ባለፀኑ ኮኒቃ የአንጨቅ መሰንጠቂያ ማሽኖች
ከውጣው ነጡ የሜመጡት አነዚህን ማሽኖች
አቢው በሀገራችገ መሥራቅ አንችላለን የሚሊውን
ማሳየቸ አስችሏል ሣቃቅንና አነስተኛ
አንቀሳቃሾቼ ይሄንገ ማሽገ ሰርተው ለሌሎች
በ"ርፉ ለተሰማሩቼ ማቅረቱ ይችላለ የሚሊው
ሰመፄት እንዲፈጫር ለመገድረጫ የተለያየ ጭረት
እያደረግገ ነፀጉ እነዚህ ማሽኖች ሞጩች
አልነበሩም በተጨሔባጣጮ መስራቅ የሜችሉ
ናቸፅኑ
አዲስ ዘመገ ፡- ኔነዚህ ለአነስቶኛና
ጭቃቅገ አንቀላቃሹ ጥቅም ኔንዲሰጡ
የሜደረገፅሙ እንዴቅ ነው ?
አቆ ቸርነት ፡- በተለይ 'የራቅ ያላቸውና
በተጨሌባዳው መሥራቅ የሜቸለ ቴኬኖሎጂዎች
ለአንቀላቃሹቼ ፓቅም እንዲሰጡ የሜደረገው
በቀጥቃ ከአነአሰራ(ማንዋላቸውሚ ዲዛይናቸው
እንዲተላለፉ ዐማድረጫ ነመኑ አንቀሳቃሾቹ አነነህን
የአሰራዲ ፔ፟ጅልና ዲዛይናኝ ተከትለጡ ልክ በቢሁ
በተሰጣቸጡ አጫዐቁ እንዲያ፟ባዙ ይደረጋዕ አውን
ባለመኑ ዐኔታም በ-ዙዎኝ ይሄገ እያደረጉ ነፀጉ
አንቀሳቃሾቹ ይሄገ በተለይም የአን፟ሔት
መሰንጠቅያሜ ሊሎ፟ኝ አ፟ልነሎቅ የሚሰጡ ማሽኖችን
ሰርተው በ"ርፉ ለሜሰ፟ጊሩ ማቅረብ ይችላሉ
እ፟ቀረቡም ነፀሙ አንቀሳቃሾቹ አነዚህገ ማባዛትና
ጨገቅረብ ከቻሲ ክውጣጮ የሜነጡ ማሽኖችን
ማስቄረቅና ውጫጫን በከፍተኖ ሁኔቃ ማዳን
አ፣ሳለገ ።
የቴከኒከና መፀ፟ያ ተቋመገቴ ሰልጣኞችሜ
ባለሙያዎቹም ሆኑ የአነስተኛሜ ኖቃቅገ ኢንተርፕ
ራይዞ፤ በሾጡ በኤሜዚብሽገ ማፅከልሜ በኤ፟ጂነሲው
ሣቢ ፟ቀረቧ፟ቸው ማሽኖች አዚህ ተመርተፀጉ ገበያ
ላደ ሊቀርጡ ነማሽ በነማሽ አንዳንዴም መዶ
በመቆ ቅናሽ እንዳላቸው አይተናል
ጳ፣ፅ ፔ፟መ፯ቋ፟ቋ፟ ምሞኃ ሙየ፟ ምሃ
ጸፅቀሞ፟ሠ ጳጸሙዖ
ጳ ቆ ቾ፟ሙዶ የማሽነሪ ዋጋ መዓነሰ በሀገር
ጡሰጫ የሚመረቱ ምርቶ፣ ዋጋቸው እንዲወርድ
ፓራታቸጡ እንዲጨምር ያደርጋል የጩኑጣጮ

(i) Column one result

ምርቶችን ተከተው ተወዳዳሪ አንዲሆኑም ይረዳል
ይሄም የአነስተኛና ጥቃቅን ኢንተርፕራይዞች
በጎበ፟ኟ ውስጫ ተወዳዳሪ እንዲሆኑ ከፍተኞ ዕድል
ይፈጫራፅ አንፔ ህጎርም የሥሪ ፅድፅ በመሜፅዉር
እ፟ጭጫ ከፍተኛ የሆነፀኑገ የሥራ አጫነት ቄየር
ይቀንላፅ የወጣው ምንዛሪን እናድናሰገ በሌላ
በኪልም የሠራችገ ፔ፟ገፔዐ የሜፈልጋቸው! የተለ፟ዩ
አጎልጫሎዶቀሩ የምርቅ ዓይነቶኝ በተመጣጣሜ ዋጋ
አንዲያጎኝ ያደርጋል ይሄን ዓይነቅ ክፍተኛ የዋጋ
ቅናሽ ፔ፟ለ፟ቀሩገ ማሽገ ወይም ምርት በስፋጭ እያባ
ዛነውና አየሰራነው በሄድገ ቋ፟ር ኔድገቃችን
የሜፉጠንበትን ውኔቃ ይፊጥራል ይሄገ ታሳቢ
በማድረጫ ነመኑ በርካቃ ቴከኖሎጂዎኝ በየቴቋማቱና
በየኤጫዚብሽን ማፅከለ አየቀረጡ ያሉጭ
አብዛኞቹም ውጤታማ ናቸው
አዲፅ ዘመን ከክይ"ገ የአሰራር ስርዓቅና
ከሥራ ፈወራ ታ፟ር ቶያይዘ በሥራቸው ያሉት
የቴከኒከና ሙቄ ቅምህርቅና ስልጠና ተቋማት
የሚያደርጉት ድጋፍ አለ?
አቆ ቸርነት ፡- ተጳማቴ ከክይ"ን የአሰራር
ፓ፟በብሜ ፍልስፍሜ እንዲኸም ከሥሪ ሪጣሪነጭ ጋር
ተያይው የሚሰጡች ድጋ፟ና አለ 28 ተቋሞች
ኤሉን ስድስቱ ከፍተኛ ፖሊ ቴክኒክ ኮሌጆች
ሲሆኑ፤ 22ቴ ደነሞ መካከለኛና ገፔ እያደጉ ያለ
ናቸው አነኛህ በመንፓ፟ስቅ ደረጳ ሆነፀቻ በሥ4
የሚንተሳተሱ ናቸው በድጋፉ ይዳተቆሉ የሩ
አፔሃሜመ አላቸው ከዚህ በተጨሌማሪም በግል

(ii) Column two result

(b) Sample recognition result using the proposed character segmentation method

(i) Column one result *(ii) Column two result*

(c) Sample recognition result using vertical projection profile character segmentation method

Figure 4.37 –Sample Recognition Results

The above result shows, vertical projection profile considers connected and overlapped characters as a single character that causes character misclassification. Though, the proposed method first segments the overlapping characters and detect the connected characters as well. Second the algorithm splits the connected characters based on the location (i.e. x and y) height, width and area analysis of the components as discussed in section 4.3.3. Some of the examples of detected connected characters in the above expirment are ‘800’, ‘ዓመት’, ‘በረ’, ‘ወቅ’, ‘ነበ’, ‘ስረ’, ‘ይቶ’, ’ስቶ’, ’ʒ00’, ‘00’, ‘0)’, ’ሲሆ”, etc.. All of them were sucessfully segmented, normalized and their feature is fed into the classifier inorder to be recognized.

However, low recognition rate is obtained in some of the cases due to the erroneous segmentation of characters having a discontinuity in their body and connected characters as well. Figure 4.38 presents some of the erroneous segmentations that causes mis classifications.

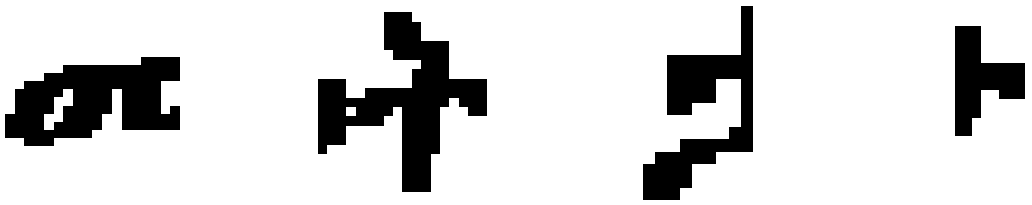


Figure 4.38 –Examples of Erroneously segmented characters

Also, another reason for the mis-classification is the problem of the recognition performance for similar characters. For example, it recognizes similar characters such as ‘ን’, ’ተ’, and ‘በ’ as ‘ገ’, ‘ቀ’, and ’ጠ’ respectively.

The recognition result using the proposed method performs better than the previously used vertical projection profile towards segmenting the overlapping and connected characters from the document image. However, the major challenge for mis-segmentation of characters is document degradation and the variation in the size of Amharic character sets.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

A huge amount of real life documents such as books, newspapers, magazines, and reports that contains vital information regarding various aspects are available inside churches, caves, governmental and private institutions including information centers, libraries, and museums. Those documents might contain different components rather than text and they might be written in a printed, typewritten and hand-written format.

To make these rich information sources accessible, reachable and searchable, there is a need to convert them into their corresponding computer representation with high accuracy. Therefore, designing OCR systems is vital and researchers made an attempt to develop recognition system for different languages.

This research is a continuation of other attempts to develop an OCR system for printed real life document images with varying noise levels. Here, various image pre-processing algorithms such as skew detection and correction, noise removal, and binarization methods are adopted to fix the deficiencies found in real life document images. Also automatic page segmentation methods are developed to tackle the problem in recognition of real life documents that contains tables, lines, graphics, logos, column blocks and titles. Additionally, some text segmentation techniques to segment text lines, words and characters from the text image are experimented.

To deal with the difficulty in text line segmentation from document images with an ink-bleeding noise, a new method based on smoothing, binarization and projection profile is proposed. Also for segmenting connected and overlapping characters, another method that is based on connected component analysis is developed and integrated with Amharic OCR system.

5.1 Summary and Conclusion

The main objective of this research is to design and integrate an effective image preprocessing, page segmentation (i.e. table, text/graphic, column block and title segmentation) and text segmentation (i.e. lines, words, characters segmentation) techniques to improve the effectiveness

and applicability of Amharic OCR for real-life document images with different levels of degradations.

To achieve this, after the digitization of document images, skew angle of each document image is checked and corrected. The goal of skew detection is to check and fix the skewness error made at the time of image acquisition. The Hough transform based method of `AForge.Imaging` library is imported and used for the detection and correction of rotation angle.

Following this, automatic page segmentation methods that are based on hough transform, morphological dilation and cc analysis is developed to be applied on the de-skewed document image to detect table and other lines, pictures, logos, drawings, graphics and column blocks from the document image and extract the text area. Based on performance result, an accuracy of 90.47%, 92.31%, 96.67% and 77.59% is obtained to segment tables, text/graphics, columns and titles respectively. For this purpose, the threshold is calculated automatically using the height, width and area of each component in the document image.

Text/graphics and column block segmentation provides an advantage of excluding noises found in the border of the document and only text area is processed. Then, the extracted text area is preprocessed by noise removal and thresholding techniques to clean the deficiencies found in the document image. A combination of three noise filtering algorithms along with two binarization methods is tested to see their performances. From the candidate methods, wieners filtering with 3x3 window size and sauvola thresholding with 20x20 is found to perform best for document images with varying backgrounds.

Following this, two text lines segmentation methods are experimented to extract text lines from document images. From the candidate methods, an integration of morphological dilation with horizontal projection profile performs better to segment text lines from both training and testing datasets. The algorithm successfully segments 99.09% of the training sets and 100% of both low and medium level noise images, 79.73% and 75.44% of both high and very high level noise testing images respectively. For high and very high noise levels, low performance is registered due to the presence of ink-bleeding noises on those images and a new method based on image smoothing by wiener filtering and sauvola thresholding with projection profile is introduced and an improved accuracy of 99.18% is obtained.

After segmenting out lines, two word segmentation techniques are tested. Vertical projection profile method is selected as best performer by segmenting 99.23%, 96.24%, 87.12% and 54.80% for low, medium, high and very high level noise document images respectively.

To segment characters from the given word image, two algorithms are experimented. A new method based on cc labeling is developed to solve some of the weaknesses of each method. It gives an encouraging result on the detection and splitting of connected characters, segmentation of overlapping characters, and characters with appendages. It successfully segments 87.61% and 82.29% of characters from low and medium noise level images respectively and 50.64% of high and very high noise level images. Finally, the extracted characters are normalized by *50x50* using bi-cubic interpolation method and fed into the modified zoning feature extraction technique.

By integrating the proposed techniques with the previous Amharic OCR system, recognition performance for the selected sample test documents from '*Addis Zemen Newspaper*' and '*Magazine*' is measured. The documents include both single and two column blocks which either contains non text elements such as graphics and table lines or not. First, all the document images are processed using the proposed methods and recognition is measured using the two character segmentation methods and an increase in an average of 20.06% recognition rate is obtained in noisy document images. All the expirments are made by using the previously developed combined model by Michael [43].

The single columned testing pages from *'Addis Zemen Newspaper'*, *'Sports Newspaper*, and *'Magazine';* that does not contain non-text parts are preprocessed and segmented by both the previously used and the proposed system. 93.66%, 82.34% and 41.67% recognition rates are registered using vertical projection profile whereas the proposed method registered 95.89%, 85.97% and 61.87% recognition rates respectively. For the two-columned testing pages from *'Addis Zemen Newspaper'* that contains graphics, 59.92% and 75.63% recognition rates are achieved for the previous and new method respectively. For the single columned document that contain table lines, 17.77% and 76.31% recognition rates are achieved for the previous and new method respectively.

In this study, it is observed that the developed CC labeling based character segmentation method is good enough to segment overlapping characters and characters having upper lines and appendages. The algorithm performs better to detect connected characters and split them based

on their width, and the developed rules to protect erroneous detection and splitting of characters based on word image and component size also performs better.

However, the variation in sizes of Amharic characters and the discontinuity of the characters caused by degradation are some of the identified causes for an erroneous detection and segmentation of connected characters in real life document images as well as the major reason for the erroneous recognition. Another major reason of mis-classification of characters is the performance of recognition algorithms to identify structurally similar and degraded characters.

5.2 Recommendation

The current research work made an attempt to improve the recognition rate of the Amharic OCR system for real life documents by adopting and developing different image preprocessing and segmentation techniques. Nevertheless, for further improvement of the performance of Amharic OCR system, the following recommendations are forwarded.

- The application of OCR system for real life documents is a challenge because of the current segmentation techniques did not detect different layouts and objects from those documents. Therefore, exploring adaptive page segmentation techniques is needed.
- The employed table and other lines (i.e. overline and underline) segmentation technique did not detect short and broken lines, which eventually causes erroneous segmentation and classification of characters. Therefore, better line detection methods have to be explored to detect those short lines.
- The text/graphics segmentation method applied in this study detects some large titles as a graphics. Thus, tackling this problem is one of the future research directions.
- The column block segmentation technique explored in this study is dependent on the white space found between column blocks, which failed to segment overlapping columns. Thus, there is a need to explore more flexible column block detection.
- The employed technique for title detection is based on the location and size of the title. It failed to segment titles with smaller fonts and misses some titles from the middle of the page. Future researches can also improve the limitations in title detection by exploring a technique that can tackle such problems.

- The image denoising and thresholding technique adopted in this study erodes some pixels from the character images, which ultimately create erroneous character segmentation and recognition. Therefore, a learning image restoration technique that can restore only characters body needs to be explored.
- The image denoising and thresholding techniques explored to be used in the detection of text line from ink-bleedings cannot exclude ink-bleedings that are on the same line with the front text line, which makes word and character segmentation very difficult. Thus, there is a need to develop a robust image denoising technique that can separate the bleeded ink from the text line.
- The adopted text segmentation for an ink-bleeded image misses some of the character body and appendages as well. Therefore, there is a need to enhance the text line segmentation method for document images with ink-bleeding images.
- Since there is a possibility that noises might fill the gap which exists between words and characters, segmentation of words and characters is a challenge. Thus, there is a need to explore a method that can extract words and characters from noisy document images.
- The developed character segmentation technique did not segment characters with horizontal discontinuity. Thus, a learning character segmentation technique needs to be explored.
- Some long characters are detected as connected character that causes mis-classification of the character. Thus, there is a need to enhance the detection technique.
- This study uses the splitting of characters into equal (i.e. 5, 4, 3 or 2) parts and it works well for two connected characters having similar size. But there is a need to develop flexible splitting method that is robust for the varying width of Amharic characters.
- The recognition result shows mis-classification of similar characters. Thus, advanced feature extraction and recognition techniques must be explored.
- Most of the historic documents contain ink-bleeding noises and previous studies have not experimented on those types of document images. Thus, a benchmark test set that contain ink-bleeded images for Amharic document images needs to be constructed and studies have to be made on them.

References

[1] Abay, T. (2010). *Amharic OCR System for Printed Real Life Documents.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[2] Abby Technologies (https://www.abbyy-developers.eu), Accessed at February 12, 2015.

[3] Abinet, S. (2005). *Online Handwriting Recognition for Ethiopic Characters.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[4] Acharya, T., & Ray, A. K. (2005). *Image processing: principles and applications.* John Wiley & Sons.

[5] Adane, L. (2011). *Feature extraction and matching in Amharic recognition system. (Masters Thesis)*, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia

[6] Alginahi, Y. (2010). Preprocessing techniques in character recognition. *INTECH Open Access* Publisher, pp. 1-20.

[7] Alginahi, Y. (2013). A survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition (IJDAR), vol. 2, no. 16, pp. 105-126.*

[8] Antolovic, D. (2008). Review of the Hough transform method, with an implementation of the fast Hough variant for line detection. *Department of Computer Science, Indiana University*.

[9] Bar-Yosef, I., Hagbi, N., Kedem, K., & Dinstein, I. (2009, July). Line segmentation for degraded handwritten historical documents. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on IEEE (pp. 1161-1165).*

[10] Baye, Y. (1992). (Ethiopian) Writing System. Addis Ababa, Ethiopia: *Addis Ababa University. Dialogue, vol. 1, no. 1.*
http://www.ethiopians.com/bayeyima.html (accessed March 20, 2015).

[11] Berhanu, A. (1999). *The application of OCR Techniques to the Amharic Script.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[12] Bieniecki, W., Grabowski, S., & Rozenberg, W. (2007). Image preprocessing for improving OCR accuracy. In *Perspective Technologies and Methods in MEMS Design. MEMSTECH 2007. International Conference on IEEE (pp. 75-80).*

[13] Biniyam, A. L. (2012). *Retrieval from Real-Life Amharic Documents.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[14] Blanchet, G., and Charbit, M. (2014). Digital Signal and Image Processing using MATLAB. *London and John Wiley & Sons.*

[15] Bukhari, S. S., Shafait, F., & Breuel, T. M. (2011, January). Improved document image segmentation algorithm using multiresolution morphology. In *IS&T/SPIE Electronic Imaging (pp. 78740-78740*). International Society for Optics and Photonics.

[16] Chaudhuri, B. B., & Pal, U. (1998). A complete printed Bangla OCR system. *Pattern recognition*, *vol. 31, no. 5, pp. 531-549.*

[17] Cheriet, M., Kharma, N., Liu, C. L., & Suen, C. (2007). *Character recognition systems: a guide for students and practitioners*. John Wiley & Sons publication (Wiley Inter science), *pp. 1-4*.

[18] Cowell, J., & Hussain, F. (2003, July). Amharic character recognition using a fast signature based algorithm. *In Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on IEEE (pp. 384-389).*

[19] Cristóbal, G., Schelkens, P., & Thienpont, H. (Eds.). (2013). *Optical and digital image processing: fundamentals and applications.* John Wiley & Sons.

[20] Das, M. S., Reddy, C. R. K., Govardhan, A., & Saikrishna, G. (2010). Segmentation of Overlapping Text lines, Characters in printed Telugu text document images. *International Journal of Engineering Science and Technology*, vol. *2, no. 11, pp. 6606-6610.*

[21] Dereje, T. (1999). *Optical character Recognition of Typewritten Amharic Text.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[22] Due Trier, Ø., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. *Pattern recognition, vol. 29, no. 4, pp. 641-662.*

[23] Eikvil, L. (1993). *Optical Character Recognition.* Oslo: Document Image Analysis Publications.

[24] Ermias, A. (1998). *Recognition of Formatted Amharic Text Using OCR Techniques.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[25] Gautam, A. (2013). Segmentation of Text from Image Document. *International Journal of Computer Science and Information Technologies, Vol. 4, no. 3, pp. 538-540.*

[26] Gedion, A. L. (2013). *Page Segmentation in Amharic Document Image Collections* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[27] Genovese, J. A. (1970). *Character Recognition: Encyclopedia of Library and Information Science.* New York, Marcel Deker Inc, USA *Vol. 4, no. 3.*

[28] Gupta, G. (2011). Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter. *International Journal of Soft Computing and Engineering, vol. 1, no. 5, pp. 2231-2307.*

[29] Hough Transform, Open CV 2.4.9.0 Documentation.

[30] http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm. Accessed at: April 10, 2015.

[31] https://www.aforgenet.com. Document Skew Checker Class of AForge.NET Framework. Accessed at: March 25, 2015.

[32] https://www.bcs.uconn.edu. Accessed at: March 25, 2015.

[33] https://www.interactiondesign.se Being Human: Human-Computer Interaction in the year 2020. *Editors: Richard Harper, Tom Rodden, Yvonne Rogers and Abigail Sellen.* Accessed at: January 16, 2015.

[34] Jiawei, H., & Kamber, M. (2001). Data mining: concepts and techniques. *San Francisco, CA, itd: Morgan Kaufmann.*

[35] Jurafsky, D., & James, H. (2000). *Speech and language processing.* An introduction to natural language processing, computational linguistics, and speech. *Prentice Hall Series In Artificial Intelligence.*

[36] Kaur, E. H. (2014). A Survey of Feature Extraction and Classification Techniques Used In Character Recognition for Indian Scripts. *An International Journal of Engineering Sciences, Vol. 3, no. 3, pp. 238-241.*

[37] Kaur, S., Mann, P. S., & Khurana, S. (2013). Page Segmentation in OCR System-A Review. *An International Journal of Computer Science and Infrmation Technologies, Vol. 4, no. 3, pp. 420-422.*

[38] Kieri, A. (2012). Context Dependent Thresholding and Filter Selection for Optical Character Recognition (Masters Thesis), Uppsala, Sweden: Uppsala University.

[39] Kothari, C. R. (2004). *Research Methodology: Methods and Techniques, Types of Researches.* New Delhi, India: New Age International Publishers (NAIP), pp. 1-4.

[40] Krishnamoorthy, S., Loganathan, R., & Soman, K. P. (2010). Recursive Projection Profiling for Text-Image Separation. In *Innovations in Computing Sciences and Software Engineering* (pp. 1-5). Springer Netherlands.

[41] MathWorks (2013). *Matlab Image Processing Toolbox User Guide*. The MathWorks, Inc.

[42] Mesay, H. M. (2003). *Line Fitting To Amharic OCR: The Case of Postal Address.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[43] Michael, A. (2014). *Recognition of Real-Life Amharic Document Images*. (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[44] Million, M. (2000). *A Generalized Approach to Optical Character Recognition (OCR) of Amharic Texts.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[45] Million, M. (2008). *Recognition and Retrieval from Document Image Collections* (Doctoral dissertation), Hyderabad 500 032, India: International Institute of Information Technology.

[46] Million, M., & Jawahar, C. V. (2005). Recognition of printed Amharic documents. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*IEEE, pp. 784-788.

[47] Million, M., & Jawahar, C. V. (2007). Indigenous scripts of African languages. *Indilinga African Journal of Indigenous Knowledge Systems, vol. 6, no. 2,* pp. 132-142.

[48] Million, M., & Jawahar, C. V. (2007). Optical character recognition of Amharic documents. *African Journal of Information & Communication Technology, vol. 3, no. 2, pp. 14, ISSN 1449-2679.*

[49] Mitchell, T. M. (2006). *The discipline of machine learning*. Carnegie Mellon University, School of Computer Science, Machine Learning Department.

[50] Mori, S., Suen, C. Y., & Yamamoto, K. (1992). Historical review of OCR research and development. *Proceedings of the IEEE*, vol. *80,* no. 7, pp. 1029-1058.

[51] Motwani, M. C., Gadiya, M. C., Motwani, R. C.c & Harris Jr, F. C. (2004). Survey of image denoising techniques. *In proceedings of GSPX, pp. 27-30.*

[52] Nigussie, T. (2000). *Handwritten Amharic Text Recognition Applied to the Processing of Bank Checks.* (Masters Thesis), Addis Ababa, Ethiopia: School of Information Studies for Africa, Addis Ababa University.

[53] Patel, U. An Introduction to the Process of Optical Character Recognition. *International Journal of Science and Research (IJSR), India. Vol. 2, Issue 5, May 2013. pp. 155-158.*

[54] Pratap, N., & Arya, D. S. (2012). A Review of Devnagari Character Recognition from Past to Future. *International Journal of Computer Science and Telecommunications, vol. 3, no. 6, pp. 77-82.*

[55] Randhir S., & Priya, S. (2013). A Survey Paper on Character Recognition. *International Journal for Scientific Research and Development, vol. 3, Issue 9, ISSN (online) 2321-0613.*

[56] Randhir S., & Priya, S. (2013). English Character Recognition System: Digitization of Printed Documents. *International Journal of Computer & Organization Trends, vol. 3, Issue 6, pp. 231-234.*

[57] Randhir S., & Priya, S. (2013). Optical Character Recognition. *International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, Issue 1, ISSN 2278-1021.*

[58] Randriamasy, S., & Vincent, L. (1994, June). Benchmarking page segmentation algorithms. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on* (pp. 411-416). IEEE.

[59] Saha, S., Basu, S., Nasipuri, M., & Basu, D. K. (2010). A Hough transform based technique for text segmentation. *arXiv preprint arXiv:1002.4048.*

[60] Sethi, R. K. (2014). *Use of Adaptive Methods to Improve Degraded Document Images* (Doctoral dissertation, National Institute of Technology Rourkela).

[61] Sharma, O. P., Ghose, M. K., Shah, K. B., & Thakur, B. K. (2013). Recent trends and tools for feature extraction in OCR technology. *International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 6, pp. 220-223, ISSN: 2231-2307.*

[62] Shrestha, S. (2014). Image Denoising using New Adaptive Based Median Filters. *arXiv preprint arXiv:1410.2175.*

[63] Verma, Rohit, and Jahid Ali. *(2012).* "A Survey of Feature Extraction and Classification Techniques in OCR Systems". *International Journal of Computer Applications & Information Technology (IJCAIT) Vol. 1, no. 3: pp. 1-3.*

[64] Wikipedia, http://www.en.wikipedia.org Accessed at: April 18, 2015.

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

[66] Worku, A. (1997). *The application of OCR Techniques to the Amharic Script.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[67] Yaregal, A. L. (2002). *Optical character recognition of Amharic text: an integrated approach.* (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University.

[68] Yaregal, A., & Josef, B. (2006). Ethiopic Document Image Database for Testing Character Recognition Systems.

[69] Yaregal, A., & Josef, B. (2006, August). Ethiopic character recognition using direction field tensor. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* IEEE, *vol. 3, pp. 284-287*.

[70] Yaregal, A., & Josef, B. (2007). Multifont size-resilient recognition system for Ethiopic script. *International Journal of Document Analysis and Recognition (IJDAR), vol. 10, no. 2, pp. 85-100*.

[71] Yaregal, A., & Josef, B. (2007, August). A neural network approach for multifont and size-independent recognition of Ethiopic characters. In *Progress in pattern recognition pp. 129-137*. Springer London.

[72] Yaregal, A., & Josef, B. (2007, September). A Hybrid System for Robust Recognition of Ethiopic Script. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on IEEE, vol. 1, pp. 556-560*.

[73] Yaregal, A., & Josef, B. (2008). Online Handwriting Recognition of Ethiopic Script. In *Proceedings: Eleventh International Conference on Frontiers in Handwriting Recognition, Montréal, Québec-Canada, August 19-21, 2008*. Montréal: CENPARMI, Concordia University.

[74] Yaregal, A., & Josef, B. (2008). Writer-independent offline recognition of handwritten Ethiopic characters. *Proc. 11th ICFHR, pp. 652-656.*

[75] Yaregal, A., & Josef, B. (2008, December). Lexicon-based offline recognition of Amharic words in unconstrained handwritten text. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on IEEE, pp. 1-4.*

[76] Yaregal, A., & Josef, B. (2009). Offline Handwritten Amharic Word Recognition Using HMMs.

[77] Yaregal, A., & Josef, B. (2009, July). HMM-Based Handwritten Amharic Word Recognition with Feature Concatenation. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on* IEEE, *pp. 961-965.*

[78] Yaregal, A., & Josef, B. (2011). Offline handwritten Amharic word recognition. *Pattern Recognition Letters, vol. 32, no. 8, pp. 1089-1099.*

[79] Yaregal, A., Premaratne, L., & Josef, B. (2004). Recognition of Modification-based Scripts Using Direction Tensors. *In ICVGIP vol. 4, pp. 587-592*.

Appendices

Annex I: The Amharic Writing System (Fidel)

							Labialized				
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ			ሗ		
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ			ሧ		
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ			ቯ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኍ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ			ኧ		
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኵ	ኳ	ኴ	ኵ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ			ዧ		
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ			ጇ		
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጕ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ			ፗ		

Numerals									
	1	2	3	4	5	6	7	8	9
× 1	፩	፪	፫	፬	፭	፮	፯	፰	፱
× 10	፲	፳	፴	፵	፶	፷	፸	፹	፺
× 100	፻								
× 10.000	፼								

Punctuation Marks	
፡	።
፣	?
፤	()
!	" "

Annex II: Sample Test sets

Test Set I: Sample Document Image Containing Table

የጨረታ ጥሪ
የጨረታ መለያ ቁጥር ኢ/መ/ደ/ኤ/
ግልጽ ጨረታ ፕሮጀክት 10/07

የኢንፎርሜሽን መረብ ደህንነት ኤጀንሲ LOT 1 server application፤ Server እና TELEVISION ዕቃዎች በሀገር ውስጥ ግልጽ ጨረታ አወዳድሮ ለመግዛት ይፈልጋል።

<u>በጨረታው መወዳደር የሚፈልጉ ተጫራቾች፡-</u>

ሀ. በመንግሥት ግዥና ንብረት አስተዳደር ኤጀንሲ <u>ድህረ ገፅ</u> በአቅራቢዎች ዝርዝር ውስጥ የተመዘገበ እና የምዝገባ ፕሪንት አውት ማቅረብ የሚችሉ።

ለ. የተጨማሪ እሴት ታክስ ሠርተፊኬተ የግብር ከፋይ ሰርተፊኬት እና የዘመኑን ግብር የከፈሉበትን ማስረጃ ማቅረብ የሚችሉ

ሐ. በሚወዳደሩበት ዘርፍ የተሰማሩበትን የታደሰ ንግድ ፈቃድ ማቅረብ የሚችሉ

መ. በአማርኛ የተዘጋጀውን የጨረታ ሰነድ ብር 100.00 (አንድ መቶ ብር) በመክፈል ዘወትር በሥራ ሰዓት ይህ ማስታወቂያ ከወጣበት ማግስት ጀምሮ ከኤጀንሲው ፋይናንስ ቢሮ መውሰድ ይችላሉ።

ረ. ተጫራቾች የጨረታ ማስከበሪያ ማስያዝ ይኖርባቸዋል።

ሰ. ተጫራቾች በጨረታ መሳተፍ እንዲችሉ ከአገር ውስጥ ገቢ የተሰጠ ሠርተፊኬት ኮፒ አያይዘው ማቅረብ ይኖርባቸዋል።

<u>የጨረታ ማስከበሪያ መጠን እና ጨረታው የሚዘጋበት ቀን</u>

ሎት	የግዥው ዓይነት	የማስረከቢያ ቦታ	የጨረታ ማስከበሪያ	ጨረታው የሚዘጋበት ቀን	ጨረታው የሚከፈትበት ቀን
ሎት 1	1. server application 2. server 3. TELEVISION	በኢ.መ.ደ.ኤ ስቶር	በሚያቀርቡት ዋጋ ጠቅላላ ላይ 1%	በ 16/7/2007 ዓ.ም 8:30 ሰዓት የጨረታው ሳጥን ይታሸጋል	ተጫራቾች ወይም ሕጋዊ ወኪሎቻቸው በተገኙበት በ 16/6/2007 ዓ.ም በ 8:35 ሰዓት ቴክኒካል ፋይናንሻል ሰነድ ይከፈታል።

ሸ. ተጫራቾች ለሁሉም የግዥ ዓይነቶች ቴክኒካል እና ፋይናንሺያል ሰነድ በተለያዩ ፖስታዎች በሰም በታሸገ ኤንቨሎፕ አሽገው ማቅረብ አለባቸው።

Test Set II: Sample Document Image Containing Graphics

- ***Test Set From Newspaper***

6ኛው ዙር የኤፍ ኤ ካፕ ፍጥጫ - ሊውዝ ቫንሃል ከአር

ታምሩት አበራ

በየሳምንቱ በመላው ዓለም የሚገኙ ስፖርት አፍቃሪያን በጉጉት የሚጠብቁት የእንግሊዝ ፕሪሚየር ሊግ በዚህ ሳምንት በኤፍ ኤ ካፕ 6ኛ ዙር ውድድር ተተክቷል። በእንግሊዝ እግር ኳስ ማህበር (ኤፍ ኤ) አዘጋጅነት ከሚደረጉ የሀገር ውስጥ ውድድሮች ከፕሪሚየር ሊጉ ቀጥሎ ከፍተኛ ትኩረት የሚስበው የኤፍ ኤ ካፕ ውድድር በርካታ ክለቦችን ጥሎ ሲያስተናግድ ዙር ላይ ደርሷል። በስድስተኛው ዙር የኤፍ ኤ ካፕ ውድድር አራት ጨዋታዎች በተለያዩ የእንግሊዝ ከተሞች ይካሄዳሉ።

የመጀመሪያው ጨዋታ ቅዳሜ በእንግሊዞች የምሳ ሰዓት ጨዋታ 9፡45 ላይ የሊግ አንዱ ተወዳዳሪ ብራድፎርድ ሲቲ የሻምፒዮንሺፑን ሬዲንግ በሚጫወትና በደጋፊው ፊት በሚፋለምበት መርሐ ግብር ይጀመራል።

ብራድፎርድ ሲቲ በሊግ እንድ ውስጥ በ48 ነጥብ ስምንተኛ ደረጃ ላይ ይገኛል። ቀጋሚው ሬዲንግ በበኩሉ በሻምፒዮንሺፑ በ41 ነጥብ በአሥራ ስምንተኛ ደረጃ ላይ ተቀምጧል። የሊግ እንድ የደረጃ ሰንጠረዥን ብሬስቶል ሲቲ በ73 ነጥብ፣ ፕሬስተን በ63 ነጥብ እና ሚክዮንስ በ62 ነጥብ ከአንደኛ እስከ ሦስተኛ ያለውን ይዘዋል። የሻምፒዮንሺፑን የደረጃ ሰንጠረዥ ደግሞ ሚድልስቦሮው በ66 ነጥብ፣ ደርቢ ካውንቲ በ65 ነጥብ እንዲሁም ዋትፎርድ በሣብ ተቀድሞ በተመሳሳይ 65 ነጥብ ከአንደኛ እስከ ሦስተኛ ያለውን ደረጃ በመያዝ በሚቀጥለው ዓመት ወደ ፕሪሚየር ሊጉ ለማደግ የሚደረገውን ሩጫ አያባብዱ ይሸኛሉ። ሁለቱ ክለቦች ለስድስተኛ ዙር ጨዋታ የደረሱት ብራድፎርድ ሲቲ ቼልሲና ሰንደርላንድ ጥለው ሲሆን፣ ሬዲንግ በበኩሉ ኩሪዳፍ ሲቲንና ደርቢ ካውንቲን በየሰሙ ግስፍ አሸንፎ መሆኑ ይታወሳል።

ጨዋታው ቀጥሎ ምሽት 2፡30 ላይ አስቶንቪላ በሚጫወቱ ቪላ ፓርክ ላይ ዌስትብሮሚች አልቢዮንን ይዋዥማል። ይህ ጨዋታ ሁለቱ ክለቦች ባለፈው ማክሰኞ በቪላ ፓርክ በ28ኛው ሳምንት የእንግሊዝ ፕሪሚየር ሊግ መርሐ ግብር በተገናኙ ሳይሸነፉ ለአስተኛ ጊዜ የሚገናኙበት ጨዋታ ይሆናል። አስቶንቪላዎች የቀድሞ አሠልጣኛቸውን ፖል ላምበርትን አባረው በምትኩ የቶትንሃም ሆትስፐር ዋና አሠልጣኝ የነበሩትን ቲም ሸሩድን በአሠልጣኝነት ከቀጠሩ በኋላ ጠጣንና የሚባል ለውጥ ያመጡ ሲሆን፤ ሸሩድም የመጀመሪያ ምስት ነጥባቸውን አግኝተዋል። ባለፈው ማክሰኞ የቀኒ ፕሪሴን ዌስትብሮሚች አልቢዮን አስተናግደው ምኢጋ ለጥቃው ይማውና በባክ ሰዓት በተሰናበቸው የፍጻም ቅጣት ምት ማሸነፋቸው እና የመጀመሪያ ምስት ነጥባቸውን ማግኘታቸው ከፍተኛ መነሳሳት ፈጥሮላቸዋል። ዌስትብሮሚች አልቢዮኖች በበኩላቸው በከፍተኛ ጭብ አስቆጣሪያቸው በራሄና ታጣው ከፍተኛ የፕሪሚር ሊጉን ሽንፈታቸውን ለመበቀል ወደ ሜዳ እንደሚገቡ ይጠበቃል።

መርሐ ግብሩ ዕሁድ ቀጥሎ ምሽት 1፡00 ላይ የመርሲሳይዱ ሊቨርፑል በሚጫወቱ አንፊልድ ሮድ ሲታዲየም የሻምፒዮን ሺፑን ብላክበርን ሮቨርስን ያስተናግዳል። ሊቨርፑሎች በፕሬሚየር ሊጉ ባደረጓቸው ተከታታይ ጨዋታዎች በማሸነፍ ከከፋባት የ12ኛ ደረጃ ወደ አምስተኛ ደረጃ መምጣታቸው እና ይህንን ጨዋታ በሜዳቸው የሚያደርጉ መሆኑ ከፍተኛ የማሸነፍ ዕድል

- ***Test Set From Book***

የአዲስ አበባ ከተማ የሀገራቱ መዲና እንደመሆኗ መጠን በነዋሪዎቿ ተሰርተው የተሳለፉ ብዙ ሳይሆኑ ከሥላ ሀገራቱ የተሰባሰቡ ቅርሶች የሚገኙባት ከተማ በመሆኗ የአፍተኛ ቅርስ ነሥችት ባለቤት ለመሆን በቅታለች። ተማሪዎች በአዲስ አበባ ከተማ ውስጥ ያሉ ቅርሶች ምን አይነት እንደሆኑና የት ቦታ እንደሚገኙ ታውቃላችሁ? ለአብነት ያህል የሚከተሉትን ተመልከቱ።

ሀ. ሐውልቶች

ይህ ሐውልት ዘጥያጣ ቅዱስ ጊዮርጊስ ዩኒቨርሲቲን አጠገብ ይገኛል። መታሰቢያነቱ አጎቴዮጵያ በሠራዊቱ የማሲያን ጠር ላይ በአድዋ ላስመዘገባችው ድል ነው።

ጥያቄዎቹን መልሱ

1. በሐውልቱ ላይ በፈረስ ተቀምጠው የሚታዩት ሰው ማን ይባላሉ?

2 የኘህ ሰው ቅርጽ ለሐውልቱ ለምን ተመረጠ? መልሶቹን ከመምህራችሁ ጠ ይታችሁ ተረዱ

ሥዕል 4.2ፀ የአድዋ ድል መታሰቢያ ሐውልት

Test Set III: Sample Document Image Containing Multiple Column Blocks and Titles

ሰጋት እየሆነ የመጣው የትራፊክ አደጋ !

እናት ትምህርት ቤት ብሎ የተለያትን ልጇን ዓይን ሳታይ አባትም ለቤተሰቡ የእለት ጉርስና የዓመት ቀለብ የሚሆን ገቢ ለማፍራት ማልዶ እንደወጣ በዚያው የቀሩበትን፣ ሰጋቸው በእሳት ተለብልቦ ጸታቸውን ለመለየት እስኪያዳግት የደረሰ ዘግናኝ አደጋን ተመልክተናል።

ከሕፃናቱም ባሻገር በርካታ እዛውንቶቹንም ያለጧሪ ቀባሪ እንዲቀሩ ማድረጉ ነው። ባለፈው ጊዜ በፊንፊኔ ዙሪያ ሰበታ ከተማ አካባቢ የተመለከትነውም ይህንኑ አሰቃቂ ሁኔታ ነው። እናት ትምህርት ቤት ብሎ የተለያትን ልጇን ዓይን ሳታይ አባትም ለቤተሰቡ የእለት ጉርስና የዓመት ቀለብ የሚሆን ገቢ ለማፍራት ማልዶ እንደወጣ በዚያው የቀሩበትን፣ ሰጋቸው በእሳት ተለብልቦ ጸታቸውን ለመለየት እስኪያዳግት የደረሰ ዘግናኝ አደጋን ተመልክተናል።

የአፍሪካ መዲና የሆነችው አገራችን በየዓመቱ ማለት በሚቻልበት ሁኔታ በሺዎች የሚቆጠር የሰው ኃይሏን በትራፊክ አደጋ የምታጣ ሲሆን፤ ይህ ኃይልም በመካከለኛ የዕድሜ ክልል ውስጥ የሚገኝ እንደሆነ አንዳንድ መላ ምቶች ያሳያሉ። እ.አ.አ በ2013 የዓለም የጤና ድርጅት ያወጣው አንድ መረጃ እንደሚያመላክተው በዓመት 1 ነጥብ 3 ሚሊዮን የሚሆኑ የዓለማችን ሰዎች በትራፊክ አደጋ ህይወታቸውን የሚያጡ ሲሆን፤ በሺዎች የሚቆጠሩት ደግሞ ለከባድና ቀላል የአካል ጉዳት ተዳርገዋል። ከግማሽ ትሪሊዮን ወይም ከአምስት መቶ ቢሊዮን ዶላር በላይ የሚገመት ንብረት እንደሚወድምም ነው ጥናቱ የሚያመላክተው። ይህን ሁኔታ ወደ አገራችን ነባራዊ ሁኔታ ወስደን ስንመለከተው ደግሞ በተወሰነ ደረጃም ቢሆን ስጋቱ እየደረሰን ስለመሆናችን መካድ ወደማይቻልበት መንገድ ይወስደናል። በአገራችን ኢትዮጵያ ባለፈው ዓመት ብቻ ያለውን የትራፊክ አደጋ ወስደን ስንመለከት 3ሺ331 ሰዎች ህይወታቸውን ሲያጡ ከ13ሺህ በላይ የሚሆኑት ደግሞ ለከባድና ለቀላል የአካል ጉዳት ተዳርገዋል። ከግማሽ ቢሊዮን ብር በላይ የሚገመት ንብረትም መና ቀርቷል።

አሰፈዳሚው አካል

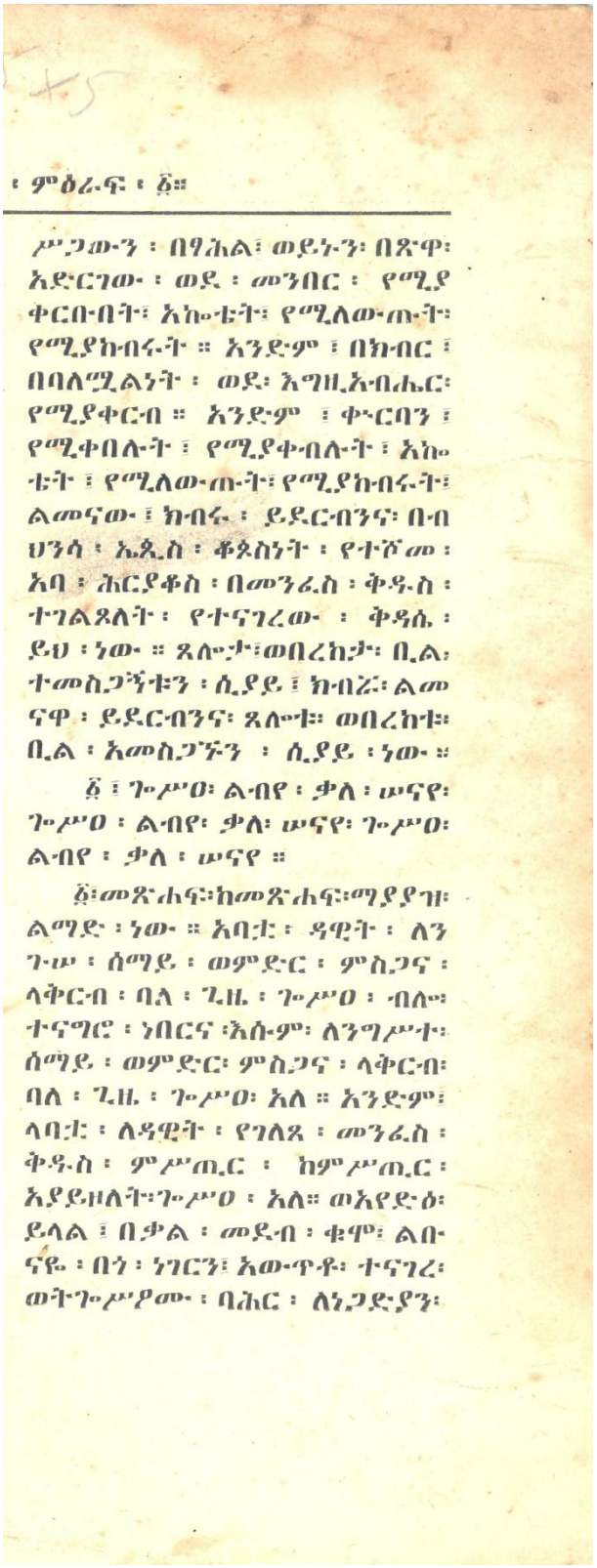
ይህ አካል እየደረሰ ያለውን የትራፊክ አደጋ በመቀነሱ ረገድ ያለበት ኃላፊነት ከፍተኛ ነው የሚል ጽኑ ዕምነት አለኝ። ምክንያቱም ዘርፉ የሚተዳደርበትን ሕግና መመሪያ በማዘጋጀትም ይሁን ተፈጻሚ እንዲሆን በህዝብ የተሰጠው ሥልጣን ቀላል የሚባል አይደለም። ይሁን እንጂ እየተስተዋለ ያለው ግን ከዚህ በተቃራኒው የሆነ ጉዳይ ሆኖ እናገኘዋለን። በትራፊክ አደጋ እየሞቱ ካሉት የኅብረተሰብ ክፍሎች መካከል ከ5 እስከ 50 ዓመት የዕድሜ ክልል ያሉትና የአደጋው 75 በመቶ በላይ የሚሆኑት ዜጎች ጉዳይ እያሳሰበው ያለም አይመስለኝም። የአገሪቱን የህዳሴ ጉዞ በማሳለጡ ሂደት ውስጥ የነቃ ተሳትፎ ያላቸው የኅብረተሰብ ክፍሎች ጉዳይ ካላሳሰበ አጠቃላይ የሆነው የአገር እድገትም ከግንዛቤ ውስጥ አልገባም ማለት ነው።

በአደጋ ምክንያት የሚወድሙት ንብረቶችም ከፍተኛ የውጭ ምንዛሪ ተከፍሎባቸው ወደ አገር የሚገቡ እንደመሆናቸው መጠን ተፈላጊውን አገልግሎት እንኳን በአግባቡ ሳይሰጡ ከጥቅም ውጪ መሆናቸውንም ይኸው አካል ከግምት ውስጥ ማስገባት ይኖርበታል። በመሆኑም ቆንጣጭነት ያላቸውን ህጎችና እርምጃዎች መውሰድ አለበት። ከዚህም ባሻገር የቸልተኛና የዕውቀት አልባ አሽከርካሪ መፈልፈያ የሆኑትን የማሰልጠኛ ጣቢያዎችም በመከታተል ተገቢውን የእርምት እርምጃ መውሰድ ይኖርበታል የሚል የጸና እምነትና አቋም አለኝ። ከዚህም ባሻገር ከዚህን ቀደም የነበረውን የመንጃ ፈቃድ አሰጣጥ የጊዜ ርዝማኔ በመፈተሽ ተግባራዊ ማድረግ በሚቻልበት ሁኔታ ላይ መወያየትም ይኖርበታል። ቀደም ሲል የነበረውን የመንጃ ፈቃድ አሰጣጥ ሂደት ለእናንተ አንባቢያን ለማስታወስ ያህል የሁለት ዓመታት የጊዜ ቆይታ የነበረበት ነው። ይህም ማለት የአንደኛ ደረጃ መንጃ ፈቃድ ያወጣ ሰው ሁለተኛ ደረጃን ለማግኘት ቢያንስ ሁለት ዓመታትን ይቆያል ማለት ነው።

ይህን የአገራችንን ኢቦላ በሽታም በላይ የሆነ የትራፊክ አደጋ ችግር እያደረሰ የሚገኘውን ጉዳት ለመቀነስ ይቻል ዘንድ የአስፈጻሚው አካል ጥንካሬ ጎልቶ እስካልታየ ድረስ ኅብረተሰቡና የሚመለከታቸው ሌሎችም ባለድርሻ አካላት ለጉዳዩ ልዩ ትኩረት ሰጥተው ይሠራሉ ብሎ ማሰብ ሞኝነት ነው። የመዲናዋን ብሎም የአገሪቷን የእድገትና የዜጎች ጸር የሆነውን የትራፊክ አደጋ በመቀነሱም ሆነ እስከመጨረሻው በማጥፋቱ ረገድ ይህ አካል ተገቢውን ሥራ ለመሥራት ዛሬውኑ ከተኛበት ይንቃ!

ባጠቃላይ በአገራችን እየተስተዋለ ያለው ዘግናኝ የትራፊክ አደጋ አሁን በተጀመረው የመከላከል መንገድ ብቻ ይቀንሳል የሚል ዕምነት የለኝም። በመሆኑም የተለየ የማስተማሪያ ስልቶችን ይዘን መቅረብ አስፈላጊ ነው። የአሽከርካሪነት ሙያን እንደጊዜ ማሳለፊያ ወይንም ለዕለታዊ ጥቅም ብቻ ሲሉ የሚሹት አሽከርካሪዎችም ቢሆኑ ትራፊኮችን ብቻ ሳይሆን ሙያውንም ማክበር ይጠበቅባቸዋል። ስለ እያንዳንዱ ዜጋ ውድ ህይወትና ስለ ሚያሽከረክሩት የህዝብም ሆነ የመንግሥት ንብረት ዘወትር መጨነቅ አለባቸው። ልክ ኢቦላን በአጥፊነቱ የፈራነውን ያህል የትራፊክ አደጋንም በህይወት አጥፊነቱ እንፍራው መልዕክቱ ነው።

Test Set IV: Sample Document Images from Qidassie Mariam (Low Level Noise)

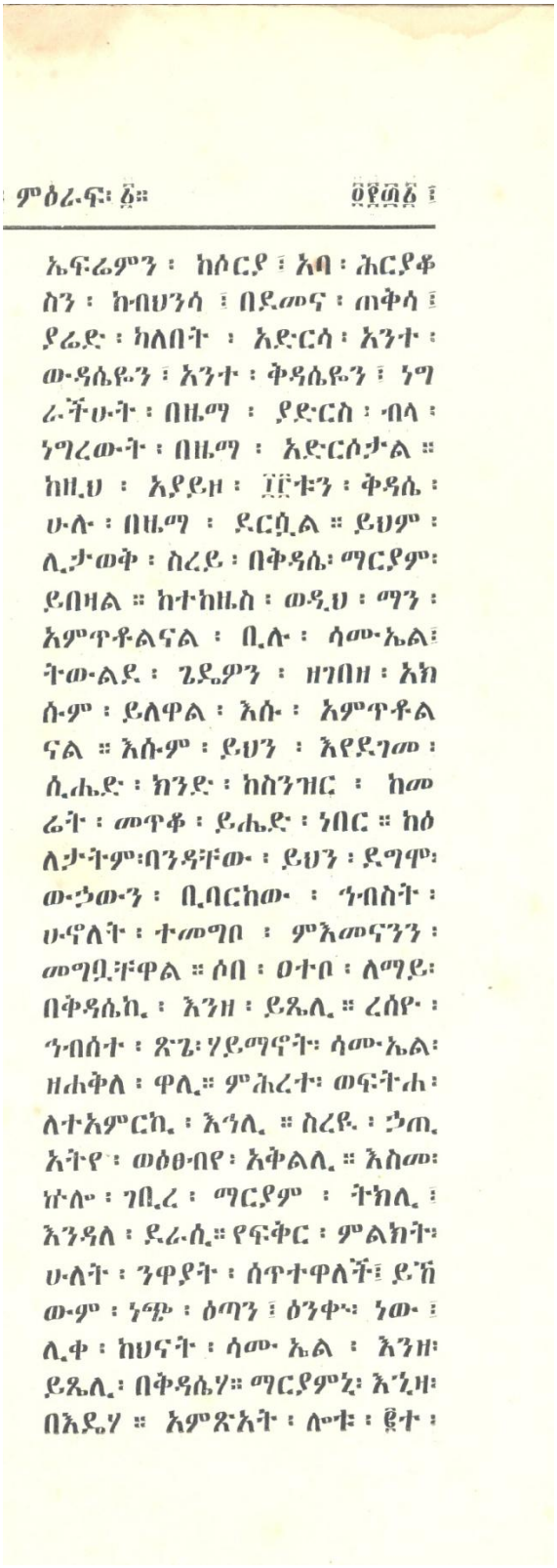


፡ ምዕራፍ ፡ ፩።

ሥጋውን ፡ በዓሕል፡ ወይኑን፡ በጽዋ፡ አድርገው ፡ ወደ ፡ መንበር ፡ የሚያ ቀርብበት፡ አኰቴት፡ የሚለውጡት፡ የሚያከብሩት ። አንድም ፤ በክብር ፤ በባለሟልነት ፡ ወደ፡ እግዚአብሔር፡ የሚያቀርብ ። አንድም ፡ ቊርባን ፤ የሚቀበሉት ፤ የሚያቀብሉት ፤ አኰቴት ፤ የሚለውጡት፡ የሚያከብሩት፡ ልመናው ፤ ክብሩ ፡ ይደርብንና፡ በብ ሀንሳ ፡ ኤጲስ ፡ ቆጶስነት ፡ የተሾመ ፡ አባ ፡ ሕርያቆስ ፡ በመንፈስ ፡ ቅዱስ ፡ ተገልጾለት ፡ የተናገረው ፡ ቅዳሴ ፡ ይህ ፡ ነው ። ጸሎታ፡ወበረከታ፡ ቢል፡ ተመስጋኝቱን ፡ ሲያይ ፤ ክብሯ፡ ልመ ናዋ ፡ ይደርብንና፡ ጸሎቱ፡ ወበረከቱ፡ ቢል ፡ አመስጋኙን ፡ ሲያይ ፡ ነው ።

፪ ፤ ጕሥዐ፡ ልብየ ፡ ቃለ ፡ ሠናየ፡ ጕሥዐ ፡ ልብየ፡ ቃለ፡ ሠናየ፡ ጕሥዐ፡ ልብየ ፡ ቃለ ፡ ሠናየ ።

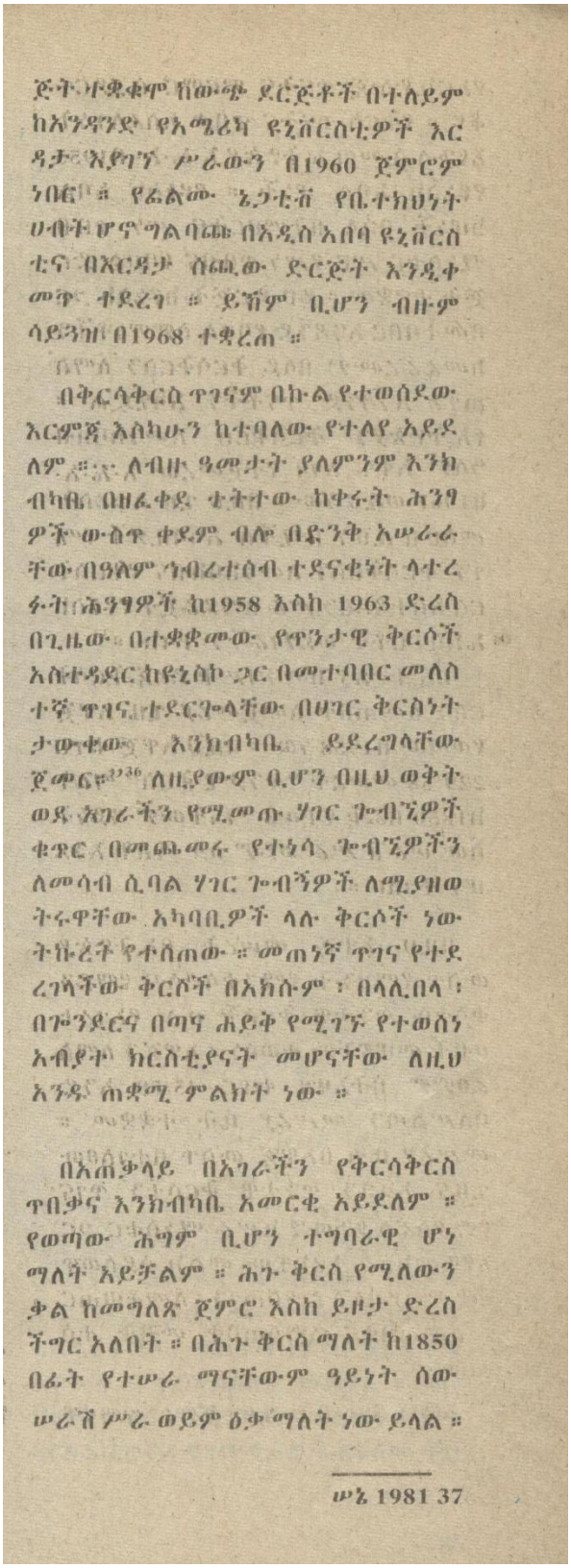
፫፡መጽሐፍ፡ከመጽሐፍ፡ማያያዝ፡ ልማድ ፡ ነው ። አባቲ ፡ ዳዊት ፡ ለን ጉሥ ፡ ሰማይ ፡ ወምድር ፡ ምስጋና ፡ ላቅርብ ፡ ባለ ፡ ጊዜ ፡ ጕሥዐ ፡ ብሎ፡ ተናግሮ ፡ ነበርና ፡እሱም፡ ለንግሥት፡ ሰማይ ፡ ወምድር፡ ምስጋና ፡ ላቅርብ፡ ባለ ፡ ጊዜ ፡ ጕሥዐ፡ አለ ። አንድም፡ ላባቲ ፡ ለዳዊት ፡ የገለጸ ፡ መንፈስ ፡ ቅዱስ ፡ ምሥጢር ፡ ከምሥጢር ፡ አያይዘለት፡ጕሥዐ ፡ አለ። ወአየድዕ፡ ይላል ፤ በቃል ፡ መደብ ፡ ቁሞ፡ ልቡ ናዬ ፡ በጎ ፡ ነገርን፡ አውጥቶ፡ ተናገረ፡ ወትጕሥዮሙ ፡ ባሕር ፡ ለኒጋድያን፡



ምዕራፍ፡ ፩። ፬፬፭፩ ፤

ኤፍሬምን ፡ ከሶርያ ፤ አባ ፡ ሕርያቆ ስን ፡ ከብህንሳ ፤ በደመና ፡ ጠቅሳ ፤ ያሬድ ፡ ካለበት ፡ አድርሳ ፡ አንተ ፡ ውዳሴዬን ፤ አንተ ፡ ቅዳሴዬን ፤ ነግ ራችሁት ፡ በዜማ ፡ ያድርስ ፡ ብላ ፡ ነግረውት ፡ በዜማ ፡ አድርሰታል ። ከዚህ ፡ አያይዘ ፡ ፲፫ቱን ፡ ቅዳሴ ፡ ሁሉ ፡ በዜማ ፡ ደርሷል ። ይህም ፡ ሊታወቅ ፡ ስረይ ፡ በቅዳሴ፡ ማርያም፡ ይበዛል ። ከተከዜስ ፡ ወዲህ ፡ ማን ፡ አምጥቶልናል ፡ ቢሉ ፡ ሳሙኤል፡ ትውልደ ፡ ጌዴዎን ፡ ዘገበዘ ፡ አክ ሱም ፡ ይለዋል ፡ እሱ ፡ አምጥቶል ናል ። እሱም ፡ ይህን ፡ እየደገመ ፡ ሲሔድ ፡ ክንድ ፡ ከስንዝር ፡ ከመ ሬት ፡ መጥቆ ፡ ይሔድ ፡ ነበር ። ከዕ ለታትም፡በንዳቸው ፡ ይህን ፡ ደግሞ፡ ውኃውን ፡ ቢባርከው ፡ ኅብስት ፡ ሁኖለት ፡ ተመግቦ ፡ ምእመናንን ፡ መግቧቸዋል ። ሶበ ፡ ወተበ ፡ ለማይ፡ በቅዳሴኪ ፡ እንዘ ፡ ይጼሊ ። ረሰዮ ፡ ኅብስተ ፡ ጽጌ፡ሃይማኖት፡ ሳሙኤል፡ ዘሐቅለ ፡ ዋሊ። ምሕረተ፡ ወፍትሐ፡ ለተአምርኪ ፡ እኅሊ ። ስረዪ ፡ ኃጢ አትየ ፡ ወዕፀብየ፡ አቅልሊ ። እስመ፡ ነሥሎ ፡ ገቢረ ፡ ማርያም ፡ ትክሊ ፡ እንዳለ ፡ ደራሲ። የፍቅር ፡ ምልክት፡ ሁለት ፡ ንዋያት ፡ ሰጥተዋለች፤ ይኸ ውም ፡ ነጭ ፡ ዕጣን ፤ ዕንቁ፡ ነው ፤ ሊቀ ፡ ከህናት ፡ ሳሙ ኤል ፡ እንዘ፡ ይጼሊ፡ በቅዳሴሃ። ማርያምኒ፡ እኂዝ፡ በእዴሃ ። አምጽአት ፡ ሎቱ ፡ ፪ት ፡

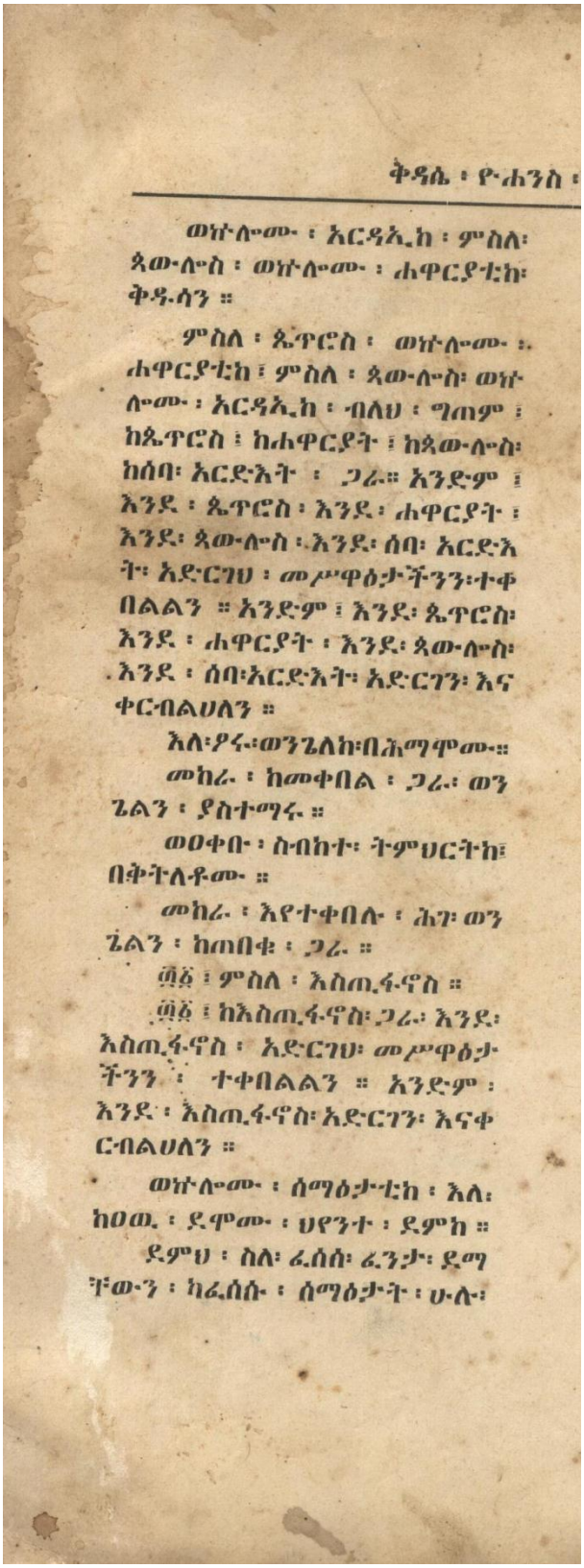
Test Set V: Sample Document Image from Book and Qidassie Yohannes (Medium Level Nose)



ጅት ሁዋቁሞ ከውጭ ደርጅቶች በተለይም ከአንዳንድ የአሜሪካ ዩኒቨርስቲዎች እርዳታ እያገኘ ሥራውን በ1960 ጀምሮም ነበር ። የፊልሙ ኔጋቲቭ የቤተክህነት ሀብት ሆኖ ግልባጩ በአዲስ አበባ ዩኒቨርስቲና በአርዳታ ሰጪው ድርጅት እንዲቀመጥ ተደረገ ። ይኸም ቢሆን ብዙም ሳይቆይ በ1968 ተቋረጠ ።

በቅርሳቅርስ ጥገናም በኩል የተወሰደው እርምጃ እስካሁን ከተባለው የተለየ አይደለም ። ለብዙ ዓመታት ያለምንም እንክብካቤ በዘፈቀደ ተትተው ከቀሩት ሕንፃዎች ውስጥ ቀደም ብሎ በድንቅ አሠራራቸው በዓለም ጎብረተሰብ ተደናቂነት ላተረፉት ሕንፃዎች ከ1958 እስከ 1963 ድረስ በጊዜው በተቋቋመው የጥንታዊ ቅርሶች አስተዳደር ከዩኒስኮ ጋር በመተባበር መለስ ተኛ ጥገና ተደርጐላቸው በሀገር ቅርስነት ታውቀው እንክብካቤ ይደረግላቸው ጀመር[30] ለዚያውም ቢሆን በዚህ ወቅት ወደ አገራችን የሚመጡ ሃገር ጐብኚዎች ቁጥር በመጨመሩ የተነሳ ጐብኚዎችን ለመሳብ ሲባል ሃገር ጐብኝዎች ለሚያዘወትሩዋቸው አካባቢዎች ላሉ ቅርሶች ነው ትኩረት የተሰጠው ። መጠነኛ ጥገና የተደረገላቸው ቅርሶች በአክሱም ፣ በላሊበላ ፣ በጐንደርና በጣና ሐይቅ የሚገኙ የተወሰኑ አብያት ክርስቲያናት መሆናቸው ለዚህ አንዱ ጠቋሚ ምልክት ነው ።

በአጠቃላይ በአገራችን የቅርሳቅርስ ጥበቃና እንክብካቤ አመርቂ አይደለም ። የወጣው ሕግም ቢሆን ተግባራዊ ሆነ ማለት አይቻልም ። ሕጉ ቅርስ የሚለውን ቃል ከመግለጽ ጀምሮ እስከ ይዞታ ድረስ ችግር አለበት ። በሕጉ ቅርስ ማለት ከ1850 በፊት የተሠራ ማናቸውም ዓይነት ሰው ሠራሽ ሥራ ወይም ዕቃ ማለት ነው ይላል ።



ወኵሎሙ ፡ አርዳኢከ ፡ ምስለ፡ ጳውሎስ ፡ ወኵሎሙ ፡ ሐዋርያቲከ፡ ቅዱሳን ።

ምስለ ፡ ጴጥሮስ ፡ ወኵሎሙ ፡ ሐዋርያቲከ ፡ ምስለ ፡ ጳውሎስ፡ ወኵሎሙ ፡ አርዳኢከ ፡ ብለህ ፡ ግጠም ፡ ከጴጥሮስ ፡ ከሐዋርያት ፡ ከጳውሎስ፡ ከሰባ፡ አርድእት ፡ ጋራ። አንድም ፡ እንደ ፡ ጴጥሮስ ፡ እንደ ፡ ሐዋርያት ፡ እንደ፡ ጳውሎስ ፡ እንደ፡ ሰባ፡ አርድእት፡ አድርገህ ፡ መሥዋዕታችንን፡ተቀበልልን ። አንድም ፡ እንደ፡ ጴጥሮስ፡ እንደ ፡ ሐዋርያት ፡ እንደ፡ ጳውሎስ፡ እንደ ፡ ሰባ፡አርድእት፡ አድርገን፡ እና ቀርብልህለን ።

እለ፡ያሩ፡ወንጌልክ፡በሕማሞሙ።

መከራ ፡ ከመቀበል ፡ ጋራ፡ ወንጌልን ፡ ያስተማሩ ።

ወዐቀቡ ፡ ስብከተ፡ ትምህርትክ፡ በቅትለቶሙ ።

መከራ ፡ እየተቀበሉ ፡ ሕገ፡ ወንጌልን ፡ ከጠበቁ ፡ ጋራ ።

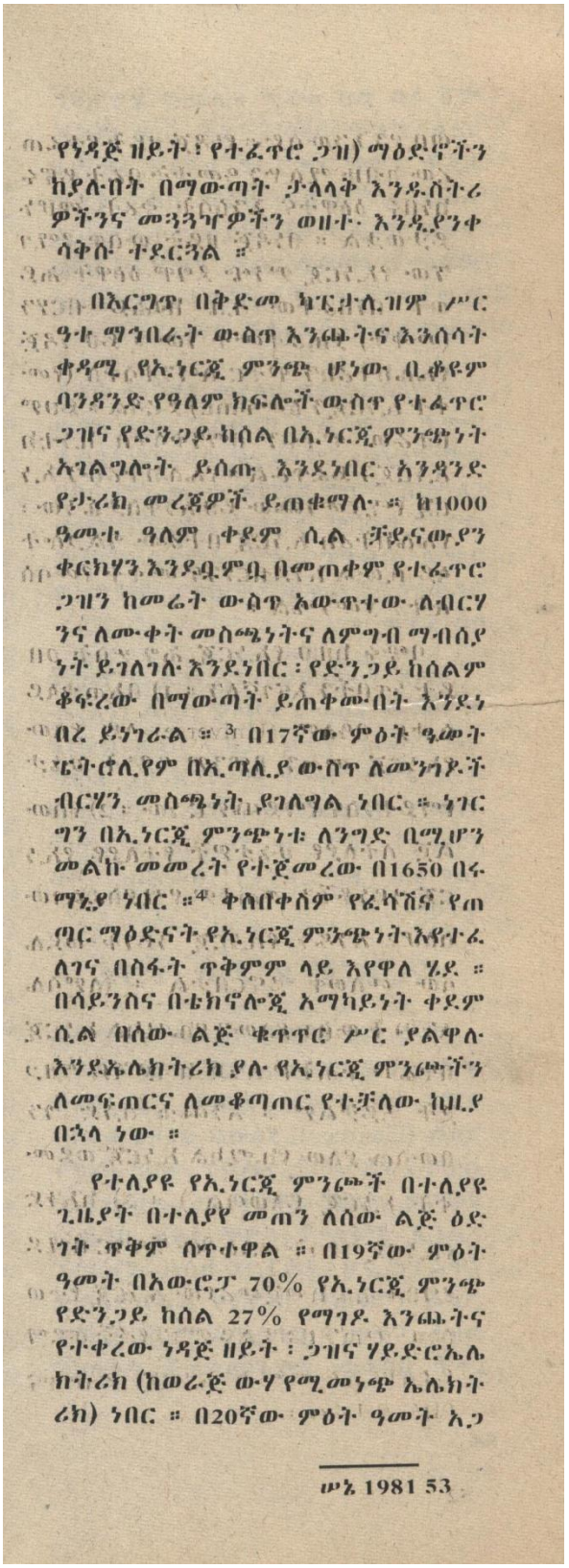
ፀ፩ ፡ ምስለ ፡ እስጢፋኖስ ።

ፀ፩ ፡ ከእስጢፋኖስ፡ጋራ፡ እንደ፡ እስጢፋኖስ ፡ አድርገህ፡ መሥዋዕታችንን ፡ ተቀበልልን ። አንድም ፡ እንደ ፡ እስጢፋኖስ፡ አድርገን፡ እናቀርብልህለን ።

ወኵሎሙ ፡ ሰማዕታቲከ ፡ እለ፡ ከዐዉ ፡ ደሞሙ ፡ ህየንተ ፡ ደምከ ።

ደምህ ፡ ስለ፡ ፈሰሰ፡ ፈንታ፡ ደማቸውን ፡ ካፈሰሱ ፡ ሰማዕታት ፡ ሁሉ፡

Test Set VI: Sample Document Image from Books (High Level Noise)



የነዳጅ ዘይት ፣ የተፈጥሮ ጋዝ) ማዕድኖችን ከያሉበት በማውጣት ታላላቅ እንዱስትሪ ዎችንና መጓጓዣዎችን ወዘተ· እንዲያንቀ ሳቅስ ተደርጓል ።

በእርግጥ በቅድመ ካፒታሊዝም ሥር ዓተ ማኅበራት ውስጥ እንጨትና እንሰሳት ቀዳሚ የኢነርጂ ምንጭ ሆነው ቢቆዩም ባንዳንድ የዓለም ክፍሎች ውስጥ የተፈጥሮ ጋዝና የድንጋይ ክሰል በኢነርጂ ምንጭነት አገልግሎት ይሰጡ እንደነበር አንዳንድ የታሪክ መረጃዎች ያጠቁማሉ· ። ክ1000 ዓመተ ዓለም ቀደም ሲል ቻይናውያን ቀርክሃን እንደቧምቧ በመጠቀም የተፈጥሮ ጋዝን ከመሬት ውስጥ አውጥተው ለብርሃ ንና ለሙቀት መስጫነትና ለምግብ ማብሰያ ነት ይገለገሉ እንደነበር ፣ የድንጋይ ከሰልም ቆፍረው በማውጣት ይጠቀሙበት እንደነ በረ ይነገራል ። [3] በ17ኛው ምዕት ዓመት ፒትሮሊየም በአሊሊያ ውስጥ ለመንገዶች ብርሃን መስጫነት ያገለግል ነበር ። ነገር ግን በኢነርጂ ምንጭነቱ ለንግድ በሚሆን መልኩ መመረት የተጀመረው በ1650 በሩ ማኒያ ነበር ።[4] ቅስበቀስም የፈሳሽና የጠ ጣር ማዕድናት የኢነርጂ ምንጭነት እየተፈ ለገና በስፋት ጥቅምም ላይ እየዋለ ሄደ ። በሳይንስና በቴክኖሎጂ አማካይነት ቀደም ሲል በሰው ልጅ ቁጥጥር ሥር ያልዋሉ እንደኤሌክትሪክ ያሉ የኢነርጂ ምንጮችን ለመፍጠርና ለመቆጣጠር የተቻለው ከዚያ በኋላ ነው ።

የተለያዩ የኢነርጂ ምንጮች በተለያዩ ጊዜያት በተለያየ መጠን ለሰው ልጅ ዕድ ገት ዋቅም ሰጥተዋል ። በ19ኛው ምዕት ዓመት በአውሮፓ 70% የኢነርጂ ምንጭ የድንጋይ ከሰል 27% የማገዶ እንጨትና የተቀረው ነዳጅ ዘይት ፣ ጋዝና ሃይድሮኤሌ ክትሪክ (ከወራጅ ውሃ የሚመነጭ ኤሌክት ሪክ) ነበር ። በ20ኛው ምዕት ዓመት አጋ

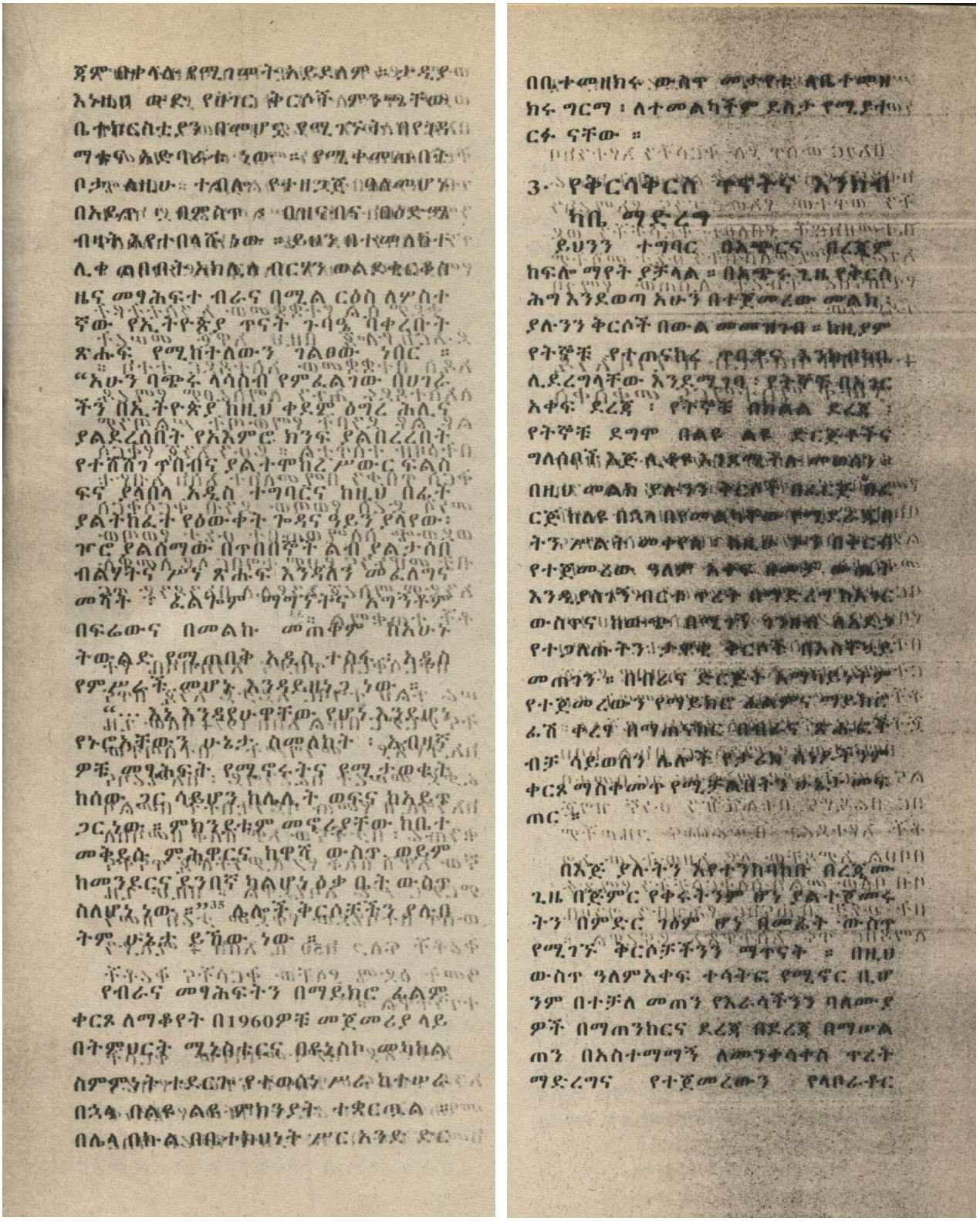
ትነት የተነሳ ከኛ አልፈው የዓለም ቅርስ ተብለው የተመዘገቡም በርካታ ቅርሶች አሉን ። እነዚህ የሰሜን ፓርክ ፣ መካከለ ኛው አዋሽ ፣ ታችኛው አሞ ፣ ጢያ ፣ አክሱም ፣ ላሊበላ ፣ የፋሲል ግቢ ሲሆኑ ፣ ከነዚህ ውስጥ በተለይ ከአንድ ወጥ ድንጋይ ተፈልፍለው የተሠሩት የላሊበላ አብያተ ክርስቲያናት በጣም ተደናቂ ናቸው ። ከውጭ አገር ሰዎች መካከል እ·ኤ·አ· ከ1521—1525 ለመጀመሪያ ጊዜ እነዚህን ከድንጋይ የተቀረፀ አብያተ ክርስቲያናት የጐበኘው አልቫሬዝ የተባለ ፖርቱጋላዊ የእነዚህን ውበት ለውጭ አገር ሰው ብን ልጽ ሊያምኑ ይችላሉ ወይ እስከማለት አድንቋቸዋል ።

አገራችን የሰርካታ ተደናቂ ቅርሳቅርስ ባለቤት መሆኗ ግልጽ ነው ። ታዲያ እነዚህ ቅርሳቅርስ በሚገባ ተጠንተው ተጠብቀ ዋል ወይ ? አሁን የሚገኙበት ሁኔታ ምን ይመስላል ? ወደፊትስ ምን እርምጃ መወ ሰድ ይኖርበታል ? የሚሉት ጥያቄዎች ምላሽ ማግኘት ያለባቸው ናቸው ። በመሆ ኑም በዚህ ክፍል ውስጥ ለእነዚህ ጥያቄ ዎች ምላሽ ለመስጠት ይሞክራል ።

የቅርሳቅርስ አጠባበቅ በኢትዮጵያ

የቅርሳቅርስ ጥበቃና እንክብካቤ በዚህን ጊዜ ተጀመረ ማለት በኢትዮጵያ ብቻ ሳይ ሆን በሌሎች የዓለም ክፍሎችም ቢሆን የሚቻል አይደለም ። ብዙውን ጊዜ ፣ እንዲያውም ሁሉ ማለት ይቻላል ፣ አገሮች ቅርሳ ቅርስ ጥበቃ በዚህን ጊዜ ጀመሩ የሚባሉት ግለሰቦች በቤተ መዘክር መልክ ቅርሳ ቅርስን አሰባስበው ማስቀመጥ የጀመ ሩበትን ፣ ወይም መንግሥት መመሪያና

Test Set VII: Sample Document Image from books (Very High Level Noise)



Annex III: MATLAB® Functions

MEAN /AVERAGE FILTERING ALGORITHM

```
function image=averagefilter(img, varargin)
     %Adopted from source www.mathworks.com
      image = rgb2gray(img);
      numvarargs = length(varargin); % Parameter checking.
      if numvarargs > 2
          error('myfuns:somefun2Alt:TooManyInputs', needs 2 optional inputs');
      end
      optargs = {[3 3] 0};            % set defaults for optional inputs
      optargs(1:numvarargs) = varargin;
      [window, padding] = optargs{:}; % use memorable variable names
      m = window(1);
      n = window(2);
      if ~mod(m,2) m = m-1; end       % check for even window sizes
      if ~mod(n,2) n = n-1; end
      if (ndims(image)~=2)            % check for color pictures
          display('The input image must be a two dimensional array.')
          display('Consider using rgb2gray or similar function.')
          return
      end
       [rows columns] = size(image);   % size of the image
      imageP  = padarray(image,[(m+1)/2 (n+1)/2], padding, 'pre');%Pad the image.
      imagePP = padarray(imageP, [(m-1)/2 (n-1)/2], padding, 'post');
      imageD = double(imagePP);
      t = cumsum(cumsum(imageD),2);
      imageI = t(1+m:rows+m, 1+n:columns+n) + t(1:rows, 1:columns)...
          - t(1+m:rows+m, 1:columns) - t(1:rows, 1+n:columns+n);
      % Now each pixel contains sum of the window but we need the average.
      imageI = imageI/(m*n);
      % Return matrix in the original type class.
      image = cast(imageI, class(image));
end
```

MEDIAN IMAGE FILTERING ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [filteredImage] = medianFilter(image,row,column)
    image = rgb2gray(image); % change image to gray-level
    filteredImage = medfilt2(image, [row column]);
end
```

WIENER IMAGE FILTERING ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [filteredImage] = wienerFilter(img,row,column)
    image = rgb2gray(image); % change image to gray-level
    filteredImage = wiener2(image, [row column]);
end
```

IMAGE QUALITY MEASURE ALGORITHM

```
%Author: Michael Abebaw (abebaw.michael@gmail.com), 2014
function [psnr,mse] = iqm(originalImage, filteredImage)
    [psnr,mse] = measerr(originalImage, filteredImage);
end
```

SAUVOLA THRESHOLDING ALGORITHM

```
%Author: Michael Abebaw (abebaw.michael@gmail.com), 2014
function output=sauvola(image, varargin)
    image = rgb2gray(image); % change image to graylevel
    % Initialization
    numvarargs = length(varargin);
    if numvarargs > 3
        error('myfuns:somefun2Alt:TooManyInputs', ...
         'Possible parameters are: (image, [m n], threshold, padding)');
    end
    optargs = {[3 3] 0.34 'replicate'}; % set defaults
    optargs(1:numvarargs) = varargin;   % use memorable variable names
    [window, k, padding] = optargs{:};
    if ndims(image) ~= 2
        error('The input image must be a two-dimensional array.'); 121
    end
    image = double(image);    % Convert to double
    mean = averagefilter(image, window, padding); % Mean value
    meanSquare = averagefilter(image.^2, window, padding); % Standard deviation
    deviation = (meanSquare - mean.^2).^0.5;
    R = max(deviation(:));% Sauvola
    threshold = mean.*(1 + k * (deviation / R-1));
    output = (image > threshold);
end
```

OTSU THRESHOLDING ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [outputImage] = otsusThreshold(inputImage)
    level = graythresh(inputImage);
    outputImage = im2bw( inputImage, level);
end
```

REMOVE FEWER PIXELS

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [cleanedImage] = removeFewerPixels(image,pixels)
    i = imcomplement(I);
    i = bwareaopen(i,pixels);
    cleanedImage = imcomplement(i);
end
```

TABLE AND OTHER LINES SEGMENTATION ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [ image ] = tableLines(image, cutoff, line_thresh)
    image = rgb2gray(image);
    bw = im2bw(image);
    BW = imcomplement(bw);
    [H,T,R] = hough(BW);
    xlabel('\theta'), ylabel('\rho');
    P = houghpeaks(H,100,'threshold',ceil(0.3*max(H(:))));
    x = T(P(:,2)); y = R(P(:,1));
    plot(x,y,'s','color','white');
    lines = houghlines(BW,T,R,P,'FillGap',1,'MinLength',cutoff); % Find lines
    set(bw, 'visible', 'off');
    figure, imagesc(bw), colormap(gray), hold on;
    [r,c,d] = size(image);
```

```
   max_len = 0;
   for k = 1:length(lines)
       xy = [lines(k).point1; lines(k).point2];
       plot(xy(1,1),xy(1,2),'x','LineWidth',3,'Color','blue'); %Plotting Begining
       plot(xy(2,1),xy(2,2),'x','LineWidth',3,'Color','green'); %Plotting end
       % Determine the endpoints of the longest line segment
       len = norm(lines(k).point1 - lines(k).point2);
       if ( len > cutoff)
         max_len = len;
         xy_long = xy;
         plot(xy_long(:,1),xy_long(:,2),'LineWidth',line_thresh,'Color','white');
         ff = getframe(gcf);
         bw = frame2im(ff);
       end
    end
    hold off
    image = im2bw(bw,0);
end
```

TEXT/GRAPHIC SEGMENTATION ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [output] = segmentImage(original_image)
    image = rgb2gray(original_image);
    bw = im2bw(image);
    se = [ 0 0 0 0;1 1 1 1;0 0 0 0];
    d = imdilate(~bw, se);
    I3 = imcomplement(~d);
    figure, imshow(original_image),title('Final Image After Segmentation');
    [Ilabel, num] = bwlabel(I3);
    Iprops = regionprops(Ilabel);
    Ibox = [Iprops.BoundingBox];
    Ibox = reshape(Ibox,[4 num]);
    hold on;
    size_info = [0 0 0; 0 0 0];
    cc = 1;
    sumHeight = 0;
    sumWidth = 0;
    sumArea = 0;
    for cnt = 1:num
        x = Ibox(:,cnt);
        component_width = x(3,:,1);
        component_height = x(4,:,1);
        size_info (cc,1) = component_width;
        size_info (cc,2) = component_height;
        size_info (cc,3) = x(3,:,1) * x(4,:,1);
        sumHeight = sumHeight + size_info (cnt,1);
        sumWidth = sumWidth + size_info (cnt,2);
        sumArea = sumArea + size_info (cnt,3);
    end
    for cnt = 1:num
        if (size_info(cnt,3) > 5000)
            rectangle('position',Ibox(:,cnt),'edgecolor','r');
            cc = cc + 1;
        end
    end
 end
```

COLUMN AND TITLE SEGMENTATION ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [output] = segmentColumn(original_image)
    image = rgb2gray(original_image);
    bw = sauvola(image, [30 30]);
    se = [0 1 1 0;0 1 1 0;0 1 1 0;0 1 1 0;1 1 1 1;0 1 1 0; 0 1 1 0;0 1 1 0;
          0 1 1 0;0 1 1 0;0 1 1 0;0 1 1 0;0 1 1 0;0 1 1 0;0 1 1 0;0 1 1 0];
    d = imdilate(~bw, se);
    d = bwdist(~d) >= 1;
    figure, imshow(original_image);
    [Ilabel, num] = bwlabel(d);
    Iprops = regionprops(Ilabel);
    Ibox = [Iprops.BoundingBox];
    Ibox = reshape(Ibox,[4 num]);
    size_info = [0 0 0; 0 0 0], position_info = [0 0 0 0];
    sumArea = 0;
    for cnt = 1:num
        x = Ibox(:,cnt);
        component_width = x(3,:,1);
        component_height = x(4,:,1);
        component_area = component_width * component_height;
        size_info (cnt,1) = component_width;
        size_info (cnt,2) = component_height;
        size_info (cnt,3) = component_area;
        sumArea = sumArea + component_area;
    end
    count = 1;
    cc = 1;
    c = 1;
    maxArea = max(size_info);
    for cnt = 1:num
        x = Ibox(:,cnt);
        if size_info (cnt,2) > maxArea(1,2)/4 && size_info...
                        (cnt,1) > maxArea(1,1)/4
            position_info(c,1) = x(1,:,1);
            position_info(c,2) = x(2,:,1);
            position_info(c,3) = x(3,:,1);
            position_info(c,4) = x(4,:,1);
            c = c + 1;
        end
    end
    len = length(position_info);
    minY = min(position_info(:,2));
    for cnt = 1:num
        if size_info (cnt,2) > maxArea(1,2)/4 && size_info...
                        (cnt,1) > maxArea(1,1)/4
            rectangle('position',Ibox(:,cnt),'edgecolor','r');
        else
            if (size_info (cnt,3) < maxArea(1,3)/2 &&...
                                    size_info (cnt,3) > 1000)
                if x(2,:,1) < minY
                    rectangle('position',Ibox(:,cnt),'edgecolor','b'); end
            end
        end
    end
    output = original_image;
end
```

MORPHOLOGICAL DILATION BASED LINE SEGMENTATION ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [ segmentedLine ] = lineSegmentation(Image,saveLocation,startFileName)
    image = Image;
    se = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
          0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
    bw = image;
    d = imdilate(~bw, se);
    I3 = d;
    figure, imshow(image);
    [Ilabel, num] = bwlabel(I3);
    Iprops = regionprops(Ilabel);
    Ibox = [Iprops.BoundingBox];
    Ibox = reshape(Ibox,[4 num]);
    hold on;
    heightT = 0;
    for cnt = 1:num
        heightT = heightT + Ibox(4,cnt);
    end
    heightT = heightT/(num+4);
    for cnt = 1:num
        if(Ibox(4,cnt) > heightT)
            rectangle('position',Ibox(:,cnt),'edgecolor','r');
        end
    end
end
```

MORPHOLOGICAL DILATION BASED WORD SEGMENTATION ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [segmentedWord] = segmentWord(Image, saveLocation, startFileName)
    dialationThresh = 5;
    bw = im2bw(image);
    dialatedIm = bwdist(~bw) >= dialationThresh;
    %z = removeFewerPixels(dialatedIm, 2000);
    I3 = imcomplement(dialatedIm );
    f = figure, imshow(Image);
    set(f, 'visible', 'off');
    [Ilabel, num] = bwlabel(I3);
    Iprops = regionprops(Ilabel);
    Ibox = [Iprops.BoundingBox];
    Ibox = reshape(Ibox,[4 num]);
    hold on;
    for cnt = 1:num
        rectangle('position',Ibox(:,cnt),'edgecolor','r');
    end
end
```

MORPHOLOGICAL DILATION (VERTICAL) ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [dilatedImage] = dialate(image)
    se = [0 0 1 0 0; 0 0 1 0 0; 0 0 1 0 0; 0 0 1 0 0; 0 0 1 0 0];
    d = imdilate(~image, se);
    dilatedImage = bwdist(d) >= 1;
end
```

FILTERING INK-BLEEDINNG FOR LINE SEGMENTATION ALGORITHM

```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function [ output ] = inkBleedingFilter(image)
    grayImage = rgb2gray(image);
    wein = wienerFilter(grayImage, 10, 10);
    sau = sauvola(wein, [25 25]);
    d = bwdist(~sau) >= 1;
    output = d;
end
```

A Sample of CC BASED CHARACTER SEGMENTATION ALGORITHM

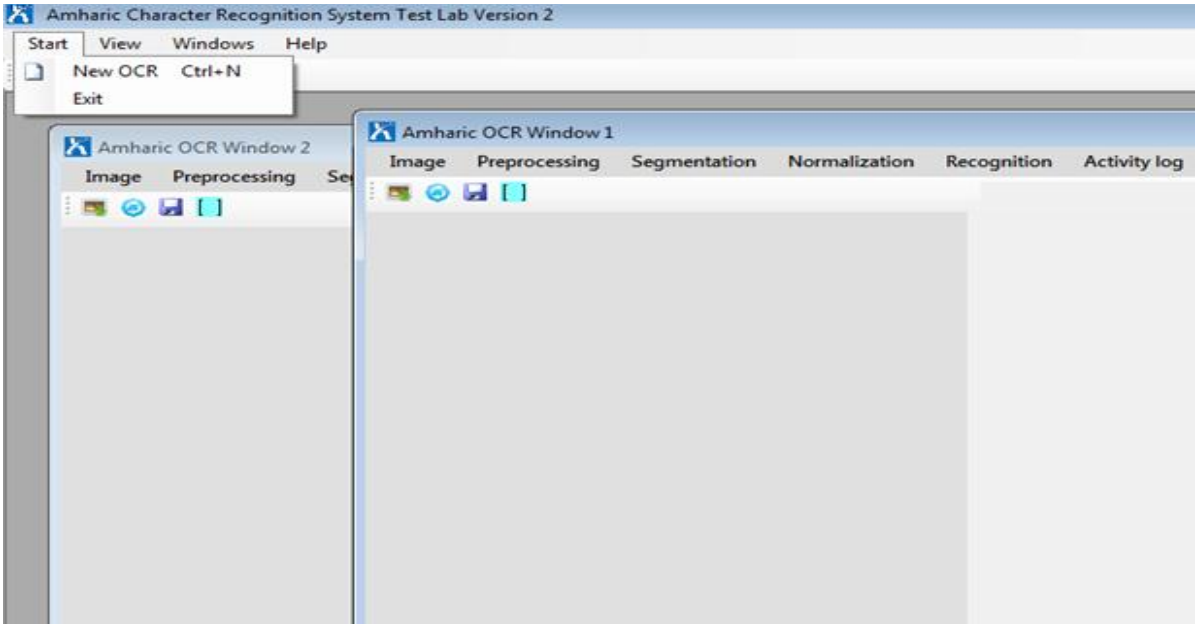
```
%Author: Berhanu Sahle (bresh19@gmail.com), 2015
function newCharSeg(Image,saveLocation, startFileName)
    size_info = [];
    f = figure, imshow(image);
    set(f, 'visible', 'off');
    [cc, num] = bwlabel(~image);
    Iprops = regionprops(cc);
    Ibox = [Iprops.BoundingBox];
    Ibox = reshape(Ibox,[4 num]);
    for cnt = 1:num
        x = Ibox(:,cnt);
        size_info(cnt,1) = cnt;
        size_info(cnt,2) = x(1,:,1);
        size_info(cnt,3) = x(1,:,1) +  x(3,:,1);
        size_info(cnt,4) = x(2,:,1);
        size_info(cnt,5) = x(2,:,1) + x(4,:,1);
        size_info(cnt,6) = x(3,:,1);
        size_info(cnt,7) = x(4,:,1);
        size_info(cnt,8) = x(3,:,1) * x(4,:,1);
        size_info(cnt,9) = (x(1,:,1) + (x(1,:,1) +  x(3,:,1)))/2;
        size_info(cnt,10) = x(3,:,1) / x(4,:,1);
        size_info(cnt,11) = x(4,:,1) / x(3,:,1);
    end
    len = length(size_info(:,1));
    hei = height/2;
    maxH = max(size_info(:,7));
    minH = min(size_info(:,7));
    maxW = max(size_info(:,6));
    minW = min(size_info(:,6));
    cc = 1;

    for cnt = 1:len
        if cnt == len && i == 4
            filename=strcat(saveLocation,'\',
            startFileName,'_char',num2str(cc),'s_.bmp');
        else
            filename=strcat(saveLocation,'\',
            startFileName,'_char',num2str(cc),'_.bmp');
        end
        if hei >=size_info(cnt,4) && hei <= size_info(cnt,5)
            if size_info(cnt,6) > size_info(cnt,7)
```

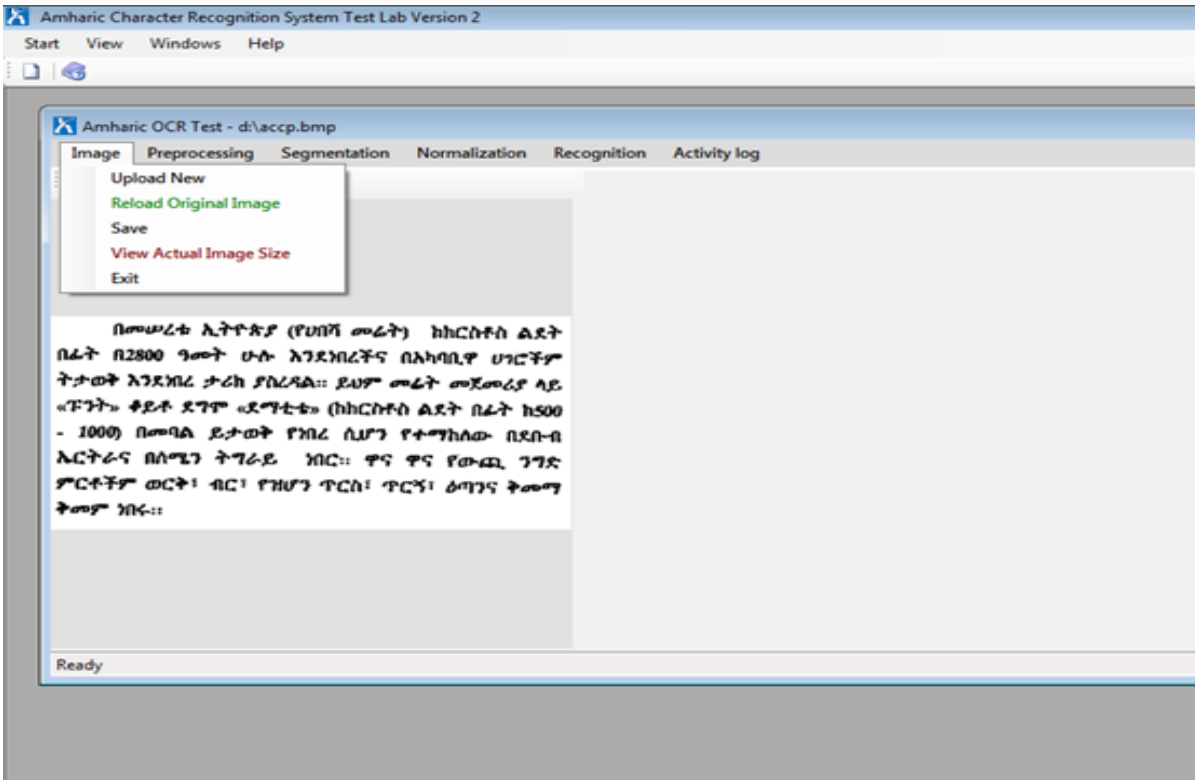
```
                if(((size_info(cnt,6)-minW)/10 >= 0.5))
                    if size_info(cnt,6)/size_info(cnt,7) >= 2.5
                        if size_info(cnt,6)/size_info(cnt,7) > 2.6
                            if size_info(cnt,6)/size_info(cnt,7) >= 3
                                if size_info(cnt,6)/size_info(cnt,7) >= 4
                                    %Crop the image into five parts
                                else
                                    %Crop the image into four parts
                                end
                            else
                               %Crop the image into three parts
                            end
                        else
                           %Crop the image into two parts
                        end
                    else
                        if(maxH - size_info(cnt,7) <= 3)
                            if (size_info(cnt,10) >= 1.18 &&
                                    size_info(cnt,11) >= 0.45)
                                    subImage = imcrop(image, [Ibox(1,cnt)
                                    Ibox(2,cnt)/(hei*2) Ibox(3,cnt)/2
                                    (size_info(cnt,7)+(hei*2))]);
                                    subImage2 = imcrop(image, [Ibox(1,cnt)+
                                    Ibox(3,cnt)/2 Ibox(2,cnt)/(hei*2)
                                    Ibox(3,cnt)/2 (size_info(cnt,7)+(hei*2))]);
                                    [cc2, num2] = bwlabel(~subImage);
                                    Iprops2 = regionprops(cc2);
                                    Ibox2 = [Iprops2.BoundingBox];
                                    Ibox2 = reshape(Ibox2,[4 num2]);
                                    [cc3, num3] = bwlabel(~subImage2);
                                    Iprops3 = regionprops(cc3);
                                    Ibox3 = [Iprops3.BoundingBox];
                                    Ibox3 = reshape(Ibox3,[4 num3]);
                                    y = Ibox2(:,1);
                                    z = Ibox3(:,1);
                                    if (((y(3,:,1) * y(4,:,1)) >= ((size_info(cnt,8)
                                            /4) + 20)) && ((z(3,:,1) * z(4,:,1)) >=
                                            ((size_info(cnt,8)/4)) + 20))
                                        %Crop the image into two parts
                                    end
                            end
                        end
                    end
                end
            end
        end
    end
end
```

Annex V: Sample Experimental Visual C# User Interface

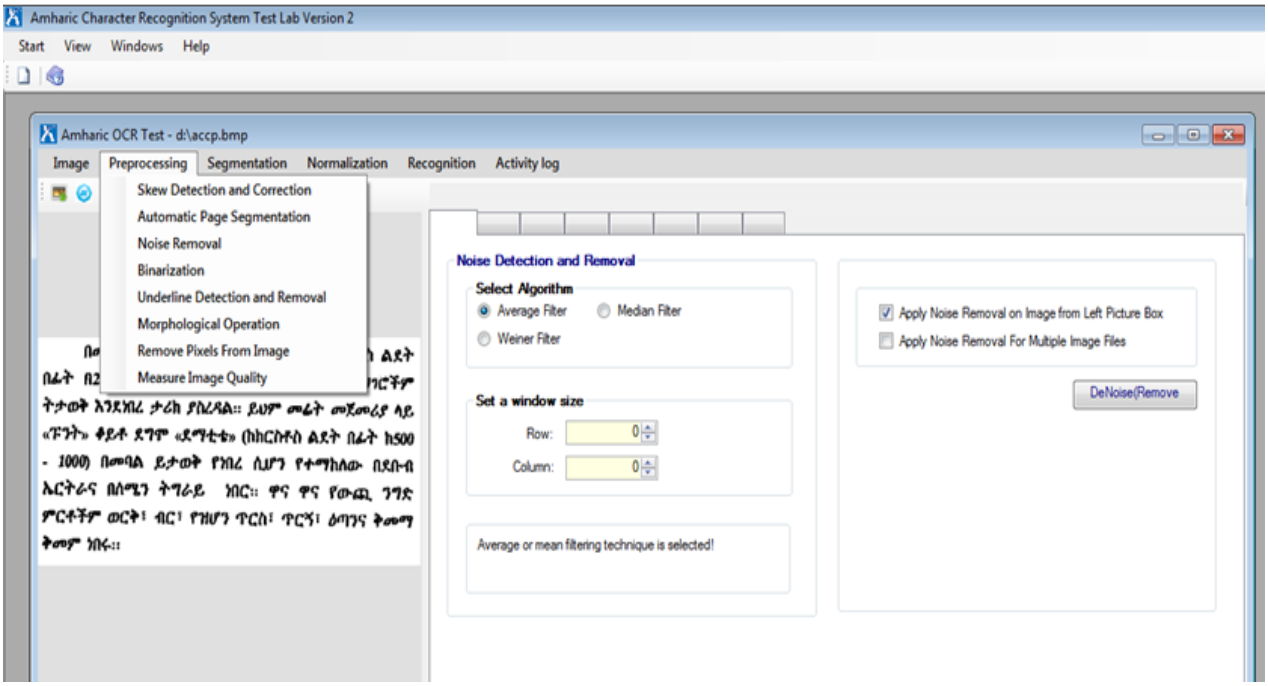
- ***Parent and Child Forms***



- ***Uploading Document Image***



- ***Image Preprocessing Menus and Screen***



- ***Sample Line segmentation Screenshot***