Addis Ababa University
College of Natural Sciences
School of Information Science

Page Column and Paragraph Layouts Segmentation and Reconstruction for Recognizing Real Life Documents

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

Andualem Ayedagne

Advisor

Million Meshesha (Ph.D)

June 2016

Addis Ababa University
College of Natural Sciences
School of Information Science

Page Column and Paragraph Layouts Segmentation and Reconstruction for Recognizing Real Life Documents

By

Andualem Ayedagne

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 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.

Andualem Ayedagne

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

Million Meshesha (Ph.D)

DEDICATION

To my Mom (Almaz Abera),

My Dad (Ayedagn Meren) and

My beloved sister (Azeb Ayedagn)

Who sacrifices a lot for everything I have

ACKNOWLEDGEMENTS

First of all, I praise the almighty GOD and his mother ST. MARY for making everything possible. Thank you for giving me the strength, courage and the opportunity to complete the masters program at AAU.

I would also like to express my deepest gratitude to my advisor Dr. Million Meshesha for being on my side from the start to the end. His sage advice, encouragement, guidance and support enabled me to develop an understanding of the subject. Besides his expertise, I really appreciate for his concern and perspective advice throughout my thesis work. This accomplishment would not have been possible without him. Thank You So Much!

Another special thanks goes to Mr. Berhanu Sahle who had graduated in 2015 from AAU. His constructive ideas, invaluable support and materials that I have looked-up aided the writing of this thesis in innumerable ways. Thank you buddy!

I would like to express my very profound gratitude for my loving family: mom, dad, my sisters (Tigist and Azeb) and my brothers (Solomon and Elias). Thank you all for your encouragement, love and support you gave me all those years. Especially, Azeb Ayedagn I don't know if there are any words to express how grateful I am to have you.. You mean the world to me. Thank you so much for everything you have done. This is the fruit of your effort.

Last but not least, my utmost gratitude is extended to my best friends Mihiret A., Bezaye B., Teddy A., Dawit B., Wondmagegn M., and to all DARO IT staffs whose love, concern, friendship and ideas have supported me in one way or the other in the preparation and completion of the study. Especially, Teddy A. thanks for your constructive and helpful comments and most importantly for your contribution in the execution of the class.

Andualem A

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABBREVIATIONS

ANN	Artificial Neural Network
ASCII	American Standard Code for Information Interchange
CPU	Central Processing Unit
CC	Connected Component
DAG	Decision directed acyclic graph
DIR	Document Image Retrieval
Dpi	Dots per Inch
GB	Giga Bytes
GHz	Giga Hertz
HP	Hewlett-Packard
KFDA	Kernel Fisher Discriminant Analysis
K-NN	K-Nearest Neighbors
KPCA	Kernel Principal Component Analysis
LDA	Linear discriminant analysis
MATLAB	MATrix LABoratory
OCR	Optical Character Recognition
PCA	Principal Component Analysis
RAM	Random Access Memory
RGB	Red Green Blue
RXYC	Recursive x-y cuts
SVM	Support Vector Machines
US	United State
2D	Two Dimensional

ABSTRACT

Nowadays a huge amount of handwritten, typewritten and printed documents contain valuable information and knowledge that still recorded, stored, and distributed in paper format. To make the information and knowledge embedded in these documents accessible and easily reachable, it is required to digitize and organize them. In the course of digitization, Optical Character Recognition (OCR) plays a vital role, since it simplifies the process of converting scanned images of text into editable digital documents, while preserving both the content and the format of documents. Different researchers explore various issues on the course of developing Amharic OCR. Most of previously conducted researches focus on character (text) recognition of the script. However, Real-life document images usually contain not only characters (text) but also some associated non text elements (graphics, column, paragraph etc.). Consequently, detecting and reconstructing non-text elements of a document image during the digitization process are important for the purpose of reusing documents.

This study applies dilation, connected component (CC) analysis, CC width, height and area analysis and a novel modified whitespace analysis page segmentation algorithm to separate graphics from text; to detect column and paragraph block and also to collect information of those layouts with the aim of reconstructing the original document image column and paragraph layouts. Based on the stored layout information, the proposed system maintains a column block 80% and paragraph block 72.22%. The performance of column and paragraph layouts reconstruction heavily depends on page segmentation stage. It reconstructs column and paragraph layouts with the efficiency of 100 % for correctly segmented column and paragraph blocks.

Maintaining original document image layout in character recognition is important to produce well-structured recognized text. However, the developed column and paragraph layouts segmentation and reconstruction techniques fails to reconstruct column blocks based on the width size of the original document image, and to segment paragraph blocks when every lines in the paragraph have equal end points. Thus, there is a need to explore on adaptive page segmentation techniques, and on preservation of width variant column blocks.

CHAPTER ONE
INTRODUCTION

1.1. Background

The drastic introduction of modern computers into every area of life has radically led to a paradigm shift in the way people trade, communicate, learn, share knowledge, and get entertained. Nowadays computers are electronic and digital, and thus they can only process data in digital format. Given that, anything that requires a computer processing must first be transformed into a digital form [1]. However, most information is still recorded, stored, and distributed in paper format. The widespread use of computers only had the effect of increasing the amount of information held on paper, instead of reducing [2]. Thus, large number of printed documents such as letters, newspapers, magazines and books and even old manuscripts are available in governmental, religious and private institutes, libraries and museums having different writing styles. Hence, enabling such rich information items to their digital format for effective access, reliable storage, computability and long term preservation is very important. This makes Optical Character Recognition (OCR) an active research area of information science [3][4][5].

OCR is the automated process of translating an input document image (which can be printed, typewritten and handwritten document) into a symbolic text file. Thus, documents can digitally stored and accessible for further use efficiently [6]. It involves image processing, pattern recognition, natural language processing, artificial intelligence, and database systems. It can be applied in 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 many other applications [10].

In general, there are two types of optical character recognition systems [7][8] [9]. The first type is the offline OCR system which extracts data from scanned images through optical scanners and cameras. The second type is the online OCR system which employs special digitizers to capture in real-time the user's writing according to the order of the lettering, speed, and pen movements and strokes.

Technically speaking, every OCR system undergoes a process of sequential stages in order to convert the input scanned document, may be a printed, handwritten, or typewritten text, into a computer digital text. This process starts with image acquisition stage which captures the input document in image format. The document image is passes through the pre-processing stage to improve quality and to removes artifacts from the input document. The pre-processing step includes skew detection and correction, noise detection and removal, binarization, thinning and normalization. [3] [10].

Segmentation then determines the elements of an image. It is an operation that seeks to decompose an image into sub-images of individual symbols. Basically, segmentation is classified into page and text segmentation. Page segmentation is a process of dividing a document images into homogeneous blocks i.e. graphics, columns, tables, etc. Whereas, text segmentation is a process of segmenting text blocks in a document image into line, word and characters [33] [10].

Feature extraction and classification is followed in the sequential steps of an OCR. Feature extraction represents character image and classification module label character to their proper class. Training and testing are the two basic phases of any pattern classification. During training phase, the classifier learns the association between samples and their labels from labeled

samples. The testing phase involves analysis of errors in the classification of unlabelled samples. This is an essential stage for character recognition process. After recognition page layout reconstruction is applied to maintain the original documents page layout. Finally, post-processing stage refines the OCR output text by correcting linguistic misspellings [7] [10].

Optical character recognition technology was invented in the early 1800s, when it was patented as reading aids for the blind. In 1870, C. R. Carey patented an image transmission system using photocells, and in 1890 P.G. Nipkow invented sequential scanning OCR [31]. Investigation into the techniques of OCR started relatively early in the field of pattern recognition. It dates back almost to that of the history of computer. The concept was introduced and got recognition after Taushbeck and Handel obtained a patent on OCR in 1929 in Germany and in 1933 in the US respectively [11]. However, the practical OCR technology used for reading characters was introduced in the early 1950s as a replacement for keypunching system. A year later, D.H. Shephard developed the first commercial OCR for typewritten data. The 1980's saw the emergence of OCR systems intended for use with personal computers [10].

Nowadays, due to less expensive electronic components, and extensive researches in the area, development of OCR systems for Latin-script is well advanced to the level of using for practical problem-solving by integrating languages in one package. One example is Tesseract which is currently developed by Google™, it can process English, French, Italian, German, Portuguese, Spanish and Dutch Scripts. However, there is limited research effort and low recognition rate achieved in this direction for the indigenous scripts of African languages in general Amharic language in particular [12] [13] [5].

1.2. Statement of the Problem and Justification

Amharic is one of the languages in Africa which has its own indigenous scripts and writing systems. It is the second most spoken Semitic language in the world after Arabic and became the dominant language in history of Ethiopia. It started to be used since around 14th century and starting from 19th century, Amharic symbols begun to be used for writing purposes [14] [15] [12]. Due to such a long history, a bulk of documents (such as correspondence letters, newspapers, magazines, and books) available inside churches, caves, governmental and private institutions including information centers, libraries, museums, etc. in different formats[16].

These huge amounts of handwritten, typewritten and printed documents contain numerous information and knowledge of different areas. To make the information and knowledge embedded in these documents accessible and easily reachable, it is desirable to convert such kind of documents into their equivalent digital format [17].In the course of digitization, OCR plays a vital role, since it is a process of converting scanned images of text into editable digital documents, while preserving both the content and the format of documents. Hence, it can be processed, edited, searched, saved, and copied for an unlimited number of times without any degradation or loss of information using computing technology[7][18].

Since the first effort made by Worku [19] in 1997, different researchers explore various issues on the course of developing Amharic OCR. Most of previously conducted researches focus on character (text) recognition of the script. However, Real-life document images usually contain not only characters (text) but also some associated non text elements (graphics, column, paragraph etc.). Thus, detecting and reconstructing non-text elements of an image during the digitization process are important for the purpose of reusing (utilizing) documents. Since the

main concern of those studies were maintaining the text part of an image document, page segmentation stage of an OCR were only used in order to separate text from non-text area and to remove the non-text part of the original document image. As a result, for document image which have multiple columns, can only manage to produce a single column plain text and the recognized text outputs are unstructured, every lines of text block considered as a paragraph.

Hence, this study explores on various page segmentation techniques for identifying column and paragraph blocks, for recognizing texts in them and reconstructs column and paragraph blocks with the text into their original form. In an effort to solve the above stated problem this research address the following research questions:

- What kind of page column and paragraph layout detection and reconstruction techniques is fitting in order to maintain the original document image column and paragraph layouts?
- Which page segmentation technique is effective for identifying column and paragraph block in real life document images?
- To what extent the performance of Amharic OCR system improves after applying the proposed page column and paragraph layout segmentation and reconstruction techniques?

1.3. Objectives of the Study

1.3.1. General Objective

The general objective of this research is to apply an effective page segmentation technique that can identify column and paragraph blocks with the aim of reconstructing document image column and paragraph layouts to increase readability, usability and accuracy of recognized real life Amharic documents.

1.3.2. Specific Objectives

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

- To review different researches on page layout segmentation and reconstruction to understand the area, approaches, and algorithms
- To identify different layouts of real life document images.
- To explore and select potential page segmentation techniques for identifying column and paragraph layouts of real life document images;
- To prepare a document image corpus to measure the performance of the proposed system;
- To design a technique for page column and paragraph segmentation and reconstruction after recognition
- To integrate with previously developed Amharic OCR systems and measure the performance of the proposed system

1.4. Scope and limitation of the study

On the course of developing an improved Amharic OCR system for real life documents, this study experiment some available page column and paragraph layouts segmentation and reconstruction techniques for multiple columns in Amharic document images. Among many available algorithms, preferred techniques study and test in real-life document images by integrating with pervious works. The performance of the proposed page column and paragraph segmentation and reconstruction technique is measured by sample scanned documents from newspapers and magazines.

As mentioned in the previous sections an effective page segmentation technique is important to preserve original document image page layout. Real-life document image have different physical and logical page layouts. However, this study only focuses on text/graphics separation, column and paragraph block detection and reconstruction. Due to time limitation graphics block of a document image is not reconstructed. Segmentation and reconstruction of other document image page layouts such as table, header, footer, etc are also not included. Handwritten documents, vertical text detection and recognition are out of the scope of this study. Preprocessing, text segmentation and recognition stage of an OCR adopted from the previous studies with slight modification.

1.5. Methodology of the Study

Methodology provides a way to achieve the objectives of a research problem. Literatures, such as books, journal articles, conference proceedings and the Internet about OCR in general and page layout segmentation and reconstruction in particular have been intensively reviewed in order to acquire detailed understanding of the subject matter and the research areas. Also the past and present research works on Amharic and other languages reviewed to have a better background on the best performing algorithms and techniques regarding column and paragraph blocks segmentation and reconstruction. Since this research is supposed to be a continuation of the previous researches and need to be integrated with them, local researches on Amharic OCR are given more emphasis. Therefore, in order to undertake and achieve the objective of this research the following methods and techniques are used.

1.5.1. Study design

This research follows an experimental research, which uses manipulation and controlled testing to understand causal processes. This type of research will come up with conclusions which are capable of being verified by observation or experiment [7]. Following experimental research dataset preparation, techniques identification, system design, system development and evaluation are the procedures that this study follows.

1.5.2. Dataset Collection

Documents that contain graphics, column and paragraph have been collected from various sources to measure the performance of the proposed system. The datasets are collected from newspapers and magazines for training and testing to understand the impact of page column and paragraph layout segmentation and reconstruction techniques in optical character recognition. The total number of dataset collected is 50; among those dataset 16 of them are taken from Berhanu [24] dataset. The remaining 34 document images are newly added.

1.5.3. Implementation tools

Berhanu [24] and Michael [4] used MATLAB® Image Processing and Visual C# in earlier works. Thus, Visual C# libraries and MATLAB® Image Processing Toolbox™ are also integrated and used in this research. Because, the researcher is familiar with this languages and MATLAB is better for the image processing due to availability of its rich libraries. It also simplifies integration of the proposed approach with the previous study. HP Scanjet 8200 scanner is used to scan newly added images of real life document. The documents are scanned in grayscale having a resolution of 300 dpi.

1.5.4. Performance Evaluation

The performance of the system is tested at various stages. Since, this study focuses on page column and paragraph layouts segmentation and reconstruction, the performance of the proposed system is measured by direct mapping, which determines the performance of layout segmentation and reconstruction by finding the correspondences between detected entities and ground truth [60]. For page column and paragraph segmentation, it counts the expected correct segmentation vis-à-vis erred segmentation made by the proposed page segmentation and calculates the segmentation accuracy percentage. The expected correct segmentation represents the expected number segmented block. Similar procedures are followed to calculate accuracy percentage for page column and paragraph layout reconstructions.

1.6. Significance of the research

Large amount of real-life and historical Amharic documents articulated in printed, typewritten and handwritten formats and are available in information centers, libraries, museums, governmental religious and private institutes [8]. Manual conversion of these documents is very tedious, labor intensive, error prone and time consuming. Therefore, OCR systems can provide an automatic transformation into computer representation of these documents without the need of typing. So that, it enables valuable printed, typewritten and handwritten documents electronically available, portable and accessible for future reference with only small memory requirement [18]. It also provides tremendous opportunity in handling repetitive task like postal mail sorting according to destination address, bank check processing, bill processing and so forth.

This research in particular plays its own role in the attempt of developing fully integrated Amharic Character Recognition System, which is not developed yet. There are different research works on the area that explored to improve recognition system on different sequential stages of an OCR. But, there is no any page layout reconstruction techniques applied after recognition. So that, this research focuses on page column and paragraph layouts segmentation and reconstruction of an OCR to advance the effort made in the area.

1.7. Organization of the study

The thesis is organized in to five chapters. The first chapter of the thesis discusses the background, the statement of the problem and its justification. It also includes objectives, scope and significance of the study. The chapter also presents the methodology used to accomplish the objectives.

The second chapter is literature review. It provides an overview of OCR system and phases conducted under OCR systems. Moreover, a brief review on document image page layout segmentation, local related works on document image recognition, Amharic writing system and the challenges in building Amharic OCR.

In chapter three, selected page column and paragraph layout detection and reconstruction techniques are reviewed and explained. The evaluation matrix that is used for measuring the performance of each algorithm discussed.

Chapter four deals with details of experimentation on selected page layout detection and reconstruction techniques, and experimental results used to confirm the validity of the proposed techniques are presented. Based on the results of the experiment, the last chapter presents the conclusion on the findings of the study and forward recommendations for further research works in the area of page layout segmentation and reconstruction.

CHAPTER TWO
LITERATURE REVIEW

The development of OCR is highly motivated by the availability of large amount of printed, typewritten and hand written documents in information centers, libraries, museums and government and private institutes [10].Manually accessing these bulks of real-life documents is time consuming. It is also costly to copy and make the documents available for large number of people. Thus, these documents need to be digitized and easily accessible via the Internet and digital libraries [13]. The purpose of automatic recognition of texts is to convert texts stored in a paper or other media to a standard encoding scheme like ASCII or Unicode representing the texts to the effect that efficient automatic services can be provided [11]. Reviewing different researches on OCR system in general and page layout detection and reconstruction in particular is important for this research.

2.1. Overview of OCR System

In the 21st century of our world, information in digital form plays a significant role in our daily life. This is because anyone from anywhere can access them easily [19]. There are lots of handwritten, typewritten and printed documents which need to be introduced to the digital world and be accessed by anyone. Therefore, there should be a means of digitizing these documents. OCR is automatic reading of optically sensed document texts of human-readable characters to machine-readable codes [9].The concept was introduced and got recognition after Taushbeck and Handel obtained a patent on OCR in 1929 in Germany and in 1933 in the US respectively. However, the practical OCR technology used for reading characters was introduced in the early 1950s as a replacement for keypunching system. Since then it becomes very active field of

research and now it is one of the most successful applications of automatic pattern recognition [25][10][11].Due to less expensive electronic components, and extensive researches in the area, typical accuracy rates of OCR systems for Latin-script has exceeded 99%, although certain applications demand even higher accuracies. Open systems such as Tesseract (which is currently developed by Google™) is an example of this which can process English, French, Italian, German, Spanish, Brazilian, Portuguese and Dutch scripts [16].

2.2. The major steps in OCR

To perform the recognition process, OCR system undergoes through a process of sequential stages in order to convert a paper text document into a computer digital text. Different literatures classified the steps in different ways (see figure 2.1). The process consists of image acquisition, image pre-processing, segmentation, feature extraction, classification and finally post processing [12]. The general overview of OCR is presented in figure 2.1 below.

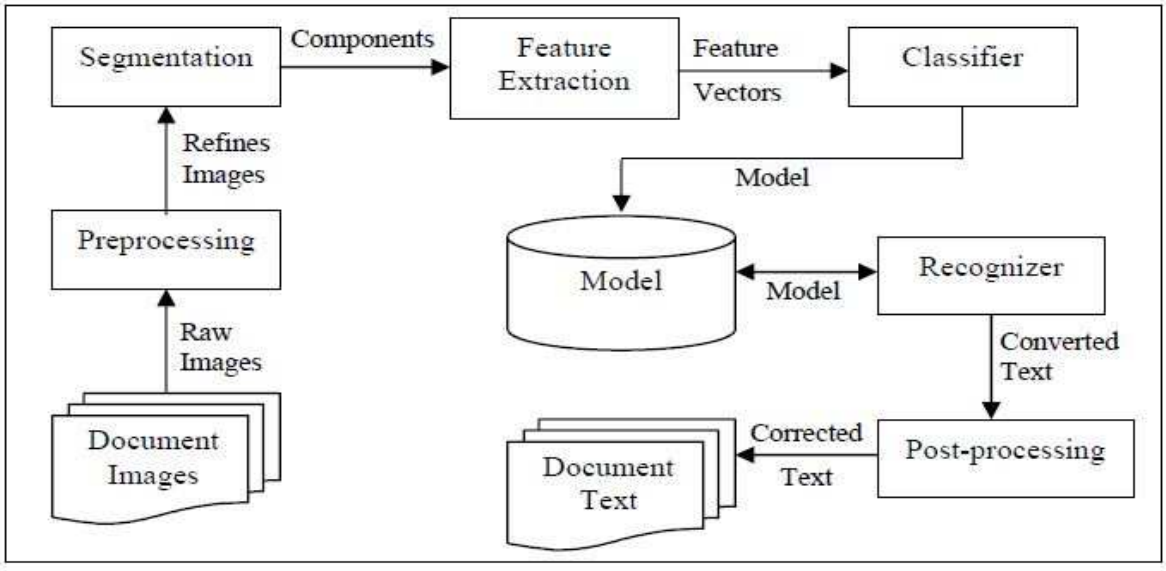


Figure 2.1: General overview of Amharic OCR System adopted from [10]

Image acquisition (document image scanning): is the first step in character recognition. It is a process of capturing real life document images through a scanner, camera or through on-screen pen up and pen down information [26][16].Michael [4] and Dereje [20] explained that most OCR systems use are solution that range between 300 dpi to 1000 dpi for better accuracy in text extraction. For a better recognition, studies show that using high resolution is necessary while scanning real world historic documents which are very noisy and degraded [4][20].

Image preprocessing: This is most important and crucial step that is very helpful to enhance the performance of recognition. It is a course of correcting the deficiencies in data acquisition process due to a lack of paper quality, resolution of the scanned image, the amount of skew in the image, the format and layout of the images and text and so on. Major preprocessing tasks while working with document images are noise reduction, binarization, skew correction, underline removal thinning, smoothing and normalization which are an important approach to prepare data for subsequent activities like segmentation and classification stage [4][5].

Preprocessing is helpful to reduce noises, degradation and inconsistencies over the document image. Noise is the random variation of brightness or color information in images produced by the sensor and circuitry of a scanner or digital camera [27]. Removing any type of noise from document images is one of the preprocessing tasks. Low-pass filtering techniques like: mean filter, median filter, adaptive median filter, etc are the most common preprocessing techniques that are used by OCR systems [28].

Document images have foreground and background pixels. Binarization is a technique that automatically chooses a threshold that separates the foreground region with a single intensity and background region with a different intensity [4]. The two common techniques of binrazation

are global and adaptive binrazation techniques. Global thresholding techniques effectively applied to separate foreground and background of images because it can works well with variable illumination, shadows, smears and blurred documents. Adaptive thresholding changes the threshold dynamically over the image. This change of threshold value is done for each specified area of the image. For each pixel in the image, applying adaptive thresholding, a threshold has to be calculated on the move from its pixels. If the pixel value is below the threshold value computed uniquely for the target pixel, it is set to the background value; otherwise it assumes the foreground value. Thus, for adaptive thresholding, a pixel will be categorized as one of the two possible pixel types, foreground or background, by consulting pixels found in its vicinity [3].

Correcting the skewness of a document, removing underlines, thinning and normalization are also important preprocessing tasks. Thinning is an important approach to represent the shape of a plane region. The objective of thinning is to reduce the representation of a region to a chain of single pixel width while preserving all other relevant features. Normalization is the process of enlarging and shrinking an image size. It scales the input image to a manageable size for the recognizer and for subsequent preprocessor stages. Good preprocessing techniques can greatly minimize overall OCR error rates by reducing misclassification during the stage of character labeling to their proper classes [26] [4].

Segmentation: is a process that determines the elements of an image. Worku [19], described segmentation as a process of separation of an image into regions that contain pixel groups that are similar in value. It is considered as the most important part of recognition system because of the direct dependency of correct recognition on correct segmentation [8].Thus, the performance of OCR systems depends heavily on the page segmentation algorithm used. Million and Jawahar

[5] further discussed that during the process of document image retrieval, segmentation occurs at two levels [33]; on the first level, blocks of text, graphics, columns and other parts are separated and it is called page layout segmentation. On the second level, text lines, words and characters in the text image are located and it's called text segmentation.

The goal of page layout segmentation is to divide different components of document images in to homogeneous graphics, columns etc. Thus, it is an essential step before other OCR operations including segmentation of text block in to lines, words and characters. The accuracy in each stage of text segmentation assures the effectiveness of the result of recognizer. There are various types of text segmentation techniques and algorithms available [29]. The two commonly used text segmentation algorithms are stage by stage and recursive segmentation. In stage by stage algorithm a character is segmented in three steps: line segmentation, word segmentation, and character segmentation. Recursive segmentation is an approach that merges segmentation and recognition together. It is said to be convenient for characters of connected nature [8]. The problems in segmentation are divided into various categories: extraction of touching and fragmented characters, distinguishing noise from text and skewing [6].

Feature extraction: is the phase that analyzes a given character segment and selects a set of features that can be used to uniquely identify [18]. It's responsible for extracting these features that differentiate representations from the matrices of digitized characters so that the characters are easily recognized by the classifier. The main goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements and generate similar feature set for variety of instances of the same symbol [30].

Features extraction for a character in OCR systems can be broadly classified into Structural/Topological features and Global/Statistical feature [1]. Structural or topological feature is concerned with issues related with the geometrical and topological properties of the character, whereas global or statistical features are obtained from the arrangement of points constituting the character matrix and studies how there are various statistical feature extraction techniques available. Some of the common techniques are Zoning, Moments, Projection histograms, N-tuples, Crossings and distances [29][20][31]. Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition systems [10].

Classification: This is the decision part of the OCR systems that takes the output of feature extraction phase and label characters to their proper class. This module mainly performs two main tasks: training and testing. During training phase, the classifier learns the association between samples and their labels from labeled samples. The testing phase involves analysis of errors in the classification of unlabelled samples in order to evaluate classifier's performance. The most commonly used classification techniques are template matching, statistical classification, syntactic or structural matching, neural networks and kernel method [4][12][8][1].

Template matching is pattern recognition approach where new patterns are matched with stored patterns. While comparing new instances from stored patterns, the size and style of characters can be negotiated. Statistical pattern recognition relies on defining a set of decision rules based on standard statistical theory. Any character recognizers are based on mathematical formalisms that minimize a measure of misclassification [1].

Syntactic/structural methods 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. Generally, a character is represented as a production rules structure, whose left-hand side represents character labels and whose right-hand side represents string of primitives. The right-hand side of rules is compared to the string of primitives extracted from a word [1].

Artificial Neural Network (ANNs), which are closer to theories of human perception, employs mathematical minimization techniques. Use of ANN systems, offer a new computing paradigm in which the network, through a process of learning from task examples can store experimental knowledge and make it available for use at a later time [3].Kernel methods are powerful classifiers which include Support Vector Machines (SVM), Kernel Principal Component Analysis (KPCA), Kernel Fisher Discriminant Analysis (KFDA) etc[1]. Among those different types of classification methods, SVM and ANN are the most dominant classifiers that are used in the field of character recognition [32].

Post-processing: After the classification task is performed and recognition is done, this phase will take the results as an input to a further process of check and correct errors. This is due to the fact that classification algorithms are not perfect and they always make mistakes. Especially for degraded documents and alphabets that are very similar, misclassification of characters is always there. In order to enhance the accuracy of recognition process the post processing stage should handle issues with non-word and real word errors [47].

From the whole architectural phases of an OCR system, this study focus on page segmentation part, particularly page column and paragraph layout segmentation of an image document with the aim of reconstructing those layouts. Other phases of the system adopt from previous studies

and integrate the proposed page column and paragraph layout segmentation and reconstruction techniques with the existing prototype of Amharic OCR system.

2.3. Page layout Segmentation

Page segmentation into text and non-text components is an essential step before sequential OCR operation preceding. As Gedion [17] discussed, in document images, basic shapes of text characters are limited in number, but shapes of the non-text components including graphics, column, drawing, logos, tables, etc. are unlimited. Thus, both text and non-text components in OCR engines algorithms are approached differently, as a result they only recognize text components and then arrange recognized text and images of non-text components in an output document using layout information.

Page layout segmentation is the process through which the regions of interest from a document available as an image are being classified. The components of page layout segmentation are: the geometrical (physical) layout and the logical layout [35]. The task of geometrical layout detection and segmentation is to detect, label and segment the document image into homogeneous zones, each consisting of only one physical layout structure (graphics, column, text, illustration, mathematic symbols, tables, etc.), and to identify their spatial relationship. The logical layout refers to the detecting of the logical role that various regions have in the document (titles, footnotes, etc.) Page segmentation is important in the OCR system to maintain those physical and logical layouts; thereby text can be differentiated from images in the OCR systems for further processing [35].

There are several page segmentation algorithms that have been proposed in order to ensure optimal character recognition, minimum distortion, searchable and reusable documents. Those

algorithms can be categorized into three classes [45]: top-down approaches, bottom-up approaches and hybrid approaches.

2.3.1. Top-down Approach

The top-down approaches recursively segment large regions in a document into smaller sub regions. The segmentation process stops when criterion is met and the ranges obtained at that stage constitute the final segmentation results. Most of well-known top-down methods are XY cut, Projection Profile Methods, Whitespace Cover, Histogram Analysis and Space Transforms [34] [45].

X-Y cut algorithm also called recursive x-y cuts (RXYC) algorithm is a top-down algorithm which partitions a document into rectangular components which represent the nodes of the tree. It follows a tree-based approach; the root of the tree represents the entire document page and all the leaf nodes together represent the final segmentation. The bits of the binary transformed image are summed by this algorithm. In this manner a density graph is obtained. The low ends of the graph stand for empty spaces that are lines in the segmented document. If the values reach a higher point than it, the segmentation process is stopped and the layout component is identified. This algorithm is usable for both horizontal and vertical projections. The process is continued until an empty line threshold is reached. At that point the segmentation algorithm ends [6][37].

A positive aspect of the algorithm is that the threshold controls the size of the found component. This way, the configuration setting makes the algorithm suitable for finding paragraphs, lines or words and many other elements of the page by simply performing an adjustment to one parameter. More than that, the threshold controls the size of the segmented clusters which

makes the algorithm scalable. The detection of the rows, paragraphs, section and so on is therefore possible [6].

The whitespace cover algorithm considers a collection of rectangular components as well as another component that represents the entire page and it is a container for all the other components. The main idea behind this algorithm is the maximization of the white spaces in order to obtain the optimal page. The algorithm principle is similar to quick sort. The starting point of the algorithm is represented by a rectangular component that bounds the whole page and a group of white rectangles which are considered obstacles. One of these obstacles is chosen as a pivot. Usually it is chosen one that is as central as possible. With this pivot the space is split in 4 components (right, left, top, down) which become candidates for being processed in the same way recursively. Every component is tested with a quality function in order to evaluate if there is a white space or not [6][38].

2.3.2. Bottom-up Approach

The bottom-up methods start by grouping pixels of interest and merging them into larger blocks or connected components, such as characters which are then clustered into words, lines or blocks of text. Some of the methods used here are Connected Component (CC) Analysis, Region-Growing Methods, Docstrum, Voronoi-Diagram Based, Run Length Smoothing, Smearing, Neural Networks and Active Contours [24] [45].

Connected component (CC) analysis is a bottom-up technique that scans all the pixels of document image and recursively label them based on pixel connectivity, i.e. all pixels in the connected component share similar pixel intensity values and are in some way connected with each other. Figure 2.2 shows the result of extraction of figures and the caption line candidates for the extracted figures using the rule-based CC position and area analysis approach in [39]. It uses

CC area height analysis, which takes the width and height of the bounding box of a labeled component to calculate the area in order to get some threshold value, to identify big components and extract near connected components as figure caption candidates.

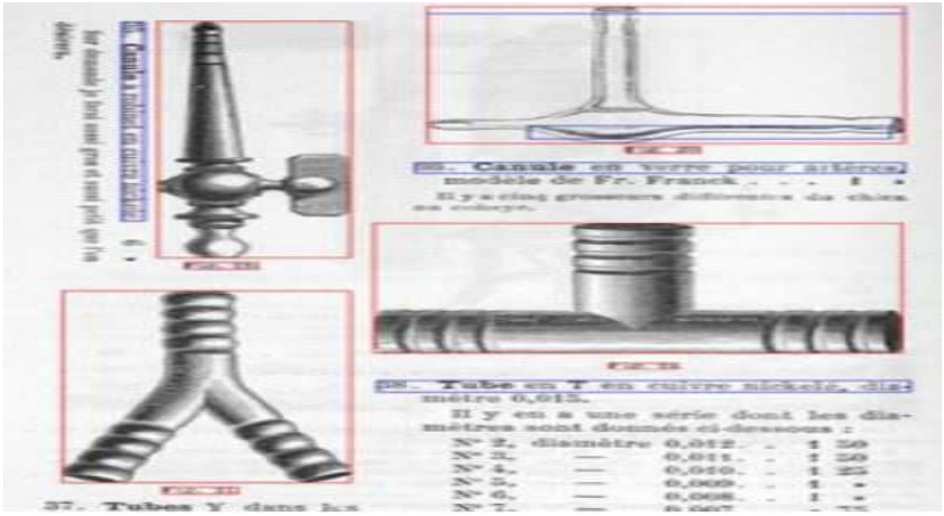


Figure 2.2: Segmentation of Figures and Figure Caption Candidates by CC Analysis [39]

Docstrum algorithm is one of the bottom-up algorithms based on nearest-neighborhood clustering of connected components extracted from the document image. After noise removal, the connected components are separated into two groups, one with characters of the dominant font size and another one with characters in titles and section headings, using a character size ratio factor. Then, K nearest neighbors is found for each connected components. A histogram of the distance and angle of each connected component from its K nearest neighbors is computed. The peak of the angle histogram gives the dominant skew in the document image. This skew estimate is used to compute within-line nearest neighbor pairs. Then, text-lines are found by computing the transitive closure on within-line nearest neighbor pairings using a threshold. Finally, text-lines are merged to form text blocks using a parallel distance threshold and a perpendicular distance threshold [40].

Voronoi-Diagram Based Algorithm: is an algorithm which first extracts sample points from the boundaries of the connected components using a sampling rate. Then, noise removal is done using a maximum noise zone size threshold, in addition to width, height, and aspect ratio thresholds. After that a Voronoi diagram is generated using sample points obtained from the borders of the connected components. The Voronoi edges that pass through a connected component are deleted to obtain an area Voronoi diagram. Finally, superfluous Voronoi edges are deleted to obtain boundaries of document components [41].

Smearing algorithm starts from the idea of extending black pixels from a binary image. This process aims to make the reconstruction of lost structures resulting from the digital conversion process by having groups of pixels turned into black where the number of white pixels is in minority based on a predetermined threshold. In the traversing process of the binary image, if the number reached by 0 pixels (in binary image this value stands for white color) is below a limit that have been given, then the neighbor pixels are transformed into 1 (value for black). If the lower limit is not reached or exceeded, the pixels are not changed. The traversal is made both horizontally and vertically and the results are then combined to gain the final solution. The algorithm is effective and simple and it depends on how the threshold value is set, value that depends on the type of document, namely the information density which is reflected in the number of black pixels. An important drawback of this algorithm is that it can only work with rectangular structures, as the traversal is done linearly [6].

2.3.3. Hybrid Approach

The hybrid methods are the combination of both top-down and bottom-up strategies. Take 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 [45][36][6]. Any of page segmentation techniques are categorized under one of the three approaches.

2.4. Real life Amharic documents

A number of Amharic documents such as letters, newspapers, magazines, books and even old manuscripts are available in governmental, religious and private institutes. These documents in general are grouped into printed, handwritten and typewritten. Printed Amharic documents have different fonts. Some of commonly used fonts in computer for printed Amharic documents are 'Power Geez', 'Visual Geez', and Nyala‘[17]. Having different font style on printed Amharic documents is one of the main challenge to develop Amharic OCR [3][4][5].

There are also a number of typewritten documents produced in the form of books, magazines, correspondence letters, etc. Typewritten Amharic document individual characters have different height and width but the space between characters is proportional [20]. Typewriter has dust filled print heads and other scrapes of ink from the ribbon. Thus, loop appendages of some characters and words appear as solid black circular image in most typewritten documents and that makes character recognition of typewritten documents difficult [13].

Handwriting 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 [20]. It is the most dominant means of written communication. There is no clear rule that abandons cursive handwriting. Commonly, hand written documents are often written in a disconnected, but non-uniform manner. Thus, automating handwritten documents is difficult due the non-uniformity manner of the document [23][21].

2.5. Challenges in Amharic document recognition

In the course of developing Amharic OCR previous studies faced a number of challenges due to the different characteristic of the language. As discussed by Million and Jawahar [5][10] Michael [4] and Abay [3], the total number of characters in Amharic script is more than three hundred. Existence of such a large number of Amharic characters in the writing system is a great challenge in the development of Amharic character recognizer. It needs very intensive memory and computational requirements. Printing variations is also a challenge. Printed Amharic documents are varying in fonts, sizes and styles.

There are a number of similar characters in Amharic script that are sometimes difficult even for humans to distinguish them easily. The shape of many Amharic characters shows similarities with few distinctions among them, many basic characters are also clearly related in structure. There are also remarkable differences in shapes among the basic characters. Amharic characters can differ in size. There are very short characters (such as ፈ, ሠ, መ) and there are very long characters (such as ቸ, ኸ, ኽ). There is also noticeable variance in width, for instance between ነ, ማ, and ፌ. Robust discriminant features needs to be extracted for classification of each of the character into their proper category or class [10]. A lack of standard representation for the fonts and encoding, lack of support from operating systems, browsers and keyboard, and lack of language processing routines are also the major concern that Amharic document recognition system faces [10] [5] [3].

Other than Amharic scripts characteristics, degradation of documents are the main issues added to the complexity of the design and implementation of an optical character recognition system for Amharic language. Document images from real life documents, such as books, magazines,

newspapers, etc. are extremely poor in quality. Popular artifacts in printed document images include: excessive dusty noise, large ink-blobs joining disjoint characters or components, vertical cuts due to folding of the paper, cuts at arbitrary direction due to paper quality or foreign material, poor quality of paper and ink, Floating ink from facing pages etc are the causes for document image degradation [10] [5] [4].

2.6. Related Local Researches

Developing OCR systems for Amharic characters was not a recent research focus. There have been various attempts done in the area by different scholars on printed, typewritten, and handwritten documents. Some of the local researches conducted for developing Amharic character recognition are presented below.

The first attempt made by Worku [19], built an Amharic OCR for printed Amharic characters. He tested the stage by stage segmentation algorithm that was suggested by Pal and Chaudhury that operates in three successive steps of detecting lines, words and characters from the document image respectively. He adopted binary tree classifier for recognition phase and better performance was seen on his test set for unconnected characters. However, the algorithm considered all connected characters as a single character. Consequently, the system performance was very poor for degrading and connected characters [19].

As a continuation of Worku's effort, Ermias [15] in his predecessor footstep attempted recognition of formatted Amharic texts by using the algorithm suggested by Pal and Chaudhuri for the removal of the matra line from the Bangla script. He modified the algorithm for underline detection and removal by adjusting the threshold value. He tested the Zang-Suen and Hilditch thinning and size normalization algorithms for Amharic writing system. Ermias

adopted Worku's algorithm to evaluate the performance of the system. Conversely, the result of the study was poor and he mentioned the reason behind this was the widths of characters are changed by adopted thinning algorithm [15].

Berhanu [18] conducted a study on Amharic character recognition for printed documents which have the same font face. He implemented the segmentation process (using stage by stage algorithm) in two steps: line segmentation and character segmentation and he used size normalizing algorithms to scale segmented character images before feeding to ANN for classification. However, the study shows poor recognition performance and he suggested feature studies to explore on segmentation, image thresholding and noise removal algorithms [18].

Dereje [20] has also attempted to further work on a research in the area with the aim of improving the Amharic OCR by enabling it recognize typewritten Amharic text. Based on his findings, Dereje mainly recommended that in order to enhance the recognition accuracy of Amharic OCR system, it is important to adopt recognition algorithms that are not very sensitive to the features of the writing styles of characters [20].

Negussie [21] conducted on recognition of Amharic handwritten characters for bank check amounts. He applied underline removal, slant normalization and character size normalization. He adopts a stage by stage segmentation algorithm which was used by previous studies 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 [21].

Million [8] conducted a research with a generalized approach that enable previously adopted algorithms recognize Amharic text with a different font styles and faces. He tested the Zang-Suen thinning algorithm in addition to parallel thinning algorithm suggested by Ha and Bunke.

Million integrated the two algorithms through iterative experimentation and come up with the hybrid algorithm which shows a remarkable result. He used stage by stage segmentation and binary tree for feature extraction and classification. Even though promising results achieved, the performance of the system was poor. Thus, he recommended a generalized and more flexible recognition algorithm to developed [8].

Yaregal [11] works on OCR of Amharic text as an integrated approach by applying structural/topological patterns (primitives) and ANN for classification for different font size. The achieved result showed that the approach used is more or less independent of the font size. Mesay [22] used line fitting to Amharic Optical character recognition by applying simple geometric calculations to determine features which could represent and describe the character as uniquely and precisely as possible. He used feed forward Neural Network back propagation algorithm for recognition and achieved 91.9% recognition accuracy. He noted the system would be more versatile if sufficient training data was obtained on the classified characters [22].

Abinet [44], attempt to develop an online handwriting character recognition engine for 33+1 basic Ethiopic characters. This new online handwriting data representation scheme that makes use of the X and Y coordinate observation code sequences applied for feature extraction. On the average, a recognition accuracy of up to 99.4% is achieved for the sample documents [44].

Million and Jawahar [10] propose a two-stage feature extraction scheme using principal component analysis (PCA) and linear discriminant analysis (LDA), followed by a decision directed acyclic graph (DAG) classifier SVM as the nodes for the development of OCR system for Amharic language. 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. They extracted features from the entire image by concatenating all the rows to form a single contiguous vector. Discriminant features are extracted for classification using the proposed two stage dimensionality reduction scheme. High recognition rate was achieved for both printed and real life poor quality documents. On the average 96.95% accuracy is obtained on different dataset 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 [10].

Abay [3] developed OCR for real life Amharic documents by using artificial neural network for classifying the features. He had tested wiener adaptive filtering for noise removal, otsu global thresholding for binarizing the digits image, liner interpolation techniques for normalization, hit and miss morphological analysis for training on real life Amharic documents. Abay managed to recognize and come up with a 96.87% on average on training sets but there was poor recognition rate achieved for the degraded real life documents [3].

Biniam [28] has tried to integrate effective image preprocessing techniques of noise reduction and thresholding as well as multiple words rendering and querying to enhance the effectiveness of relevant document retrieval from printed real-life Amharic document images. Gedion [17] 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. Michael [4] developed recognition of real life documents. He used Weiner filtering algorithm and Sauvola algorithms for segmenting lines, words and characters. He also applied the underline removal and normalization methods. Modified zoning technique is employed for feature extraction and for classification purpose, multiclass SVM is employed. Better results

were achieved and he recommended better segmentation algorithm should be explored in order to increase the recognition rate.

Berhanu [24], attempts to explore an effective page and text segmentation method to improve the applicability and performance of Amharic OCR for real life documents. He introduced a new method based on CC analysis to segment overlapping characters, and to detect and split connected characters. He integrated with previously developed Amharic OCR system, and he managed to achieve 79.13% recognition accuracy rate.

Most of the previous studies focus only on character recognition. Page segmentation steps were used as preprocessing step to remove non-text element of a document image. Any of those researches didn't try to reconstruct the layout of the page and end up only producing a plain text output. However, implementing effective page segmentation technique which can work on all type of real-life documents is vital to preserve an original document image page layout, Hence, this study especially dedicated on exploring and implementing page segmentation algorithms that is suitable for text/graphics separation, column and paragraph layout detection and reconstruction of those detected blocks after recognition.

CHAPTER THREE

PAGE LAYOUT SEGMENTATION

Page layout segmentation is a union of geometric and logical labeling [44]. Detection and labeling of the different zones (or blocks) as text body, graphics, column, illustrations, math symbols, and a table embedded in a document is called geometric or physical layout segmentation. It is essential to enable an OCR engine to process images of arbitrary pages, such as from books, magazines, journals, newspapers, letters, and reports [45]. On the other hand, text zones play different logical roles inside the document (titles, paragraphs, captions, headers, footers, numbered lists, etc) and this kind of semantic labeling is the scope of the logical layout segmentation. Most of the times page layout segmentation is applied before even preprocessing techniques take place [46].

Among a number of physical and logical page layouts this study focuses on column and paragraph layouts of a document image. Therefore, this chapter in particular explores architecture of the proposed system, techniques and evaluation metrics for page column and paragraph layouts segmentation and reconstruction of OCR that is going to be applied before and after the recognition process is performed to improve usability, readability and accuracy of recognized Amharic real life documents.

3.1. Architecture of the proposed Amharic OCR system

Every OCR system undergoes a process of sequential stages in order to convert a paper text document into a computer digital text. Figure 3.1 shows the proposed architecture for Amharic OCR system (Rectangles in bolded line represent the main focus area of this research).The recognition process first obtain the original document images to detect and correct the skewness

of the documents before page layout segmentation takes place. Page segmentation stage detects the overall content of the original document, categorizes them in to homogeneous block and stores all the information of the segmented blocks. Then, it passes through preprocessing steps to improve its quality by removing and/or minimizing degradations occurred from scanning, printing, document aging, etc. Some of the operations involved here are underline removal, noise removal and binarization.

The next step is text segmentation, which identifies text line, word and character from a processed text area of a document image. Segmented characters are normalized and fed into feature extraction phase to extract unique features of character images in the form of vector. Then classification followed, which extracts representation of objects from the input document and identify pattern for each class; so that, they can be recognized as characters and words. Finally, page layout reconstruction will follow, which maintain the original documents page layout and write recognized text on specific layout block.

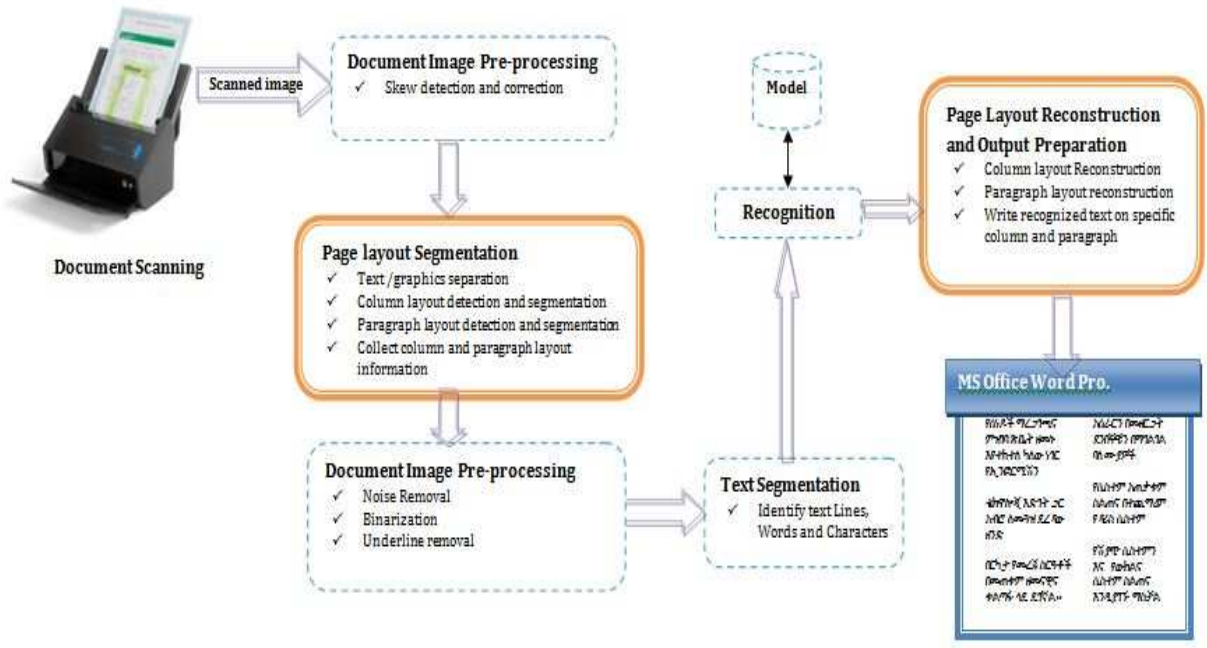


Figure 3.1: Architecture of the proposed Amharic OCR System

The proposed system uses page segmentation step not only for segmenting column and paragraph blocks of the original document image. It also used to collect information about those segmented blocks of image documents in order to make column and paragraph layout reconstruction processes of an OCR feasible.

3.2. Page segmentation techniques

This study explored and tested morphological dilation, connected component labeling or analysis, CC width, height and area analysis and modified whitespace analysis page segmentation techniques for the purpose of column and paragraph block detection, and they all are experimented on different combinations of real life Amharic document images.

3.2.1. Morphological Dilation

Dilation is one of the most basic morphological operations. It is used to connect characters in words, words in a text line, and text lines in a column by adding pixels to the boundaries of objects in an image. The number of pixels added to 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. For instance in a binary image, if any of the neighborhood pixels values are 1, the output pixel is set to 1 and if both of the neighborhood values are 0, the output pixel is set to 0 [49].

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 by using structuring element. In figure 3.2, 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. It is a matrix consisting of only 0's and 1's that can have any arbitrary shape and size. It can be vertical,

horizontal, cross-shaped, multi directional and so on. Based on its shape, structuring element determine to what direction it increase the pixie value of an image [24].

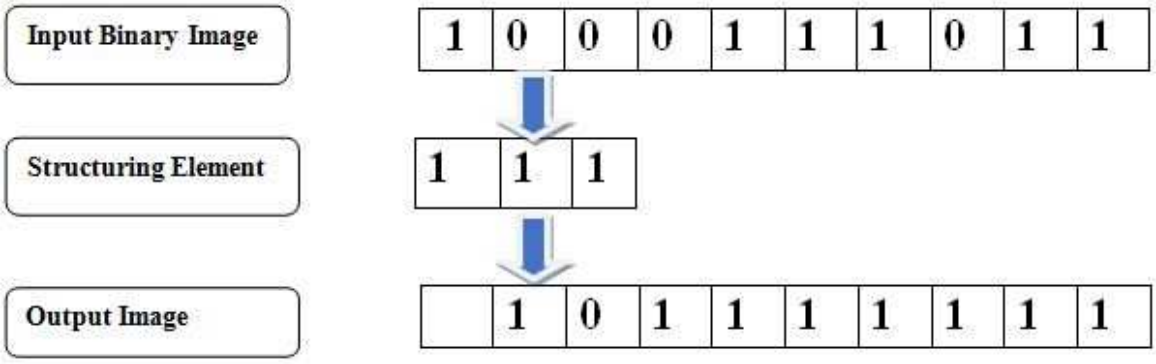


Figure 3.2: Morphological Dilation of a Binary Image

3.2.2. Connected Component Analysis

Connected component (CC) labeling is used in computer vision to detect connected regions in binary digital images. It is an algorithmic application of graph theory, where subsets of connected components within an image are uniquely labeled based on a given heuristic [17]. It scans all the pixels of document image and label them 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.

Connectivity of pixels divided in to 4 and 8 connectivity 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. right, left, up and down neighbors of P whereas 8 connectivity is known as indirect-neighbors checks all the surrounding pixels around

P including diagonal pixels. The labeled pixels represent pixels that are considered as connected to the central pixel in both approaches [50]. Figure 3.3 shows 4 and 8 CC connectivity labeling.

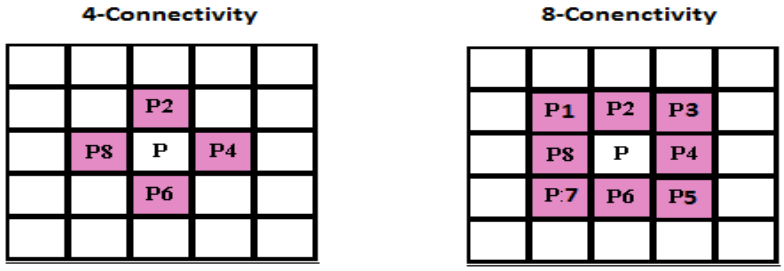


Figure 3.3: Checking mechanism of 4 and 8 CC Connectivity Labeling

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 [24].

There are two types of connected component labeling algorithm; 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. Two pass labeling takes high processing time and memory space than one pass [17]. Algorithm 3.1 below presents the one pass connected component labeling algorithm [50].

.

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

Algorithm 3.1: One pass connected component labeling algorithm

Two pass labeling scans the image two times as it has been mentioned earlier and algorithm 3.2 presents the two pass connected component labeling algorithm [50].

First Pass:

1. Iterate through each element of the data by column, then by row (Raster Scanning)

2. If the element is not the background

2.1. Get the neighboring elements of the current element

2.2. If there are no neighbors, uniquely label the current element and continue

2.3. Otherwise, find the neighbor with the smallest label and assign it to the current element

2.4. Store the equivalence between neighboring labels

Second Pass:

1. Iterate through each element of the data by column, then by row

2. If the element is not the background

2.1. Relabel the element with the lowest equivalent label

Algorithm 3.2: Two pass connected component labeling algorithm

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) [51].

The following figure 3.4 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	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

(b) Connected component labeling

Figure 3.4: A binary image with five connected components of value v = {1} [51]

3.2.3. CC width, height and area analysis

Connected components width, height and area analysis is used to identify big connected elements like: graphics, columns, logos, etc. and small connected elements like punctuation marks and small dots. It is an algorithm that takes the width and height of the bounding box of a labeled component to calculate the area in order to get some threshold value. Column and graphics usually have larger area (height and width) than normal text while punctuation marks

and dots have smaller area and height or width. Thus, finding the area of connected component is very important for page layout segmentation.

3.2.4. Modified White Space Analysis Algorithm

The whitespace analysis algorithm described by [52] analyzes the structure of the white background in document images. The first step is to find a set of maximal white rectangles (called covers) whose union completely covers the background. It has the weighting function to assign higher weight to tall and long rectangles because they are supposed to be meaningful separators of text blocks [52].

Modified whitespace analysis algorithm passes similar scanning procedure to find a continuous whitespace area. However, the scanning process of this algorithm starts from the right end corner of the image and it scans the whole image document pixels vertically until it reaches the bottom end point of the image. When a continuous whitespace area is identified, it immediately converts the remaining black pixels in that particularly identified whitespace area in to white pixels. As a result, connected components break down in to different pieces.

3.3. Performance Evaluation

For measuring the performance of page column and paragraph segmentation and reconstruction direct mapping is used. It determines the performance of layout segmentation and reconstruction by finding the correspondences between detected entities and ground truth [60]. For page column and paragraph segmentation it counts the expected correct and erred segmentation made by the proposed system and calculate the segmentation accuracy percentage. The expected correct segmentation represents the expected number of segmented blocks. Similar procedures are following to calculate accuracy percentage for page column and paragraph layout reconstructions.

CHAPTER FOUR
EXPIRMENTATION

The main purpose of this study is to experiment page column and paragraph layout segmentation techniques on real life Amharic document images with the aim of reconstructing those segmented blocks based on the information stored during page segmentation stage. Unlike the previous researches, page segmentation step uses not only for separating text from non-text regions. It also used to collect information about the different segmented blocks of an image document. The proposed techniques are finally integrated with previously developed Amharic OCR systems to evaluate the change in performance in terms of improving usability, readability and accuracy of character recognition.

For the experimentation purpose, TOSHIBA Intel(R) Core(TM) i3 CPU 3110M @ 2.4GHz (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 preparation and document scanning

Since the goal of this research is to segment and reconstruct column and paragraph layouts of real life Amharic document images, real-life documents taken from the popular government owned Amharic newspaper *'Addis Zemen'* and from *'different magazines* that encompasses multiple columns having graphics and a number of paragraphs are collected and added on the dataset prepared by Berhanu [24]. Scanned images from those documents are selected because they contain a number of page layouts, they are believed to have real-life features and they are

easily accessible. Newspapers and magazines commonly have two columns page layout. The popular Amharic newspaper *'Addis Zemen'* has a maximum of five columns in a single page.

The collected datasets enables us to evaluate if the proposed techniques are adequate to preserving the original document image column and paragraph layouts. The dataset doesn't contain handwritten and typewritten document images; rather it contains multiple column real life documents that have graphics and a number of paragraphs inside. Table 4.1 summarizes document image collections used in this study.

Documents Collected By	**Type and size of documents**		
Berhanu [24]	**Documents Contain**	**Document Type**	**Pages Extracted**
	Columns (graphics)	Newspaper	14 (4)
		Magazine	2
Newly added by the researcher	Columns (graphics)	Newspaper	31 (7)
		printed documents	3
		Total	**50**

Table 4.1: Summary of datasets used in the study

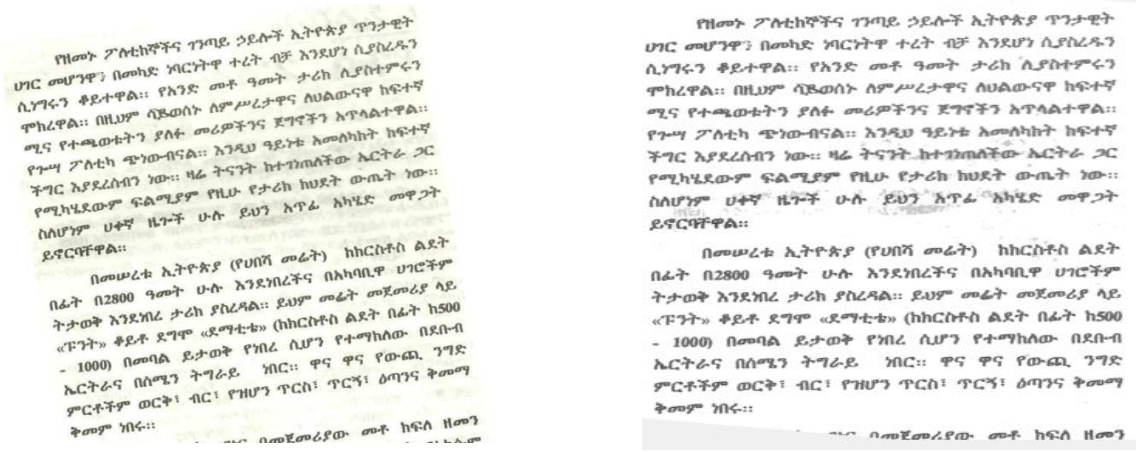
For the conversion of the newly collected documents into their digital format, HP Scanjet 8200 device is used for document scanning. The documents are scanned in grayscale level with zero brightness and contrast levels having a resolution of 300 dpi. The scanned images are stored as BMP image format. This is the first stage in OCR systems concerned with the preparation of sample training and testing datasets in real life documents.

4.2. Skew detection and correction

In a character recognition system, skew detection and correction is typically performed before page layout segmentation. Skew correction generally involves skew angle determination and correction of the document image based on the skew angle. A skew angle is determined through the steps of: providing a set of associated rectangles representing the document image, identifying a column edge associated with the set of associated rectangles, comparing rectangles from the set of associated rectangles to identify those that are in the same column and suitably far apart, calculating a tangential angle between the rectangles identified and identifying the most common tangential angle as the skew angle. Once the skew angle is determined, correction of the document image is made by constructing real skewed rectangles from corresponding extracted rectangles and rotating each of the real skewed rectangles around an origin coordinate for a distance based on the skew angle [53].

Several attempts have been made for skew detection. The methods can be mainly categorized into five groups: the ones based on Hough transformation, cross correlation projection profile, Fourier transformation and k-NN (k nearest neighbors) clustering. Hough transformation is the most popular technique used in detecting the skew angle of a document image [53]. In this study, skew detection and correction is adopted from Berhanu [24] . The algorithm works with 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. This skew detection and correction algorithm first converts the image to 2D color space that is grayscale before skew detection and correction is taking place.

This algorithm correct rotation occurred during document image scanning process by taking the fact that skew angle cannot be greater than 900. The algorithm applied on input scanned document and it automatically detects and corrects the skewed document images. Figure 4.1, shows the result of adopted skew detection and correction technique.



(a) Skewed original document image

(b) De-skewed document image

Figure 4.1: Result of skew detection and correction techniques

4.3. Page Layout Segmentation

Page layout segmentation is the next step to follow after the skewness of an image detected and corrected. It is performed to separate text from non-text region and to store layout information of non-text blocks. Then, the subsequent image processing is applied over the text area to recognize text. Whereas, the information stored plays a vital role in the course of maintaining the original document image page layouts. Therefore, page layout segmentation is an essential stage of an OCR because the remaining stages including text recognition and layout preservation heavily depends on this stage. Thus, in this study, page layout segmentation techniques are applied to extract text from non-text areas and to store column and paragraph layout information with the

aim of reconstructing original document image column and paragraph layouts. As a result, well structured, readable, and usable optical character recognition output produced.

To detect and segment page column and paragraph layouts in a document image MATLAB built-in methods are integrated with Visual C# classes and libraries. The proposed page column and paragraph layout segmentation techniques used morphological dilation, connected component (CC) analysis, CC width, height and area analysis and Modified whitespace analysis techniques for separating graphics from text area, detecting and segmenting column blocks, detecting and segmenting paragraph blocks, collecting information about segmented column and paragraph block and identifying text areas from the document images for further processing. Thus, page layout reconstruction techniques maintain the original document image layout based on the information collected in page layout segmentation stage.

4.3.1. Text/Graphic Separation

Text/graphic segmentation in document images, which separate graphics from text area, is a procedure that must be applied over the image before other stages of OCR system. It is crucial for the next sub-sequential stage, text segmentation which has a great impact to improve the performance of character recognition. In this study, morphological Dilation, Connected Component (CC) analysis and CC width, height and area analysis techniques are applied only for the purpose of extracting texts from documents that contain both text and graphics. Due to time limitation reconstruction of graphics block of a document image is not performed.

The first step in the course of text/graphics separation is dilation. It is used to connect the space between characters, word, lines and so on by increasing the pixel values of a document image. Depending on the structuring element there are different types of dilation. In this work vertical

and horizontal direction dilations are applied. After the document image is connected the next step is CC labeling analysis.

Connected components (CC) labeling algorithm is applied to identify and label each connected component in a given binary image. MATLAB built-in method **bwconncomp()** and **bwlabel()** are used to identify connected component and to label them in a given binary image respectively. In this study four connectivity of pixel are used to identify connected components. Algorithm 4.1 shows CC algorithm that identifies connected components in a given image.

Algorithm 4.1: Identifying Connected Component

```
function [cc, num] = ConnectedComp(binary_image)

    cc = bwconncomp (binary_image, 4); % 4 connectivity
Num=cc.NumObjects;
   .
```

Once connected components are identified and labeled the next step performed on CC labeled document image is connected component width, height and area analysis. In general, it is used to identify big connected elements like graphics, column, etc. and small connected elements like punctuation marks and small dots. Graphics usually have larger height, width and area than normal text while punctuation marks, dots and others have smaller area as well as height and width. Thus, a threshold value in order to separate text from graphics is set by taking the fact that graphics have larger area than text. The width and height of the bounding box are used to compute the area for each component and saved on array **size_info** to compare the results. After an iterative experiment has been conducted 8000 is found to be a better threshold value.

Algorithm 4.2 shows a MATLAB code for height, width and area analysis for text/graphics separation.

Algorithm 4.2: CC height, width and area analysis for Text\Graphics separation

```
size_info = [];
cc = 1;
      forcnt = 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) >8000)
      rectangle ('position',Ibox(:,cnt),'edgecolor','r');
      end
end
```

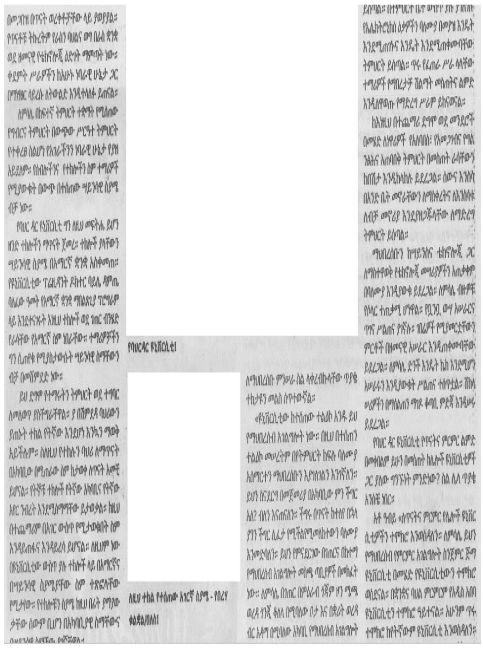
Figure 4.2 shows results of experiment after the proposed text/graphics separation techniques which include dilation, connected component analysis and CC width, height and area analysis applied on a sample images from the dataset.

(a) *original document image*

(b) *Segmented Text area*

(c) *Segmented Graphics*

(d) *original document image*



(e) *Segmented Text area*

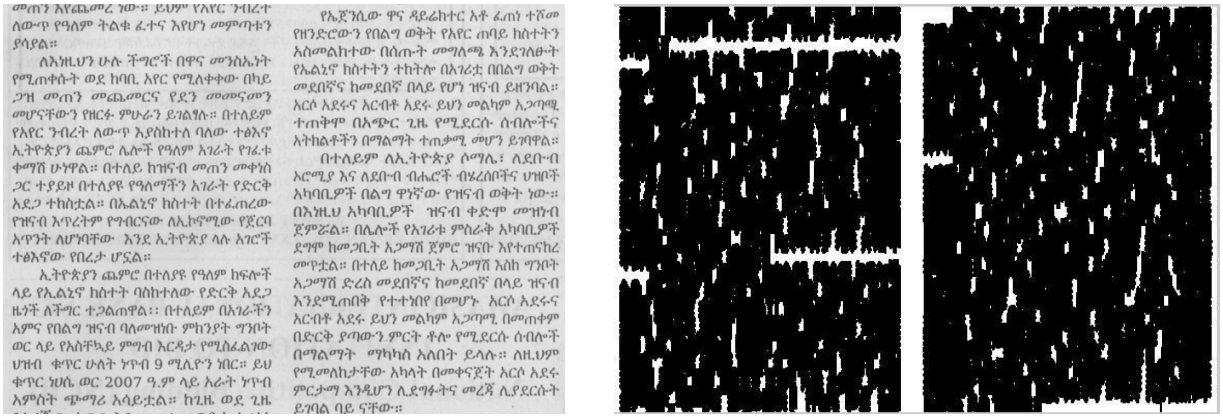
(f) *Segmented Graphics*

Figure 4.3: Experimental result after the proposed text/graphics separation techniques applied

4.3.2. Column Block Segmentation

After separating text area from graphics the next step in this sequential page layout segmentation is column block detection and segmentation. Document images might contain different column blocks. So it is important to detect and segment those regions for the benefit of subsequent stages. Likewise text/graphics separation, these study proposed similar technique for column block detection. It includes morphological dilation; CC labeling analysis and CC height, width and area analysis to identify column blocks from text document images.

Dilation technique used for text/graphics separation is also applied here for column block segmentation. But, in order to keep the white space between the column blocks only vertical direction dilation is applied Figure 4.4 shows the result of connected pixels after dilation algorithm that connects characters, words and text lines is performed.



(a) *original document image* (b) *dilated image*

Figure 4.4: Result after vertical dilation

As we can see from the above dilated image (figure 4.4 (b)), the dilation algorithm connects all the pixels only in vertical direction so that the space between the two columns is reserved. Once the dilation process is done CC labeling analysis is applied and labels all the connected

component regions. Then the area of connected component analyzed in order to find a threshold value to identify column block employed. Algorithm 4.3 shows a MATLAB algorithm to implement CC width, height and area analysis.

Algorithm 4.3: CC height, width and area analysis for Column Block Identification

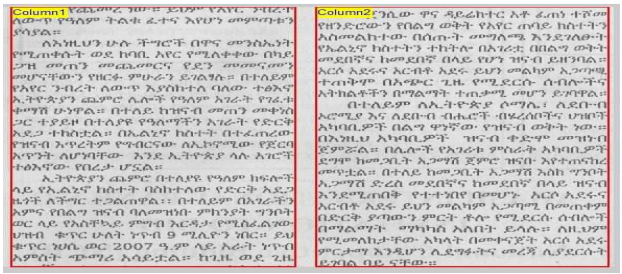
```
size_info = [];
sumArea = 0;
      forcnt = 1:num
      component_area = component_width * component_height;
      size_info (cnt,1) = component_width;
      size_info (cnt,2) = component_height;
      size_info (cnt,3) = component_area;
      sumArea = sumArea + component_area;
      end

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

After conducting an iterative experiment, the height and width of labeled connected component which are greater than one fourth of the maximum area are considered as column block of an image document. Since the goal of this research is to reconstruct detected document image page layout, column layout information is collected immediately after column layouts are segmented.

The proposed column segmentation algorithm produces promising result for document image having different number of columns. Nevertheless, it wrongly detects paragraphs as a column when the whitespace between two consecutive paragraphs is larger, and different blocks of columns are merged when it failed to keep the white space in between due to the presence of tiny

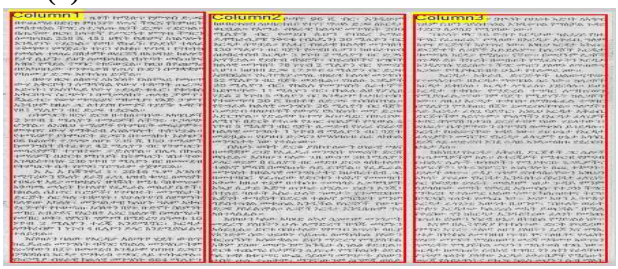
pixels. Thus, it needs further investigation on the area to undertake such problems. The experimental results for the proposed column block detection are presented in figure. 4.5

(a) *original document image with two column*



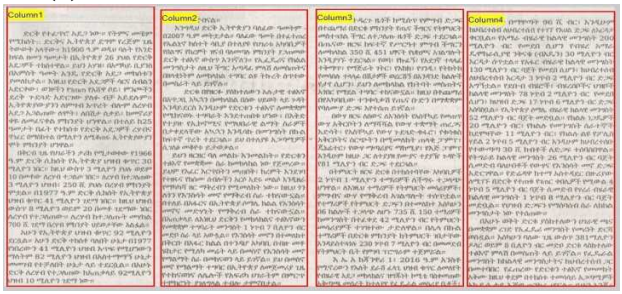
(b) *detected column blocks*

(c) *original document image* with three column



(d) *detected column blocks*

(e) *original document image with four column*



(f) *detected column blocks*

(g) *original document image with five column*

(h) *detected column blocks*

Figure 4.5: Experimental result after the proposed column block segmentation techniques applied

4.3.3. Paragraph Block Segmentation

Once column layout blocks are identified, the next step is paragraph block detection and segmentation on each identified column blocks. Columns, in general text area of a document image contain different logical layout such as paragraph and sentences. It is essential to detect and segment those paragraph blocks to produce well-structured recognition output in OCR.

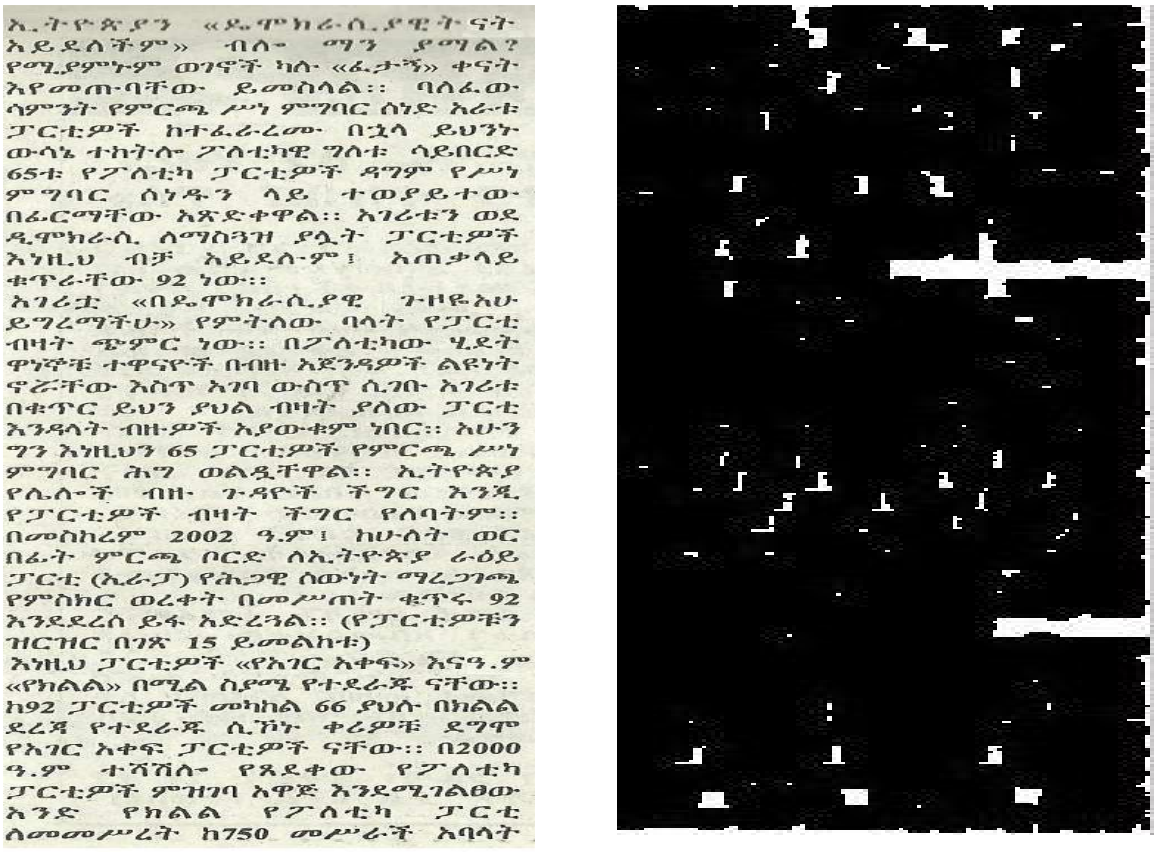
As discussed in previous sections, researches conducted on Amharic OCR only focused on text recognition without preserving the original document layout. As a result, the recognized text outputs produced in previous studies are unstructured and every line is considered as a paragraph. This is due to the lack of paragraph block detection. Thus, this study developed a technique based on morphological dilation, modified whitespace analysis and CC labeling analysis to identify paragraph from the detected column blocks of the document image.

In the previous page layout detection steps, morphological dilation and CC based approach is applied to identify and detect page layouts of each document images. Also, paragraph detection and segmentation from each column blocks is done by using morphological dilation technique. A MATLAB algorithm for dilation shown in Algorithm 4.4 accepts two parameters from the visual C# functions; the binary columned image and the decided threshold value which is structuring element and it returns the dilated binary image based on the input provided.

Algorithm 4.4: Implementation of Dilation

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

Based on the defined structuring element window, a combination of horizontal and vertical dilation is applied to connect the text elements. The result shows connected pixels due to the dilation algorithm that connects characters, words and text lines. Figure 4.8 presents the result after combined horizontal and vertical direction dilation.



(a) *Original binarized image,* (b) *vertically and horizontally dilated image*

Figure 4.6: The Experimental Result after a combined vertical and horizontal dilation

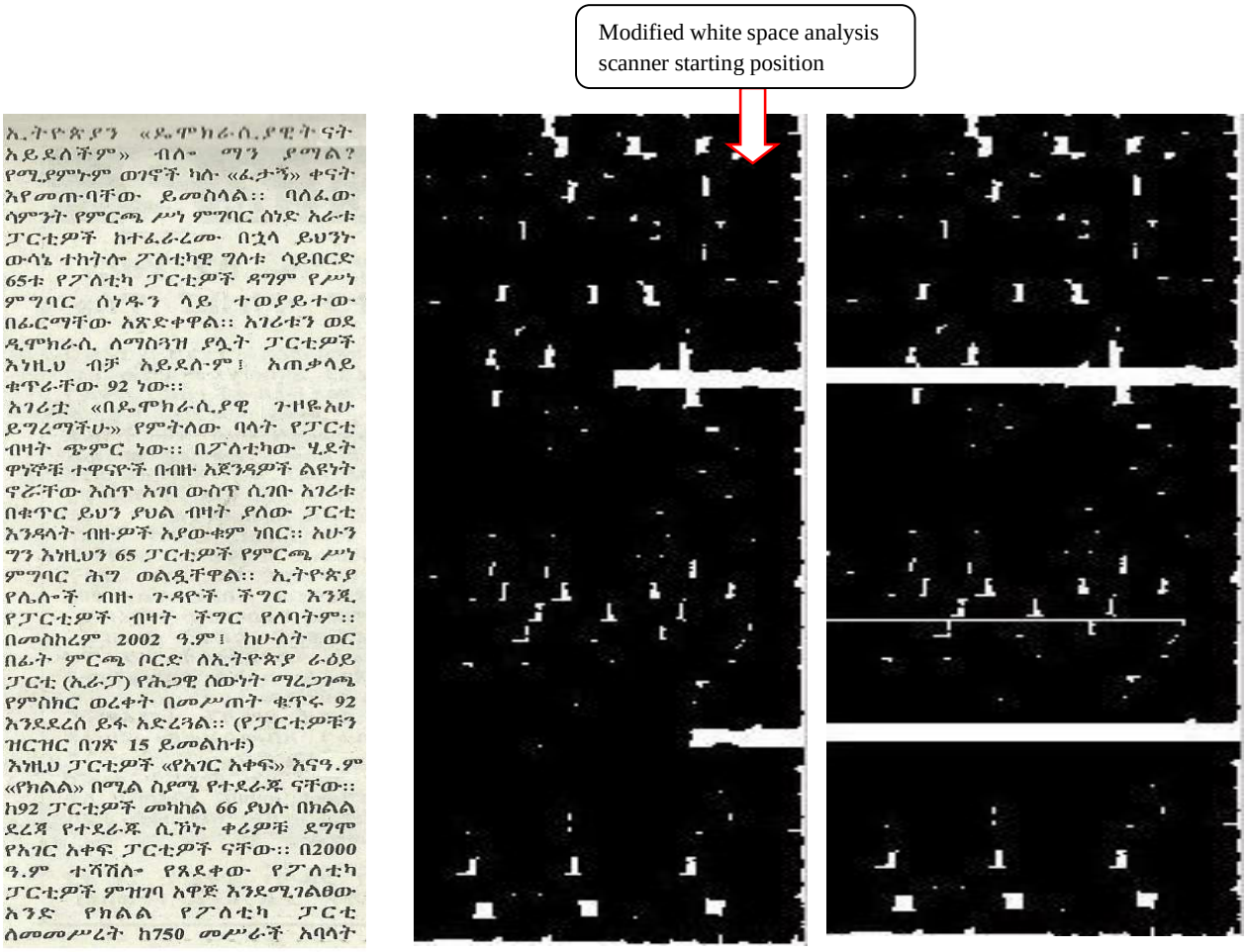
As we can see from vertically and horizontally dilated image (figure 4.6 (b)), whitespaces are created due to the fact that a sequence of 1's pixel values in between characters, word and text lines of a document image. The next step after dilation performed is modified white space analysis. It is a novel approach to find a continuous whitespace in document image. Unlike the

old white space analysis it doesn't scan the whole pixels of an image document to find a white rectangle. Algorithm 4.5 shows modified whitespace analysis.

Algorithm 4.5: Modified whitespace analysis

```
endX  =position_info(1,3) - 50;
endY = position_info(1,4);

for i = 1:endY
        a = d2(i, endX);
    if (a == 0 && i+10 <= endY)
       for cc = i:endY
         if (d2(cc, endX) == 1)
               continue;
         else
                  test = 0;
            for aa = 2:10
                  temp = d2(i+1, endX);

            if (temp == a)
                  temp = d2(i+aa, endX);
                  test = (test + 1);
            else
                        test = 0;
                  end
            end
            if (test == 9)
                  for k = 1:endX
                        d2(cc, k) = 0;
                  end
            end
         end
     end
  end
end
```

The algorithm shown in algorithm 4.5, only scan the document vertically starting from the right top end to the right bottom end coordinates of the image to find a continuous whitespace. Once it detects a continuous whitespace line, it automatically converts the remaining horizontal pixels from right to the left, into white pixels for the purpose of disconnecting paragraphs. Figure 4.9 shows dilated image after modified whitespace analysis is applied.

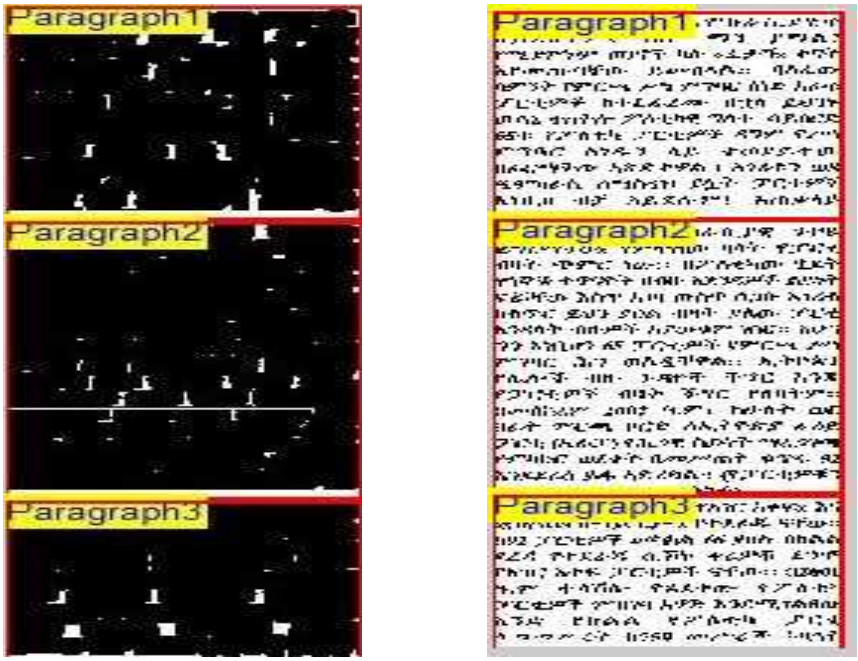


(a) *Original binarized image,* (b) *vertically and horizontally dilated image* (c) *dilated image after modified whitespace analysis applied*

Figure 4.7: The Experimental Result after modified white space analysis applied

As we can see from the experimental results (figure 4.7 (c)), modified whitespace analysis manages to identify a continuous whitespace from dilated image and disconnect connected regions into different blocks. The assumption applied in order to get a paragraph block is the last line of a paragraph. It usually has the smallest length than other lines in a paragraph. So, the modified algorithm is applied to detect a continuous whitespace between connected paragraphs based on the results of the vertical scanning. Then, it disconnecting the remaining CC in that particular line to produce separate sections of paragraph blocks and label them by using CC labeling analysis. Similarly with column layout detection, at this stage information of a

paragraph blocks collects and stored automatically for the purpose of paragraph layout reconstruction. The experimental result of paragraph block detection is presented in figure 4.8.



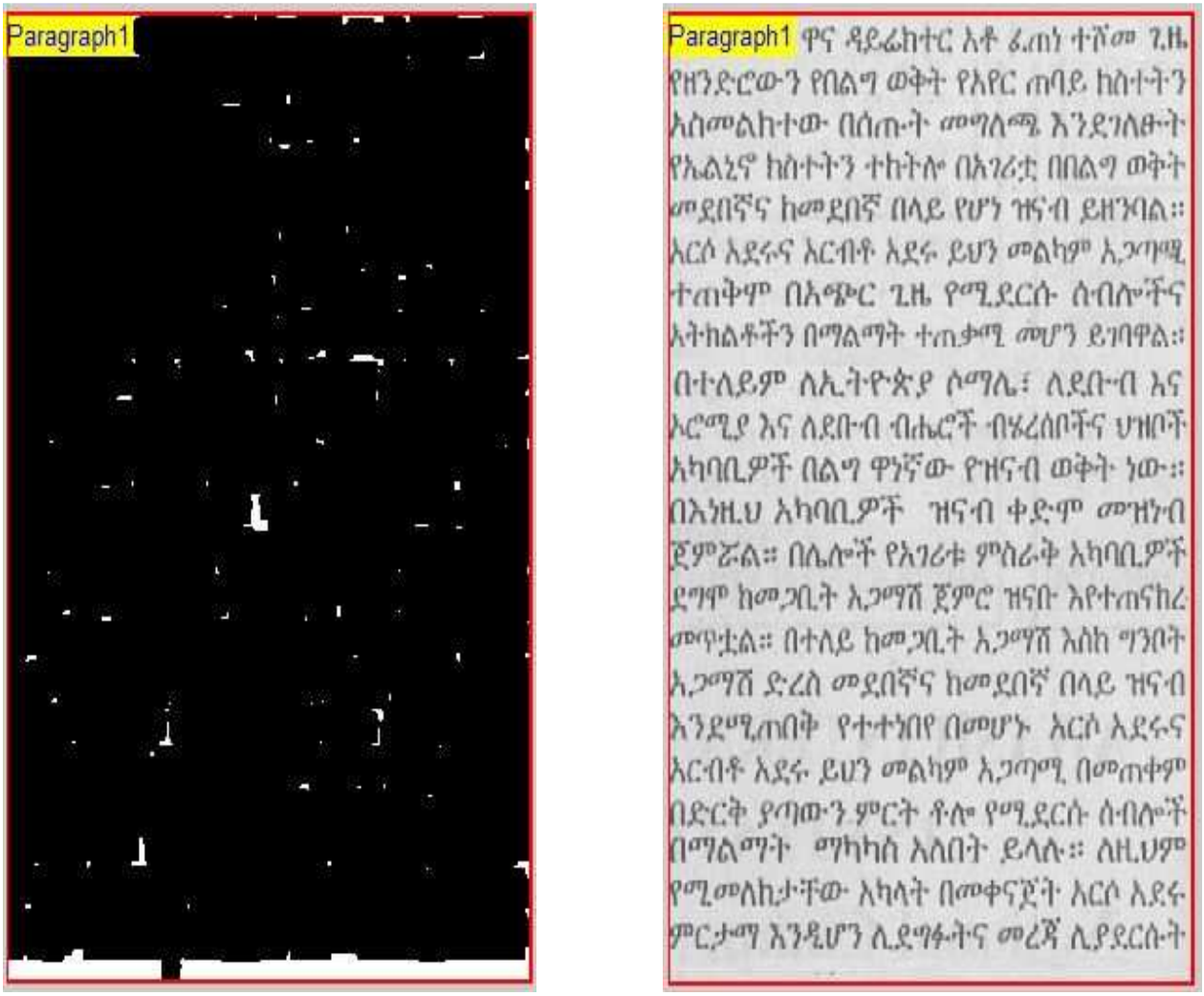
(a) Result of dilated image after modified whitespace analysis and CC labeling

(b) Result of original image after modified whitespace analysis and CC labeling

Figure 4.8: Experimental results of the proposed paragraph block detection

The proposed technique for paragraph block segmentation registers a better performance. However, for a paragraph having equal end point with other lines, the proposed algorithm has failed to detect a paragraph blocks due to the dependency of the algorithm on a whitespace created by shorten last lines of a paragraph. Thus, further studies are needed in this area of paragraph block detection that can solve the limitation of this study. Figure 4.12 shows failed paragraph block detection. Based on the different line and paragraph spacing, the experiment

shown below should have returned two paragraphs blocks but it only produced a single paragraph because of every lines in the paragraph have equal length.



(a) Result of dilated image after modified whitespace analysis and CC labeling

(b) *result of original image after modified whitespace analysis and CC labeling*

Figure 4.9: Failed paragraph block detection

4.4. Column and Paragraph layout reconstruction

Once page layout detection and segmentation is accomplished, the next stage is preprocessing the segmented text area. In this work, preprocessing steps, which includes binrization, noise removal, underline detection and removal, are adopted from previous works by Berhanu [24] and Michael [4]. Wiener filtering algorithm is applied here to remove noise from those text area and sauvola is used to binirize the scanned document image. The techniques are selected among a number of algorithms because they produced a better result in previous studies.

After preprocessing stage-by-stage text segmentation into lines, word and character proceeds. Horizontal projection profile with dilation, vertical projection profile and Modified CC Analysis techniques are used to detect and segment line, word and character respectively. Text segmentation algorithms are also adopted from Berhanu [24] and Michael [4] studies. The detected characters are normalized to extract their feature in order to make them ready for recognition.

Modified zoning algorithm is applied for feature extraction and support vector machine is applied to recognize character from those segmented text of an image document. In this study, the recognition process, which include feature extraction and classification are adopted from Michael [4] and modified for the purpose of layout reconstruction.

The next steps after a sequence of character recognition stages applied to an image document is layout reconstruction. This study uses **Microsoft.Office.InterOP** assembly class library function to create a word document and reconstruct column and paragraph layouts based on the stored page layout information during page segmentation stages. This library is added as a reference to

the C# programming languages. It has various built-in sub methods used for the purpose of creating, adding and modifying different objects to Microsoft Office Word document

The **createDocument()** class is developed to create a word document based on the stored column layout information. Algorithm 4.6 shows the detailed process of creating column layout. Here below, the algorithm accepts **num_col** value (i.e. number of columns) from the stored column layout information and returns columned word document by applying a built in c# method.

Algorithm 4.6: Algorithm to reconstruct segmented column block.

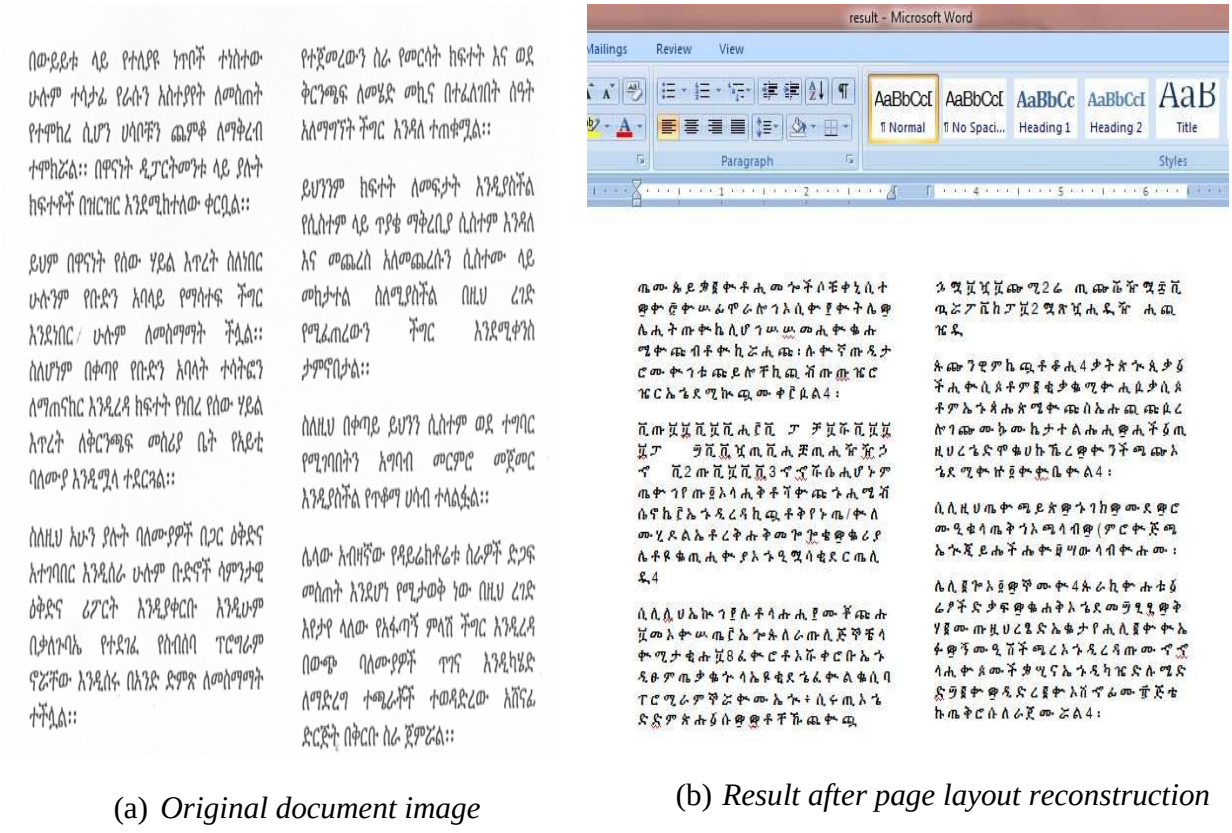
```
PrivatevoidCreateDocument(intnum_col)
        {
try
            {
//Create an instance for word app
              Microsoft.Office.Interop.Word.Applicationwinword =
              newMicrosoft.Office.Interop.Word.Application();

//Set status for word application is to be visible or not.
winword.Visible = false;

//Create a missing variable for missing value
object missing = System.Reflection.Missing.Value;

//Create a new document
Microsoft.Office.Interop.Word.Document document = winword.Documents.Add(ref missing, ref
missing, ref missing, ref missing);

//calling column creator method
//CreateCols(num_col, document);
document.PageSetup.TextColumns.SetCount(num_col);
```

After column layout is reconstructed the next stage is writing individually recognized paragraphs text to their corresponding specific column blocks. As a result, an image document with different column and paragraph block is reconstructed. Algorithm 4.7 shows an algorithm to write recognized text on specific column and paragraph block.

Algorithm 4.7: Algorithm to write recognized texts on specific column and paragraph

```
privatevoidwriteWord(object filename, string result, inti, intnum_para, int j, intnum_col)
        {
Microsoft.Office.Interop.Word.Application word =
newMicrosoft.Office.Interop.Word.Application();
Microsoft.Office.Interop.Word.Document doc = newMicrosoft.Office.Interop.Word.Document();
              .
              .
              .
              .
              .
int z = word.Selection.PageSetup.TextColumns.Count;
varbreakTypes = new[] { WdBreakType.wdColumnBreak, WdBreakType.wdLineBreak};

//if (i == z && j == 1)
if(i == 1 && j == num_para)
              {
word.Selection.TypeText("\n");
word.Selection.TypeText(result);
              }
if (j == num_para&&i != z)
              {
word.Selection.InsertBreak(breakTypes[0]);
              }
else
              {
//if (i == 1 && j == 1)
if (j == 1)
                  {
word.Selection.TypeText(result);
                  }

else
                  {
word.Selection.TypeText("\n");
word.Selection.TypeText(result);
                  }

              }
```

This algorithm accepts **filename** (indicate to which layout block the result belong); **result** (temporarily stored recognized paragraph texts); **num_col** (the number of segmented column); **num_para** (the number of segmented paragraph) and it return a word document with recognized text which are written on specific column and paragraph block by calling **word.Selection** built-in method from C# library function **Microsoft.Office.InterOP.Word**. Figure 4.10 shows the experiment result after the proposed page layout reconstruction techniques are applied.



(a) *Original document image*

(b) *Result after page layout reconstruction*

Figure 4.10: Experiment result after the proposed page layout reconstruction algorithm applied

The proposed column and paragraph layout reconstruction technique properly reconstruct column and paragraph blocks which are correctly segmented in terms of their physical layout (it produced exact number of column and paragraph blocks of the original document image). That means, the performance of the proposed technique is heavily depends on the extent to which the page segmentation method properly detect column and paragraph block.

4.5. The Proposed Page Column and Paragraph Layout Segmentation and Reconstruction Technique

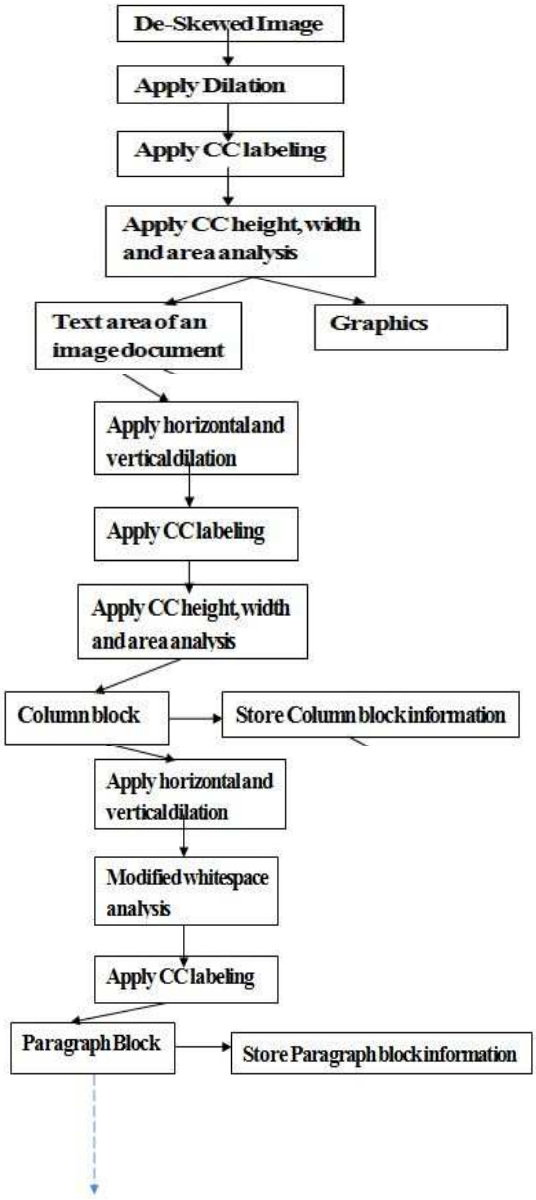
Based on the implementation and test results, this study proposed page segmentation and reconstruction method that integrates some MATLAB and Visual C# functions which are based on horizontal and vertical dilation, CC Labeling, and CC height, width and area analysis algorithms, modified whitespace algorithms. The procedure of the proposed page layout segmentation techniques is presented in figure 4.11.

As discussed in the previous sections, the first processing that must be performed over the document images before any stages of the OCR system is skew detection and correction due to the dependency of the forthcoming stages. The next stage is named as automatic page segmentation that performs a separation of text area from the graphical regions; column blocks detection, paragraph block detection as well as collection of relevant information from segmented blocks.

The input for the proposed automatic segmentation techniques, which is the de-skewed document images, can be in a gray scale/RGB/Binary format. The proposed combined page segmentation technique first applies dilation over the image to connect the words and gaps that exist 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 by applying a vertical dilation and connecting pixels vertically without losing the space between columns. Consequently, CC labeling is performed 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. The next step is finding a paragraph block by applying a combined vertical and horizontal dilation to connect pixels. Then modified whitespace analysis is performed to disconnect connected components by considering it as a paragraph block and finally CC labeling is applied to label the connected components.



Skew corrected document image

Increase pixel value to connect characters and fill gaps

To label connected components over the dilated image

To set better threshold to separate 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 pixels vertically and horizontally without losing the space between columns.

To label all the connected components over the dilated image.

To set threshold automatically for columns blocks

Column layout segmentation will end here and the output is column block and its information stored

To connect pixels vertically and horizontally

To disconnect connected component which have a white space to the right end corner of the image

To label all connected component over modified whitespace analysis

Paragraph layout segmentation will end here and the output is paragraph block and its information stored in a file

Figure 4.11: The proposed Page Segmentation Technique

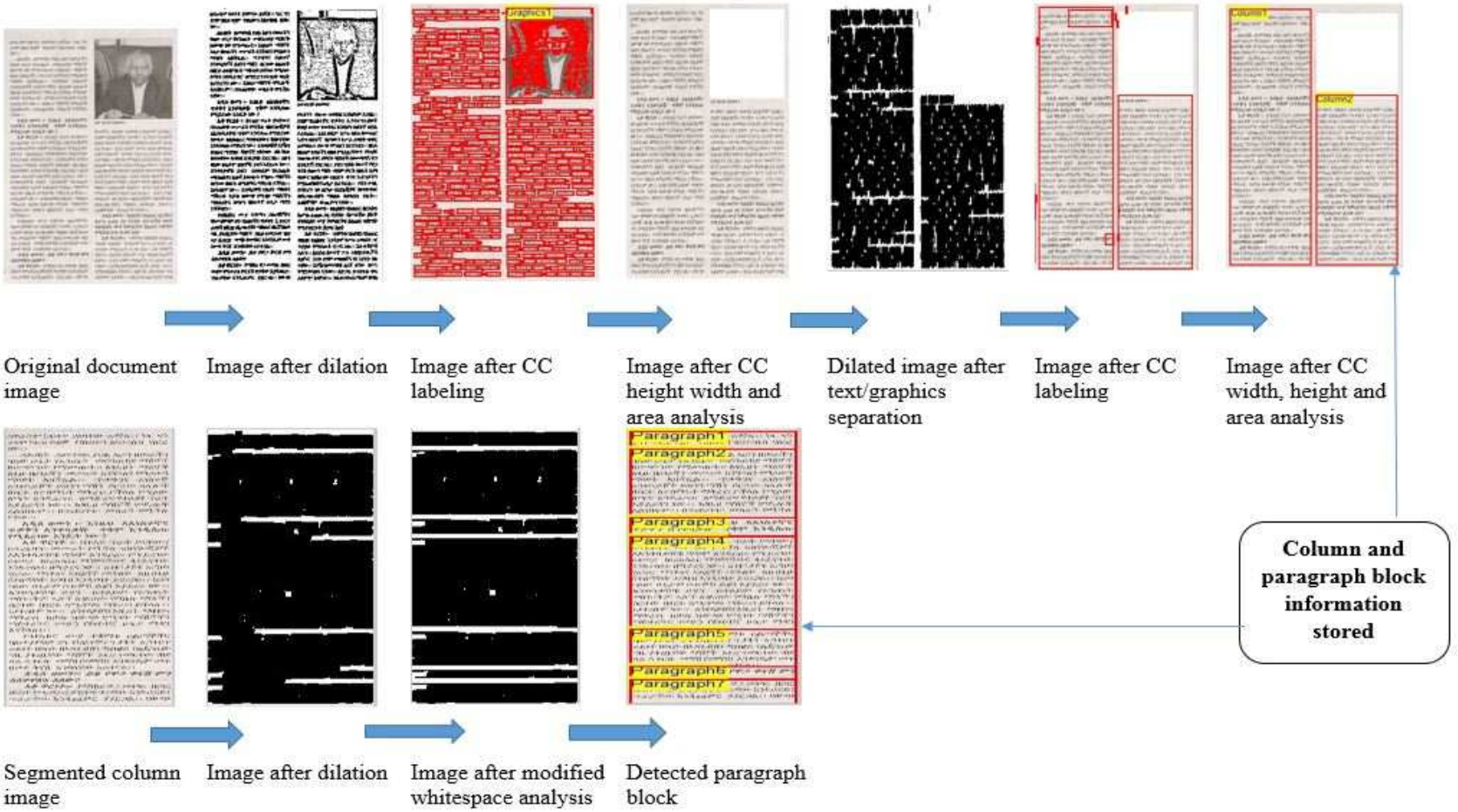
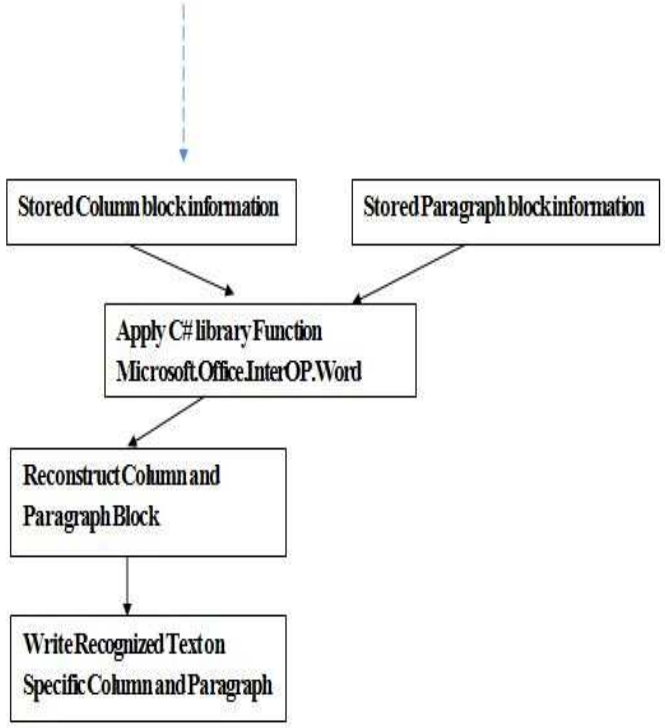


Figure 4.12: Result of the proposed page column and paragraph segmentation technique in every stage

The segmented column and paragraph block information is stored automatically during page segmentation stage, which is important for layout reconstruction stage of the proposed system. Layout reconstruction is performed by using C# built-in library function called **Microsoft.Office.InterOP.word**. The proposed page layout reconstruction techniques is presented in figure 4.13 below.



Stored column and paragraph block information

To create a word document and reconstruct column and paragraph block and to write recognized text on specific columns and paragraph

Figure 4.13: The proposed Page Layout Reconstruction Technique

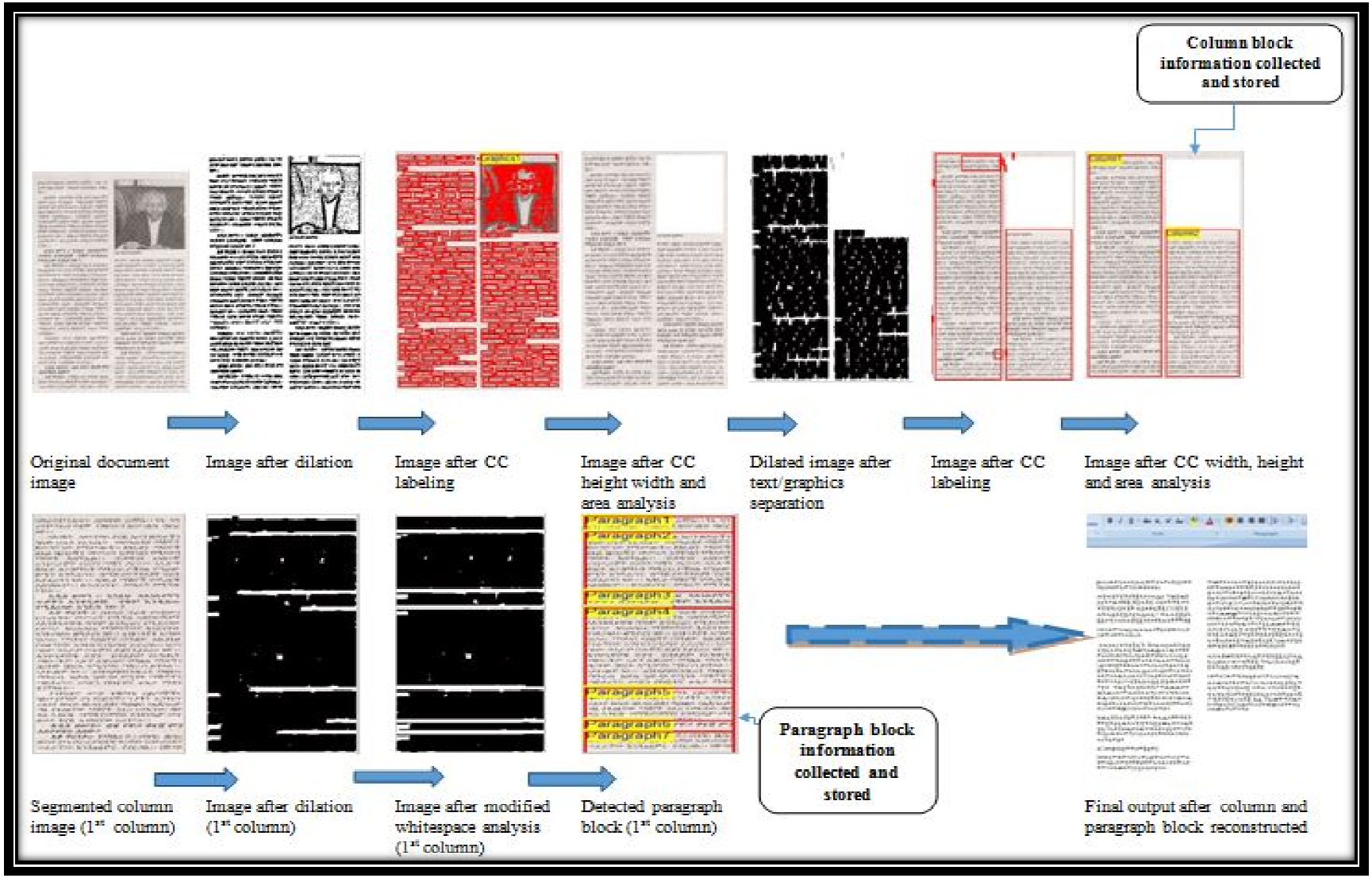


Figure 4.14: Result of the proposed page column and paragraph segmentation and reconstruction technique in every stage

4.6. Experimental result

To measure the performance of the proposed techniques, we use direct mapping. This method counts correct segmentation made by the proposed technique and compare it with the expected once to calculate the accuracy in percentage. The expected correct segmentation represents the expected number graphics, column blocks and paragraph for each page segmentation and reconstruction methods individually. For measuring the performance the test set contains real life Amharic document images with multiple columns, such as graphics with columns and paragraphs inside.

Experimental Results for text/graphics separation:

Real life document image collected for this study has a total number of 50 documents. From the total datasets eleven of the documents have graphics content inside. Table 4.1 shows the performance of the text/graphics separation proposed in this study.

Document Type		**Experiment Result of Text/Graphics Separation**			
		No of Documents	**Correctly Segmented**	**Erroneously Segmented**	**Accuracy (%)**
(i)	Documents containing images/Graphics	11	9	2	81.81%

Table 4.1: Experimental result of the proposed text/graphics separation techniques

The result indicates that the proposed text graphics separation technique based on dilation, CC labeling, CC height, width and area analysis works better on real life document images to separate graphics from text and on average 81.81% accuracy rate achieved. However, the algorithm failed to recognize smaller graphics which have smaller area below the threshold value set.

Experimental results for column layout segmentation

Most Amharic real life documents (i.e. newspapers and magazine) have two columned layouts. However, Amharic newspaper i.e. *AddisZemen* has a maximum of five columns in a single page. Thus, the proposed algorithm tested on document image which have up to five numbers of columns. Test results are presented below in table 4.2

Documents contain column		**Experiment Result of Column Block Segmentation**			
		No of Documents	**Correctly Segmented**	**Erroneously Segmented**	**Accuracy (%)**
(i)	Single column	13	12	1	92.30%
(ii)	Two columns	13	10	2	76.92%
(iii)	Three columns	9	7	1	77.77%
(iv)	Four columns	8	6	2	75%
(v)	Five columns	7	5	2	71.42%
Total		**50**	**40**	**10**	**80%**

Table 4.2: The performance of the proposed column block segmentation techniques

The experimental result indicates that the proposed technique works well for any number of columns with on average accuracy of 80%. However, the proposed technique wrongly detects paragraphs as a column when the whitespace between two consecutive paragraphs is larger. It also merged different blocks of columns as one when it failed to keep the whitespace between the columns.

Experimental results for paragraph block segmentation

Text areas of a column contain a number of paragraph insides. The proposed technique detects paragraph blocks from correctly segmented column. Table 4.3 shows the performance of the proposed paragraph block segmentation.

Document Type		Experiment Result of Paragraph Block Segmentation			
		No of Segmented Column blocks	**Correctly Segmented**	**Erroneously Segmented**	**Accuracy (%)**
(i)	Paragraph blocks from correctly segmented columns	54	39	15	72.22%

Table 4.3: The performance of the proposed paragraph block segmentation techniques

The experimental result indicate that the proposed technique based on dilation, modified whitespace analysis and CC labeling identify paragraph blocks from correctly segmented columns with on average accuracy of 72.22%. Paragraph block detection accuracy is decreased because of the limitation of the proposed technique to detect paragraph blocks when every lines in the paragraph have equal length.

Experimental result for page column and paragraph reconstruction

Page column and paragraph reconstruction of the proposed technique uses the stored information from page segmentation stage to reconstruct the segmented blocks. Table 4.4 presents performance of the proposed column and paragraph block reconstruction.

Document Type		Experiment Result of Page Column and Paragraph Reconstruction							
		From the whole test documents				From correctly segmented blocks			
		No of test documents	**Correctly reconstructed**	**Erroneously reconstructed**	**Accuracy (%)**	**No of correctly segmented blocks**	**Correctly reconstructed**	**Erroneously reconstructed**	**Accuracy (%)**
(i)	Column block reconstruction	50	40	10	80%	40	40	0	100%
(ii)	Paragraph block reconstruction	54	39	15	72.22%	39	39	0	100%

Table 4.4: Experimentation result of the proposed page column and paragraph reconstruction techniques

The experimental result shows that the proposed column and paragraph layout reconstruction techniques heavily depend on the page segmentation stage. It accurately reconstructs column block layouts that are correctly segmented in terms of the number. Because the reconstruction process only focuses on the number of segmented layout block, it doesn't reconstruct column layouts based on their width size. As a result, the reconstructed column layouts have equal width size. Likewise, paragraph block of a document image reconstructed with the accuracy of 100% when the blocks are detected correctly.

4.7. Findings and challenges

This study attempt to reconstruct column and paragraph blocks using the information stored during page segmentation stage. The proposed page segmentation techniques applies morphological dilation, CC labeling, CC height, width and area analysis and modified white space analysis and it performs better to identify column and paragraph blocks. However, detecting paragraph blocks when every lines of a paragraph have equal length is one of the challenges this study faces. The gap between two consecutive paragraphs, the presence of tiny pixels on a whitespace that separate two different column blocks (see figure 4.15) are also the challenges to identify column layouts.

(a) Original two column image (b) Dilated image

Figure 4.15: Failed column layout detection

In general, page layout reconstruction is depends on the page segmentation stage. Thus, the absence of adaptive page segmentation algorithm is the core challenge. There is also a global lack of literatures on the area of page layout reconstruction.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

In this study an attempt is made to detect and reconstruct original document image page column and paragraph layout. For this purpose, page layout segmentation is developed to detect and store information of page column and paragraph layouts. Various researches have been conducted in the course of developing Amharic optical character recognition. Most of these researches focus on maintaining the text part of the document image. Preserving original document layout is important and one of a major concern for character recognition system to increase readability, usability and accuracy by producing well-structured output. Thus, this work provides a significant contribution on the attempt to develop full-fledged Amharic OCR system.

5.1 Conclusion

The main objective of this study is to apply effective page segmentation technique that is capable of identifying column and paragraph blocks of a page in real life Amharic document images. Towards achieving this goal, recursive page layout segmentation is performed to detect and store information about column and paragraph layouts of a document image.

The first step of the proposed page segmentation system is separating text from graphics. After a combination of page segmentation techniques namely: vertical and horizontal dilation; connected component analysis and CC width, height and area analysis are applied a graphics part of a document image is detected and separated from the document. Through an iterative experiment the value of the area greater than 8000 are taken as a threshold value to separate graphics from text. Based on the experiment on the average 81.81% accuracy rate is achieved from the proposed system.

The next step after text/graphics separation is column block detection. Similar page segmentation techniques are applied to segment column layout. The threshold value to determine column block is the value of the maximum area with respect to the height and the width of connected component. If the height and width of the labeled connected component is greater than one fourth of the maximum area then those regions of connected components are considered as column block of an image document. The proposed technique accurately identified column layout with an efficiency of 80%, thereby all information about column block is stored for reconstructing stage.

After page column segmentation, then paragraph layout detection is proceeds. Vertical and horizontal dilation, modified white space analysis and CC analysis applied to detect the layout. Modified whitespace analysis is a novel approach developed based on whitespace cover algorithm to find a continuous whitespace by only scanning the document vertically starting from the right end corner point of an image down to the bottom end point. Once a continuous whitespace line is detected it automatically converts the remaining black pixel in that particular line to white pixel in order to disconnect the connected component. The assumption to find a paragraph block is the last line of a paragraph. It usually has the smallest length than the other lines in a paragraph so we can find a whitespace rectangle in that particular line. Thus, those disconnected regions can be considered as a paragraph. On the average 72.22% accuracy rate was achieved and layout information of a paragraph stored. The slightly lower accuracy rate is attained due to the limitation of modified white space algorithm. This is because of the algorithm failed to identify paragraphs when every lines of a paragraph have equal length.

Based on the stored layout information the original document image column and paragraph block is reconstructed and on the average 80% and 72.22% accuracy rates are achieved respectively from the total document datasets. From correctly segmented column and paragraph block the proposed

techniques 100% preserve page column and paragraph layouts. However, column reconstruction of the proposed system produces equal size of columns even though there are columns with different width size. Because the reconstruction process only focuses on the number of segmented layout blocks, it doesn't reconstruct in terms of the width size.

The major challenges that the proposed system faces include detecting paragraph blocks when every lines of a paragraph have equal length, a large gap between two consecutive paragraphs, the presence of tiny pixels on a whitespace that separate two different column blocks.

5.2 Recommendation

The current work tried to enhance readability, reusability and accuracy of Amharic character recognition by using page segmentation technique not only for extracting text from non-text area. It is also used to store the non-text region information for the purpose of reconstructing the original document image column and paragraph layout. However, to further improve the performance so as to develop a full-fledged Amharic OCR system the following recommendations are forwarded.

- This study reconstructs equal size column block even though a document image has different size column block. Thus, reconstructing column layout based on its width information should be one of future research direction to consider.
- Real life document image has different physical and logical layouts such as table, graphics, header, footer, etc. Hence, future studies can explore on graphics, table and others layouts of real life document preservation.
- Paragraph block detection proposed by this study has a limitation of detecting a paragraph when every line of a paragraph has equal length. Therefore, to coming up with a robust paragraph block detection and segmentation technique will be a future research direction.

- Document images might have overlapping columns. Thus, developing page column segmentation algorithm that can handle such problem is a future research area need to consider.
- Developing adaptive page segmentation algorithm which can identify different blocks of a page intelligently should be one of the future research area to consider
- Future researches also need to explore on a better recognition algorithms in the course of developing applicable Amharic OCR
- Text segmentation should also be one of the future research direction in order to improve the recognition rate of Amharic OCR

References

[1] Verma, Rohit, and Jahid Ali., ""A Survey of Feature Extraction and Classification Techniques in OCR Systems".," *International Journal of Computer Applications & Information Technology (IJCAIT)*, Vol. 1(2):, pp. 1-3, 2012,.

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

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

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

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

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

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

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

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

[10] Million, M., & Jawahar, C. V., "Optical character recognition of Amharic documents.," *African Journal of Information & Communication Technology*, vol. 3(2), ISSN 1449-2679., pp. 14, (2007).

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

(2002).

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

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

[14] Baye, T., (Ethiopian) Writing System., vol. 1, no. 1., Addis Ababa, Ethiopia,: Addis Ababa University, (1992).

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

[16] Sharma, Om Prakash, M. K. Ghose, Krishna Bikram Shah, and Benoy Kumar Thakur., ""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., (2013).

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

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

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

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

[21] Nigussie, T., 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, (2000).

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

[23] M. Wondwossen, OCR for Special Type of Handwritten Amharic Text "Yekum Tsifet" Neural Network Approach. (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University, (2004).

[24] Berhanu, S., Segmentation of real life Amharic documents for improving recognition. (Masters Thesis), Addis Ababa, Ethiopia: Department of Information Science, Addis Ababa University, (2014).

[25] Eikvil, L., Optical Character Recognition., Oslo: Document Image Analysis Publication, (1993).

[26] Alginahi, Y., Preprocessing techniques in character recognition. Character Recognition, Sciyo, (2010).

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

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

[29] Cowell, J., & Hussain, F., "Amharic character recognition using a fast signature based algorithm. In Information Visualization," in *Proceedings. Seventh International Conference on IEEE*, (2003, July).

[30] Tawde, Gaurav Y., and Jayashree M. Kundargi., "An Overview of Feature Extraction Techniques in OCR for Indian Scripts Focused on Offline Handwriting.", *International Journal of Engineering Research and Applications (IJERA),* Vols. Vol. 3, no. 1, pp. 919-926., (2013).

[31] Mori, S., Suen, C. Y., & Yamamoto, K., "Historical review of OCR research and development.," in *Proceedings of the IEEE*, (1992).

[32] A. Ng, *Machine Learning. (Lecture Slides)*, Stanford: Stanford University, 2012..

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

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

[35] Boiangiu, Costin-Anton, et al. "Voting-Based Layout Analysis." *Journal of Information Systems & Operations Management* (2014): 1.

[36] Khurram, k., Recognition. Analysis and Retrieval of Historical Document Images. Ph.D Thesis, Paris,: Universite Paris Descarte,, (2009).

[37] G. Nagy, S. Seth, and M. Viswanathan., "A prototype document image analysis system for

technical journals.," vol. 7(25):, pp. 10–22, (1992).

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

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

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

[41] K. Kise, A. Sato, and M. Iwata,, "Segmentation of page images using the area Voronoi diagram.," vol. 70(3):, no. Computer Vision and Image Understanding,, pp. 370–382,, 1998..

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

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

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

[45] Smith, Ray. "Hybrid page layout analysis via tab-stop detection." *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009.

[46] Bukhari, Syed Saqib. "Generic Methods for Document Layout Analysis and Preprocessing." (2012).

[47] Bassil, Youssef, and Mohammad Alwani. "OCR post-processing error correction algorithm using google online spelling suggestion." *arXiv preprint arXiv:1204.0191* (2012).

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

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

[50] Wikipedia, "http://www.en.wikipedia.org," [Online]. [Accessed 2016].

[51] "http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm," [Online].

[52] H. Bunke, P. Wang, and H. S. Baird, Eds., “Background structure in document images,” in Document Image Analysis, Singapore, 1994, p. pp. 17–34.

[53] Cao, Yang, Shuhua Wang, and Heng Li. "Skew detection and correction in document images based on straight-line fitting." *Pattern Recognition Letters* 24.12 (2003): 1871-1879.

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

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

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

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

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

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

[60] Shafait, Faisal, Daniel Keysers, and Thomas M. Breuel. "Performance evaluation and benchmarking of six-page segmentation algorithms." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30.6 (2008): 941-954.

Appendices

Annex I: Sample MATLAB Codes

Text/graphics separation

```
function [output] = segmentImagePlot(original_image)
    [n, c, numberOfColorChannels] = size(original_image);
    if numberOfColorChannels > 1
      image = rgb2gray(original_image);
      bw = sauvola(image, [30 30]);
    else
      image = original_image; % It's already gray or binary
      bw = image;
    end

    se = [ 0 0 0 0;1 1 1 1;0 0 0 0];
    d = imdilate(~bw, se);
    I3 = imcomplement(~d);
    figure, imshow(original_image);
    [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;

    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);
                cc = cc + 1;
    end

    sumHeight = 0;
    sumWidth = 0;
    sumArea = 0;
    for cnt = 1:num
        sumHeight = sumHeight + size_info (cnt,1);
        sumWidth = sumWidth + size_info (cnt,2);
        sumArea = sumArea + size_info (cnt,3);
    end
    cc = 1;
    for cnt = 1:num
        x = Ibox(:,cnt);
        if (size_info(cnt,3) > 8000)
            txt = strcat('Graphics ',num2str(cc));
            text(x(1,:,1)+1,x(2,:,1)+15,txt,'Color', 'b','BackgroundColor','y')
            rectangle('position',Ibox(:,cnt),'edgecolor','r','LineWidth',2,'LineStyle','-');
```

```
            cc = cc + 1;
        end
    end
    output = original_image;
end
```

Column layout segmentation

```
function [output] = segmentColumnPlot(original_image)
    [n, c, numberOfColorChannels] = size(original_image);
    if numberOfColorChannels > 1
      image = rgb2gray(original_image);
      bw = sauvola(image, [30 30]);
    else
      image = original_image; % It's already gray or binary
      bw = image;
    end


      se = [0 1 1 0 ; 0 1 1 0 ; 0 1 1 0 ; 1 1 1 1 ; 1 1 1 1 ; 1 1 1 1 ;
            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 ; 1 1 1 1 ; 1 1 1 1 ; 1 1 1 1 ; 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 ; 1 1 1 1 ;
            1 1 1 1 ; 1 1 1 1 ; 1 1 1 1 ; 0 1 1 0 ; 0 1 1 0 ; 0 1 1 0];


    d = imdilate(~bw, se);
    d = bwdist(~d) >= 1;
    figure, imshow(original_image);
    figure, imshow(~d);
    [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
        x = Ibox(:,cnt);
        if size_info (cnt,2) > maxArea(1,2)/4 && size_info (cnt,1) > maxArea(1,1)/4
            txt = strcat('Column ',num2str(count));
            text(x(1,:,1)+1,x(2,:,1)+15,txt,'Color', 'b','BackgroundColor','y')
            rectangle('position',Ibox(:,cnt),'edgecolor','r','LineWidth',2,'LineStyle','-');
            count = count + 1;

        end
    end
    output = original_image;
end
```

Paragraph layout segmentation

```
function [output] = segParaPlot(original_image)
    [n, c, numberOfColorChannels] = size(original_image);
    if numberOfColorChannels > 1
      image = rgb2gray(original_image);
      bw = sauvola(image, [30 30]);
    else
      image = original_image; % It's already gray or binary
      bw = image;
    end

    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;

    se2   = [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;
            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];
    d2 = imdilate(d, se2);
    d2 = bwdist(~d2) >= 1;

    figure, imshow(original_image);
    [Ilabel, num] = bwlabel(d2);
    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
endX  = position_info(1,3) - 50;
endY = position_info(1,4);

for i = 1:endY
    a = d2(i, endX);
    if (a == 0 && i+10 <= endY)
        for cc = i:endY
            if (d2(cc, endX) == 1)
                continue;
            else
                test = 0;
                for aa = 2:10
                    temp = d2(i+1, endX);

                    if (temp == a)
                        temp = d2(i+aa, endX);
                        test = (test + 1);
                    else
                        test = 0;
                    end
                end
                if (test == 9)
                    for k = 1:endX
                        d2(cc, k) = 0;
                    end
                end
            end
        end
    end
end
```

```
    [Ilabel, num] = bwlabel(d2);
    Iprops = regionprops(Ilabel);
    Ibox = [Iprops.BoundingBox];
    Ibox = reshape(Ibox,[4 num]);

    len = length(position_info);
    minY = min(position_info(:,2));
    for cnt = 1:num
        x = Ibox(:,cnt);
        if Ibox(3,cnt) > endX - 50
            txt = strcat('Paragraph ',num2str(count));
            text(x(1,:,1)+1,x(2,:,1)+15,txt,'Color', 'b','BackgroundColor','y')
            rectangle('position',[Ibox(1,cnt) Ibox(2,cnt) Ibox(3,cnt)
Ibox(4,cnt)+15],'edgecolor','r','LineWidth',2,'LineStyle','-');
            count = count + 1;
        end
    end
    output = original_image;
end
```

Annex II: Sample C# Methods

Column reconstruction

```
private void CreateDocument(int num_col)
        {
         //Create a word documents having ‘num_col’ columns
         //num_col is a number of segmented column blocks
            try
            {
                //Create an instance for word app
                Microsoft.Office.Interop.Word.Application winword = new Microsoft.Office.Interop.Word.Application();

                //Set status for word application is to be visible or not.
                winword.Visible = false;

                //Create a missing variable for missing value
                object missing = System.Reflection.Missing.Value;

                //Create a new document
                Microsoft.Office.Interop.Word.Document document = winword.Documents.Add(ref missing, ref missing,
                  ref missing, ref missing);

                //calling column creator method
                //CreateCols(num_col, document);
                document.PageSetup.TextColumns.SetCount(num_col);
                /*


                //Save the document
                object filename = @"D:\temp\result.docx";
                document.SaveAs(ref filename);


                MessageBox.Show("Document created successfully !");
                Microsoft.Office.Interop.Word.Application ap = new Microsoft.Office.Interop.Word.Application();
                document.Close(ref missing, ref missing, ref missing);
                document = null;
                winword.Quit(ref missing, ref missing, ref missing);
                winword = null;

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
```

Read and prepare recognized texts

```
private void button13_Click_1(object sender, EventArgs e)
        {
 //read the result and identify them using their name
               string result;
 //read the created document
            for(int i = 1; i<num_col+1; i++)
```

```
        {
            int num_para = 0;
            var searchTerm = i+"_P";
            var searchDirectory = new DirectoryInfo(@"D:\result\");

            var queryMatchingFiles = from file in searchDirectory.GetFiles()
                                     where file.Extension == ".txt"
                                     where file.FullName.Contains(searchTerm)
                                     select file.FullName;

            foreach (var fileName in queryMatchingFiles)
            {
                num_para++;
            }


            for (int j = 1; j < num_para+1; j++)
            {
                object xx = @"D:\\result\" + i + "_P" + j + ".txt";
                result = File.ReadAllText(@"D:\\result\" + i + "_P" + j + ".txt");

                object filename = @"D:\\temp\result.docx";


                writeWord (filename, result.ToString(), i, num_para, j, num_col);
            }
        }
```

Write recognized texts on specific column and paragraph blocks

```
private void writeWord(object filename, string result, int i, int num_para, int j, int num_col)
     {
        //write the results on specified column and paragraph blocks
        Microsoft.Office.Interop.Word.Application word = new Microsoft.Office.Interop.Word.Application();
        Microsoft.Office.Interop.Word.Document doc = new Microsoft.Office.Interop.Word.Document();
        object missing = System.Reflection.Missing.Value;

        word.Documents.Open(ref filename, ref missing, ref missing, ref missing, ref missing, ref missing, ref
        missing, ref missing, ref missing, ref missing, ref missing, ref missing, ref missing, ref missing, ref
        missing, ref missing);

        word.ActiveDocument.Characters.Last.Select();
        word.Selection.Collapse();
        word.Selection.Font.Size = 10;

           int z = word.Selection.PageSetup.TextColumns.Count;

           var breakTypes = new[] { WdBreakType.wdColumnBreak, WdBreakType.wdLineBreak};


           if(i == 1 && j == num_para)
           {
               word.Selection.TypeText("\n");
               word.Selection.TypeText(result);
           }
           if (j == num_para && i != z)
```

```
    {
        word.Selection.InsertBreak(breakTypes[0]);
    }
    else
    {
       if (j == 1)
        {
            word.Selection.TypeText(result);
        }

        else
        {
            word.Selection.TypeText("\n");
            word.Selection.TypeText(result);
        }

    }

    word.ActiveDocument.Save();
    word.Quit();
}
```