ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

Recognition of Amharic Braille Documents

By

Ebrahim Chekol

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

June, 2010

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF INFORMATION SCIENCE

Recognition of Amharic Braille Documents

By

Ebrahim Chekol

Signature of the Board of Examiners for Approval

Workshet Lamenew	_____
Chairman, Examining Board	Signature
Dr. Million Meshesha	_____
Advisor	Signature
Dr. Yaregal Assabie	_____
External Examiner	Signature

Dedicated to my whole family

Acknowledgement

First and foremost, I would like to praise the almighty ALLAH with His compassion and mercifulness to allow me finalizing this MSc. thesis.

Next, I would like to express my sincere gratitude to my advisor Dr. Million Meshesha for his patience, motivation, enthusiasm, inspiration, his great efforts to explain things clearly and simply and immense knowledge. Throughout my thesis-writing period he provided encouragement, sound advice, good teaching, good company, and lots of good ideas.

I am also grateful to Misrach Center staffs, especially Ato G/Medhin Wolde and Ato Amare Asfaw who have been sharing valuable material resources and for their regular collaboration to exchange ideas.

I wish to thank my friends: Teshome A. for his welcoming and invaluable technical support, Wale G. who has edited the draft report, Tatek A. for his regular collaboration to exchange ideas. Also I thank my friends: Abinew Ali, Alemu Jorgi and Miftah Hassen for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last two years.

Lastly, and most importantly, I wish to thank my parents, Chekol Jibril and Jemila Sirage. They bore me, raised me, supported me, taught me, and loved me.

Ebrahim Chekol

Table of Contents

List of Tables ..... **VII**

List of Figures ..... **VIII**

List of Acronyms ..... **IX**

Abstract ..... **X**

Chapter One **1**

1.**Introduction** **1**

1.1 Background **1**

1.1.1 Braille writing system ***1***

1.1.2 Optical Braille Recognition systems ***3***

1.2 Statement of the Problem and Justification **6**

1.3 Objective of the study **7**

1.3.1 General objective ***8***

1.3.2 Specific Objectives ***8***

1.4 Scope and limitation of the Study **8**

1.5 Methodology of the Study **9**

1.5.1 Literature review ***9***

1.5.2 Development Techniques ***10***

1.5.3 Testing Techniques ***10***

1.6 Application of results **11**

1.7 Organization of the Thesis **11**

Chapter Two **13**

2. **Review Literature** **13**

2.1 Introduction **13**

2.2 The Amharic Characters **14**

2.2.1 The Amharic Numerals and Punctuation Marks ***15***
2.3 Pre-Braille Methods of Touch Reading 15
2.4 Evolution of Amharic Braille 16
2.5 The Amharic Braille Code character 18
2.5.1 The Fourth version Amharic Braille ***19***
2.5.2 Amharic Braille Numerals ***21***
2.6 Braille System 22
2.6.1 Braille Writing ***23***
2.6.2 Braille Reading system ***25***
2.7 Computer Braille representation 26
2.8 Representation of Amharic Braille character 26
2.8.1 Braille characters ***26***
2.9 Interpretation of Amharic Braille character 27
2.9.1 Amharic Grade 1 Braille ***27***
2.9.2 Grade 2 Braille/contraction ***28***
2.10 Features of Amharic Braille Code 29
2.11 Features of embossed Amharic Braille 29
Chapter Three 31
3. Optical Braille Recognition systems 31
3.1 Introduction 31
3.2 Phases of Braille recognition system 31
3.2.1 Image acquisition ***32***
3.2.2 Image preprocessing ***32***
3.2.3 Segmentation ***33***
3.2.4 Feature Extraction ***34***
3.2.5 Classification ***35***
3.3 Noise Detection and removal 41
3.3.1 Gaussian Filtering ***42***
3.3.2 Median Filter ***43***

3.3.3 Adaptive median Filtering ***44***
3.3.4 Morphological operations ***44***
3.4 Review of related research **46**
3.4.1 Optical Braille Recognition for Latin Based Characters ***47***
3.4.2 Optical Braille Recognition for English and Chinese Characters ***49***
3.4.3 Optical Braille Recognition for Arabic Characters ***50***
3.4.4 Image processing Techniques for Braille writing recognition ***53***
3.4.5 Recognition of Amharic Braille ***54***
Chapter Four **57**
4. Braille Recognition Techniques **57**
4.1 Introduction **57**
4.2 Preprocessing of document images **57**
4.2.1 Histogram Equalization ***58***
4.2.2 Image Filtering ***60***
4.2.3 Image Binarization/Thresholding ***63***
4.2.4 Morphological Operations **687**
4.3 Image Segmentation **68**
4.4 Feature Extraction **69**
4.5 Classification **69**
4.5.1 Training a Neural Network ***70***
4.5.2 Neural Network Learning Laws ***71***
Chapter Five **73**
5. Experimentation **73**
5.1 Introduction **73**
5.2 General Design of Amharic OBR System **73**
5.3 Data Collection **74**
5.4 Braille Digitization **75**
5.5 Preprocessing **78**

5.5.1 Adaptive Histogram Equalization **78**
5.5.2 Gaussian filtering ***79***
5.5.3 Adaptive Median Filtering (AMF) ***80***
5.5.4 Binarization/Thresholding ***83***
5.5.5 Morphological operation **86**
5.6 Segmentation **86**
5.7 Feature Extraction **88**
5.8 Braille Character Recognition **89**
5.8.1 Artificial Neural Network Architecture ***89***
5.8.2 Neural Network Classifier ***90***
5.8.3 Braille Character Input Method for the Neural Network ***91***
5.8.4 Creating the neural network ***92***
4.8.5 Training the Neural Network ***93***
4.8.6 Testing the Neural Network ***95***
5.9 Performance evaluation of the Amharic OBR **97**
Chapter Six **100**
6. **Conclusions and Recommendation** **100**
6.1 Conclusion **100**
6.2 Recommendations **101**
Reference **101**
Appendix **107**

List of Tables

Table 2.1 Total Number of Amharic Characters by Type 15
Table 2.2 The First Version Amharic Braille Vowels 17
Table 2.3 Fourth Version Amharic Braille Characters 20
Table 2.4 Fourth Version Amharic Braille Vowels 20
Table 2.5 Fourth Version Amharic Braille code for punctuation Marks 21
Table 2.6 Ethiopic Vs Arabic numerals Braille code 22
Table 5.1 Summary of Amharic Braille documents collected for the study 75
Table 5.2 PSNR and MSE measure of Braille images 82
Table 5.3 Sample input for the neural network (a) with 6 input (b) with 12 input node 91
Table 5.4 Sample Amharic characters with the same ASCII value 92
Table 5.5 Network performance for the trainingset 94
Table 5.6 Error type and list in validating the network 94
Table 5.6 Performance rates for test dataset 98

List of Figures

Figure 1.1 Dot positions in a Braille cell 2
Figure 1.2 Phases of Braille recognition 4
Figure 2.1 The Genetic Structure of Amharic Scripts 13
Figure 2.2 List of Amharic core characters with seven orders 15
Figure 2.3 Braille code representation for character "h" 20
Figure 2.4 Representation of Braille cells with 6 and 8 dots, and recto and verso dot 22
Figure 2.5 Braille Dot Dimension 26
Figure 3.1 Braille Cell with Dots Position 34
Figure 3.2 Architecture of feed-forward Neural network 39
Figure 3.3 Image areas (grey levels) corresponding to protrusions [45] 47
Figure.3.4 Braille image with Global grids (the result of the first phase) 55
Figure 3.5 Segmented Braille image (the result of the second phase) 55
Figure 5.1 Block diagram of the Amharic Braille recognition system 72
Figure 5.2 Braille document image (left with gray level scale and right with color) 76
Figure 5.3 Image type by noise level 77
Figure 5.4 Distribution of Pixels Using Histogram 79
Figure 5.5 MATLAB code for Gaussian filter 80
Figure 5.6 The MATLAB code for adaptive median filtering 81
Figure 5.7 Results of the filtered image 81
Figure 5.8 The result of aaptive histogram equalization 82
Figure 5.9 Thresholding results. 85
Figure 5.10 MATLAB code for thresholding Braille Image 85
Figure 5.11 The result of Morphological operation 876
Figure 5.12 Braille dots with a) Incorrectly segmented b) correctly segmented 87
Figure 5.13 Mesh grid for high level noise ….. 88
Figure 5.14 General Architecture of Braille Recognition (taken from Teshome[51]) 90
Figure 5.15 MATLAB code for creatong the neural network 91

List of Acronyms

AAU	Addis Ababa University
AMF	Adaptive Median Filtering
ANN	Artificial Neural Network
ASCII	American Standard Code for Information Interchange
EABS	Ethiopian Association for Blind Society
IEF	International Eye Foundation
MATLAB	MATrix LABoratory
MLP	Multi-Layer Perceptron
OBR	Optical Braille Recognition
OCR	Optical Character Recognition
UNESCO	The United Nations Educational, Scientific and Cultural Organization

Abstract

There are more than half million visually impaired people in Ethiopia consisting of students, teachers, artists, lawyers, and also employees who have significant contribution in politics, religious, economics and social affairs of the society. For these people Braille is the means for codifying their knowledge. However, most of their work still remained in their Braille and of course accessed only by those who read and/or write Braille. Since 1924, there are a significant number of old Braille documents produced and used by visually impaired society throughout the country. From this, very insignificant number of Braille has actually reached to the vision society creating communication gap between the vision and visually impaired people. In addition to these, at present, blind students attend regular schools together with their sighted peers in the elementary, secondary and tertiary levels. However, it is difficult, among others, to take quizzes and examinations without the help of others (vision students) in reading and writing, and also in submitting assignments for their teachers. In general, there is no means of written communication between blind and sighted people.

In this study an attempt is made to explore the possibility of developing an OBR (optical Braille recognition) system for real life single sided Amharic Braille documents. The system is implemented using artificial neural network for 238 Amharic characters, 10 Arabic numerals and 19 punctuation marks. Gaussian filtering with adaptive histogram equalization and morphological operations are used so as to detect and remove the noises in the real life Amharic Braille documents. The noise detection and removal, and thresholding algorithms are integrated with the previous algorithm implemented for recognition of clean Amharic Braille documents. The present system achieves an accuracy of 95.5%, 95.5%, 90.5%, and 65% for clean, small level noisy, medium level noisy and high level noisy Braille documents respectively. This research approach to control noise in the real life Braille document to a great extent, however, there is some addition and deletion of dots.

Chapter One

Introduction

1.1 Background

According to Teshome [51] citing International Eye Foundation (IEF),there are 500,000,000 (five hundred million) disabilities in the world, out of which more than 45,000,000 (forty five million) are blind people in the world. The vast majority of which are live in Africa. In Ethiopia, there are over 500,000 (five hundred thousand) blind people [5][51]. Therefore, the government and the society are responsible to improve the life of them. Visually impaired people are parts of the society and plays significant role to the society[2]. So, in order to play their role they may communicate to the world by means of speaking, touching and/or hearing. The most famous and widely used means of written communication is Braille systems[11].

1.1.1 Braille writing system

Braille is the only scripts that blind people use to communicate with each other[11]. The Braille format used by sight impaired people today was invented by a visually impaired Frenchman, Louis Braille, in 1824 and published in 1829 [11]. After the invention, visually impaired people changed the way people approach education. Before the invention of Braille, they didn't have many opportunities for education or employment. The few existing schools for the blind were more like residential institutions, teaching basic trade skills while ignoring reading, writing and other academic studies. Braille changes all that by giving blind people an efficient method for communication and learning. It is a system of tactile reading and writing for visually impaired people. Each Latin character or cell is made up of 6 dot positions (2 columns by 3 rows), 63 possible characters and a space (none dots) are available by using any one or a combination of dots [18] [42].The position of the dots affects the meaning of the cell. These dots are numbered one, two and three on the left side of the cell, and four, five and six on the right side, as shown in figure 1.1. A typical Braille page is 280×292 mm with 40 characters long and 25 lines [2]. The

horizontal and vertical distance between dots in a character, the distance between cells representing a word and the distance between two lines are specified by the Library of Congress [2]. A Braille dot is raised by approximately 0.02 inches (0.5 mm); the horizontal and vertical distance between the dots is approximately 0.1 inches (2.5 mm). The blank space between dots on adjacent cells is approximately 0.15 inches (3.75 mm) horizontally and 0.2 inches (5.0 mm) vertically.

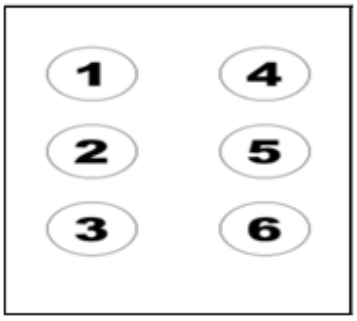


Figure 1.1 Dot positions in a Braille cell

Braille pages are also thicker and heavier than ordinary paper, and they have to be bound in a loosen format so that pages can lie flat and people can reach the cells near the book's binding. This leads to relatively bulky books. The Braille has been adopted to Amharic, English, Arabic, and other different languages in the world, and also used for mathematical and musical notation [2] [38].

Braille characters can represent virtually any language, regardless of how that language is written. In Chinese Braille, the characters represent the sounds that make up the spoken language. Amharic Braille uses cells to represent the letters of the Amharic alphabet and contractions specific to that language [11] [51].

In Ethiopia the first school for the blind was established in 1924 in Dembidolo by Ruthle in collaboration with voluntary and nongovernmental organizations [15][51]. They start to teach handcrafts and Amharic writing system from "h" to "p" which was created on wooden board with nail. However, its growth was disrupted by the Italian invasion in 1935. Later, beginning

the 1950's other special schools were opened in different parts of the country (such as Bakko, Sebeta, Soddo, Ghimbi, Shashemene, Wolayita, and Dire Dawa) with the cooperation of the Ethiopia government, by different charity organizations. Currently, there are totally six blind boarding schools in the country Bakko, Sebeta, Gonder, Shashemene, Wolayita, and Mekele (the recent one).

Thus, since 1924 there have been massive Braille documents produced and used by visually impaired society throughout the country. However, very insignificant number of these Braille has actually reached to the vision society creating communication gap and generation gap between the vision and visually impaired people. The condition even would be very serious as the trend of the world is towards information society where information is a vital resource [51]. In addition to these, at present, blind students attend regular schools together with their sighted peers in the elementary, secondary and tertiary levels. However, it is difficult, among others, to take quizzes and examinations without the help of vision students in reading and writing, and also in submitting assignments for their teacher. In general, there is no means of written communication between blind and sighted people.

Research in Braille recognition would have significant contribution in facilitating smooth communication between visually impaired and vision society in every dimension of life. This in turn facilitates the efforts done towards improving the life of visually impaired society.

1.1.2 Optical Braille Recognition systems

Verbal and written communications are integral components of human society and have been transformed by the development of the respective communication devices. Through the dramatic development in information processing devices, the need and access to the digitized printed information items become possible by means of Optical Character Recognition (OCR) software [59].

Optical Character Recognition technology is becoming a field of research in pattern recognition, artificial intelligence and machine vision [51]. OCR system involves reading scanned bitmap images of machine-printed or handwritten text and translating document images into a form that the computer can manipulate (for example, into ASCII or Unicode).

There are two types of OCR systems with respect to the way the input is provided to the system [59]: on-line OCR and off-line OCR. In the online OCR, the recognition task is performed parallel with the writing process, where as with off-line OCR the recognition process is performed after the whole text is made available to the OCR engine.

There are many factors affecting an OCR system. Some of them include: the mode of writing (handwritten or using Braille printer), the types of Braille paper (gray or light yellow; single side or double side), the presence of artifacts like extraneous markings, and the resolution and lighting during scanning [14].

OCR technology has shown great promise in providing access to textual information and making accessible and computable for visually impaired people. Optical Braille Recognition (OBR) system allows visual people to read volumes of handwritten or using printer Braille documents with the help of flatbed scanners and OBR software. According to Teshome [51] the Braille documents can exist in paper or plastic, gray or color formats. As shown in figure 1.2, the recognition of Braille documents consist of six phases [12]: Scene construction, Image acquisition, Pre-processing, Segmentation, Feature extraction and Interpretation.

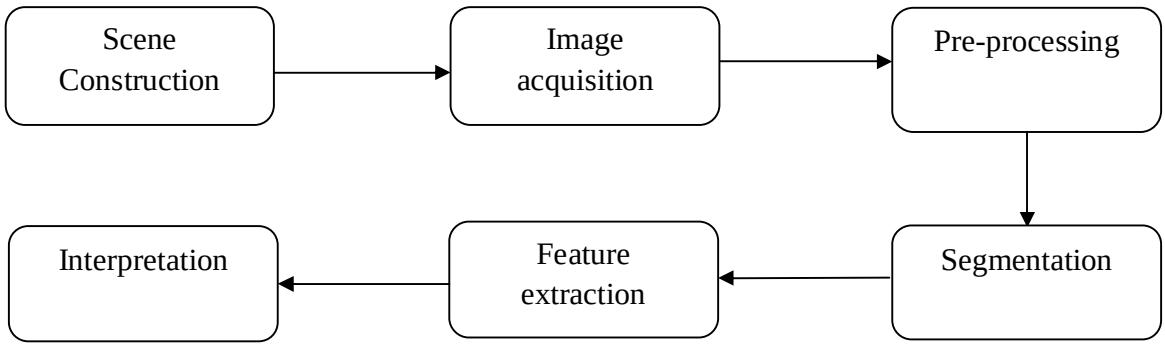


Figure 1.2 Phases of Brille Recognition

Scene construction/initial preparation: This is the first phase of the system. The aim of this phase is maximizing the use of prior knowledge of the scene, through exploiting existing knowledge, and reducing the problem of image analysis in any possible way. This can be achieved by imposition of effective constraints such as illumination. This is because of the fact that most Braille papers are embossed on both sides of a white paper, and the dots are embossed with a sphere shape.

Image Acquisition- The second phase involves image acquisition, which is the process of converting the Braille image into digitized image array by using digital camera or scanner. Scanners are capable of producing the image in different formats. The most common of these is the bit map (.BMP) format [12] [59].

Pre-Processing: This phase performs the following operations: image binarization, noise filtering and edge enhancements. The noise can be generated due to the nature of the paper (paper or plastic), converting color Braille into gray, lack of good storage, repetitive use of the Braille or during scanning in the image acquisition process. So, by applying noise detection and removal techniques there is a need to minimize the noise in the Braille. The objective of edge enhancement is to sharpen the fine details of the image that has been blurred.

Segmentation: It is a process that seeks to decompose an image of a sequence of Braille dots into sub-image of isolated symbols (dots). The identification process first separate the foreground (content) and background from the acquired images. Then, organize a set of pixels (dots) in the foreground to form each half character, character and words. This operation also separates the front-faced dots and back-faced dots.

Feature extraction: It is the representation of the Braille dots to extract the Braille dots from the binarized image and group them to cells. As the shape of a Braille dot is known, the boundary points of the dots can be obtained by the boundary based Chain Code algorithm. As the coordinates of the dots are detected, the diameter of the dot can also be deduced. As the dots are embossed on both sides of a page, the dots on the front-face and on the back-face must be separated before further processing.

Interpretation: The work of this phase is to group the Braille dots into cells and converts them into their correspondences languages by determining the centriod between dots and the four possible neighbors. Each word is then checked against an English dictionary. If the word cannot be found, then the word with the highest percentage of similarity will be selected and will be highlighted for later edition.

1.2 Statement of the Problem and Justification

There are more than half million visually impaired people in Ethiopia [5]. These are students, teachers, artists, lawyers, and also employees who have significant contribution in politics, religious, economics and social affairs of the society [51]. For these people, Braille is the means for codifying their knowledge. However, most of their work still remained in their Braille and of course accessed only by those who know, read and write Braille.

According to Nebyuluel [64], unless there is a smooth information flow from visually impaired people to vision and vice versa, people do not have the necessary information about visually impaired once. Hence, the work of many visually impaired people would have remained hidden [64]. This would create a wide generation gap between the visually impaired and sighted society.

It is true that in the world printed documents are dominant and the information contained in them is more useful in electronic format. In Ethiopia, there are massive collection of Braille documents that have been produced and found at different parts of the country; typically at AAU Kennedy library Braille documentation, Misrach Center, Sebeta visually impaired school, German Church visually impaired school, Ethiopian Association for Blind society (EABS), Entoto Blind people school and also in different regions of the country [51]. Digitizing Braille documents necessitates scanning and applying OCR for translating them into machine readable format.

The use and application of OCR systems are well demonstrated for printed and Braille documents of many languages in the world. Recognition systems with high accuracy rate are commercially available for Latin scripts and some of the Oriental languages. However, for those

languages in Africa with their own scripts, much attention is not given for developing robust OCRs that successfully recognize diverse printed text images [35].

In case of Ethiopia, though designing Amharic OCR was started in 1997, most of the research are targeted towards machine printed document recognition. These works include Worku (1997), Ermias (1998), Dereje (1999), Berhanu (1999), Nigussie (2000), Million (2000), Yaregal (2002), Mesay (2003), Wondossen (2004), and Teshome (2009) are worth to mention.

As far as the researchers knowledge, the application of optical Braille technology for Amharic Braille documents in Ethiopia has been conducted in 2009 by Teshome [51]. His work is on the recognition of Amharic Braille documents. To this end, Teshome[51] used a global threshold for binarization, mesh-grid technique for segmentation, context based analysis for feature extraction and feed forward artificial neural network for classification. He tested the developed algorithm on clean Braille documents and attained 92.5% accuracy.

However, in real life application of Amharic Braille recognizer, it is essential that the system recognizes a noisy Braille because the Braille exist in different areas are repeatedly used, bend and scratch [51]. These noises and image impurity affect mainly the results of the binarization processes. Most of the noises are scattered type but it increases around the edge of the documents. These all noises and image impurity significantly affect the subsequent recognition processes. As a result Teshome [51] strongly recommended the need to integrate noise detection and removal algorithms for the recognition of noisy Braille documents. A preliminary experiments on real life documents exhibited near zero performance.

The aim of this study is therefore, to explore different noise detection and removal algorithms in order to enhance the performance of the Amharic Braille recognizer in real life Braille document images.

1.3 Objective of the study

The general and specific objectives of this research work are described here under.

1.3.1 General objective

The general objective of this study is to explore the possibility of designing Amharic Braille recognition system that tolerates noises or degradations in real life Amharic Braille document images and achieves better performance.

1.3.2 Specific Objectives

In order to achieve the general objective of this study, the following specific objectives are drawn.

- Review previous works related with Optical Braille Recognition for conceptual understanding of theories and principles.
- Examine and identify attributes and features of Amharic Braille characters.
- Select the best noise detection and removal algorithms for reducing artifacts in Braille documents.
- Develop a program for the selected noise removal algorithm and integrate to the Amharic Braille recognizer.
- Generate training and test datasets required for training the classifier (neural network) and testing its performance.
- Evaluate the performance of the Amharic recognizer and report the findings of the research.
- Make conclusions and forward recommendations for future research.

1.4 Scope and limitation of the Study

This study is a continuation of previous works in the area. The main purpose is to improve the performance of the recognizer by adopting an approach that reduces the effect of noise during recognition. To this end, various noise detection and removal algorithms are investigated and

suitable one is finally integrated to Amharic OBR. This is followed by training the classifier with training datasets and experiment with real life scanned Braille pages to report the accuracy of the system.

The study plans to train the OBR system on all characters, numerals and punctuation marks in the Amharic writing system. However, because of time constraints the current study is limited to basic Amharic characters and not considered extended Amharic Braille characters, Ethiopic numerals and some punctuation marks. Besides, the Amharic Braille characters are defined with one, two and three cells. However, Braille characters with three cells are not included in the study for the same reason.

1.5 Methodology of the Study

According to Worku [60], Methodology provides an understanding of how a research is conducted and organized in order to obtain information that is helpful to understand concepts, theories and principles for developing the design of a research. To achieve the aim of the present research the following methods are followed.

1.5.1 Literature review

Past researches are important to share their experience for identifying what has been done in the area, how characters are written on Braille and which approaches fit best for specific problem domain. Hence, previous works in Braille recognition, in general, and Amharic Braille recognition system, in particular, are reviewed from books, journals, proceedings and the Internet to put light on the way towards developing Amharic Braille recognizer. Besides these, the review is used to understand the concept of noise detection and removal techniques, mapping techniques and algorithms so far identified and have been used to address related problem for the recognition of machine printed and Braille documents.

1.5.2 Development Techniques

Compared to other conventional computer languages, MATLAB has many advantages for technical problem solving [10]. First: the language is so easy to use. Second, it is ideal for rapid prototyping of new programs. Third, it is supported on many different computer systems, providing a large measure of platform independence. Fourth, it comes with an extensive library of packaged solutions to many basic technical tasks. Lastly, it has many integral plotting and imaging commands. Because of all the above reasons, the preprocessing tasks are performed in this study using MATLAB.

The previous Amharic Braille recognition development has been done using Microsoft Visual C++ programming language, which lower the burden of coding for the researcher. Moreover, Microsoft Visual C++.Net programming language provides the facility of using classes, which enables the project to use objects, which again closely model the real world objects in attributes and behaviors. Thus, the programming language, which is used to integrate the preprocessed image to segment and extract the features of Braille image, is Microsoft Visual C++.Net.

Artificial Neural Network is used for the purpose of classification, which is categorized as one of the robust approaches to deal with uncertainty and noisy input data [51]. For the classification purpose, MATLAB Neural Network toolbox is used mainly due to its availability and ease of constructing and training the network. In addition, MATLAB is powerful in dealing with matrices which is the way characters are represented for recognition purpose.

1.5.3 Testing Techniques

The sample real life Braille documents are first scanned using a flat-bed scanner and preprocessed. Then, the preprocessed Braille is submitted to the recognition algorithms and it generates the extracted features of the Braille document. Based on these extracted features, the performance of the trained neural network with Amharic characters, Arabic numerals and some of the Amharic punctuation marks is tested with a total of 800 Braille characters (200 for each

level of noise, i.e. clean, small noise, medium noise and high noise level). Based on the test results, the accuracy is measured by comparing the expected results with the system output.

1.6 Application of results

The outcome of this study have significant contribution to bridge the gap between the sighted and visually impaired societies specially, teachers and students in schools and universities, for those who work with blind people and do not read Braille, computerized Braille libraries, and public organizations communicating with blind individuals.

With this automatic Braille character recognition method, written communication between blind people, sighted people and other impaired people are facilitated.

1.7 Organization of the Thesis

The thesis is organized into six chapters. The first chapter of the thesis includes the background, the statement of the problem, objective, scope and limitation, and discusses the methodology used to accomplish the research.

The second chapter presents review of literatures in Amharic Braille system, features of Braille character and characteristics of Braille recognition. The third chapter deals with review of related research work in Braille problem domain along with the basic methods of preprocessing techniques of Braille images analysis. The section also highlights basic components and issues in application of artificial neural network. The fourth chapter discusses the techniques and algorithms used to develop the current systems.

Chapter five deals with the experimentation activity undertaken to implement the methods and techniques described in chapter four. The fifth chapter also shows the success and difficulties faced while trying to implement the technique to the problem domain of interest is presented. It also presents the recognition rate achieved for different test cases after the design and training of

the neural network classifier. Finally, concluding remarks and recommendations for future research are forwarded.

Chapter Two

Review Literature

2.1 Introduction

Writing is a means of communication and presenting historical information for future use. The first writing system was in the form picture writing. The Egyptian used picture writing to draw or paint on monuments, walls or rocks in 3000 BC [14]. Bender et al. [4] noted that all Middle-Eastern scripts come originally from the Egyptian picture writing (i.e. Hieroglyphics). The Geez scripts are derived from the south Arabian (i.e. Sabaean) scripts [54]. Ethiopic script is one of the ancient alphabets in the world which is used to write in languages such as Geez, Amharic, Tigrigna, etc [35]. The present writing system of Amharic is taken from Geez evolved from Sabaean language which was brought to the highlands by immigrants from south Arabian in the first A.D. The general genetic structure of Ethiopic Script is given in figure 2.1 [4] [61].

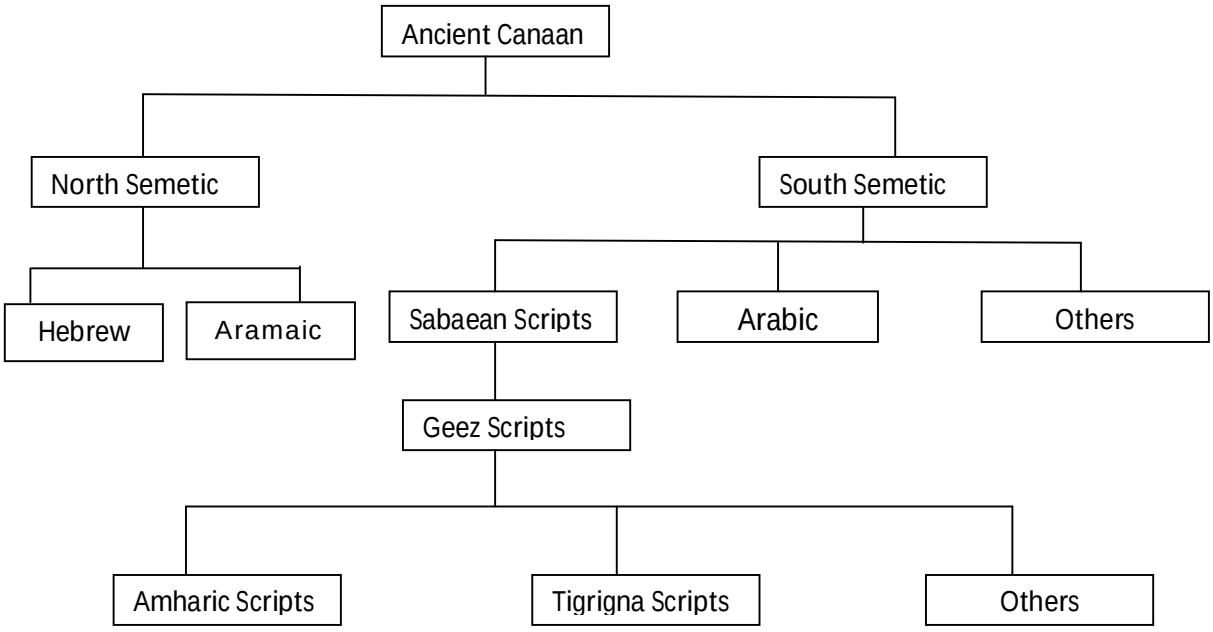


Figure 2.1 The Genetic Structure of Amharic Scripts

Amharic is the second most spoken Semetic language in the world next to Arabic [35]. Amharic language is mainly spoken in Ethiopia and Eritrea. Amharic is used as an official language in Ethiopia since the 14th century. It is also the working language of the federal government of Ethiopia and also non-government organizations as well as private institutions. Amharic language has its own scripts. Understanding the characteristics and distinct features of the script has used for feature extractions phases of the Braille recognition systems.

2.2 The Amharic Characters

Amharic script has also evolved through the centuries and has undergone many transformations in structural shape and an increase in the number of symbols. This happens on the one hand, by modifying the original Sabean characters and, on the other hand, by adding other necessary ones in the writing. The need for such transformation through time is due to [54]: (i) the tendency towards round forms, (ii) the changed direction of writing, and (iii) the turn of some characters by ninety degrees.

Amharic language writing system is composed of a total of 238 characters among which 33 are the 'core' characters and one is 'special' character. Each of them occurs in 7 forms (orders); one basic form and six non–basic forms representing syllable combinations consisting of a consonant and following vowel. The non-basic forms are derived from the basic forms by more or less regular modifications (see figure 2.2). For example the second order characters are formed mostly by attaching strokes to the right of the character.

In addition to the 231 characters set, the 33 core and its 7 forms, there are over forty others which contain a special feature usually representing special character (ሃ) - which has also seven forms, and has used to represent the 'v' sound of words in Latin-based languages, labialization, numerals, and punctuation marks. These bring the total number of characters in the script to 310 [4] [35]. Table 2.1 shows summary of the number of characters in each group.

Figure 2.2 List of Amharic core characters with seven orders. Characters in the first orders are basic and others are non-basic forms.

2.2.1 The Amharic Numerals and Punctuation Marks

Amharic writing system does not have a symbol for number zero- because the script was developed before the scientific significance of this number was realized. Ermias [19], the writing system has no symbols for negative numbers, decimal point, and mathematical operators. Instead of those numerals, the Hindu-Arabic numerals and Latin mathematical operators are used for computational purpose. The Ethiopic numbers when put in numerals form are written a top and bottom frame. The forms are used in order to uniquely identify the numerals from the textual character set. The old South Arabic Monumental scripts regularly employ a vertical stroke as a

word separator. This too was borrowed and applied after every single word in an Ethiopic text as : The sign & and @ are used as a comma and semicolon within a sentence, respectively.

No.	Types of Amharic Characters	No. of Characters
1	Core Characters (33 X 7)	231
2	Special Character (1 X 7)	7
3	Labialized Characters	44
4	Punctuation Marks	8
5	Numerals	20
	Total	310

Table 2.1 Total Number of Amharic Characters by Type

2.3 Pre-Braille Methods of Touch Reading

Braille was not the first and only means or methods of touch reading [11]. The earnest desire of the blind to find access to literature and of their sighted friends to open the door for them, led to many experiments in a variety of media. Even after the invention of Braille's, other forms of embossed symbols were planned and used- some employing dots and lines, others having the form of simplified roman capitals.

Typical of the early efforts of this nature is the blind Arab professor, Zain-Din Al Amidi in the 14^{th} century, a method by which he identified his book. He knew the place occupied by each book, and on receiving a request for information could find the exact volume without assistance, and also knew the place of each book [11].

The other efforts to create a script was made about 1517 by Francisco Lucas, of Saragossa, Spain, who arranged a set of letters embossed in thin tablets of wood. This was bought to Italy in 1575 and improved by Rampansetto of Rome, who used large blocks. Both systems were failed due to the difficulty of reading them [11].

When Valentin Hauy founded his school in Paris in 1784 [11]. Hauy at once set about embossing books and experimented with certain types. Embossed literature had been invented, but the old difficulty of a script which could be easily read by touch remained. It was the revolution of this by Louis Braille in 1829 which completed the system the blind has been using today. The Braille system was universally adopted and in use after fifty years of the invention, and in the mean time numerous other forms of embossed type were devised on different countries like Great Britain, America and other European countries [11].

2.4 Evolution of Amharic Braille

Braille was introduced in Ethiopia, after 100 years of the invention, in 1924 by a missionary Ruthle [64]. Amharic Braille was revised three times between the years 1945-1949, by taking English Braille as a reference [64]. In Ethiopia the first school for the blind was established in 1924 in Dembidolo by Ruthle in collaboration with voluntary and nongovernmental organizations [15] [51]. They start to teach handcrafts and Amharic writing system from "A" to "p" which was created on wooden board with nail. However, its growth was disrupted by the Italian invasion in 1935. Later, beginning from the 1950's other special schools were opened in different parts of the country (Bakko, Sebeta, Soddo, Ghimbi, Shashemene, Wolayita, and Dire Dawa) with the cooperation of the Ethiopia government and different charity organization.

The first Amharic Braille was comprehensive and complete. It contains all Amharic characters from "A" to "p" including a character that has similar sound (i.e. the three "A"- A H x, the two "\"- \ P the two "Ë"- Ë Ð and the two "†"- † –), punctuation marks, Amharic numerals, and musical symbols taken from other languages, and also design their own mathematical symbols. In addition to these, all the seven forms of the Amharic characters was used vowels with basic character representations (see table 2.2) [62] [64].

1st form	2nd form	3rd form	4th form	5th form	6th form	7th form
1:4	2:5	3:6	1:5	2:4	2:6	3:5

Table 2.2. The First Version Amharic Braille Vowels

When we see the characteristics of the first Amharic Braille there is a similarity of representing consonants and vowels with the English characters i.e. the English consonant 1:4 was used as the 1st form Amharic vowels, the English 6th sound vowels 1:5 was used as the 4th form vowel and also the English vowels 1:1 characters was used as Amharic consonant. Therefore in order to solve the problems of the first version the world Association of the Blind society made revision on Amharic codes in 1945 E.C. This revision was arranged to match with the English Braille by representing each sound once.

The revision was made in two parts [62]: the first was made to eliminate the redundant Amharic characters which have similar sound as mentioned above. The second change was made on the numeral parts that replace the Ethiopian numerals with the English numerals. Since this time, the modified Braille was distributed to all blind school in Ethiopia by the American missionary and the teachers were announced to adjust their system with the new one [62] [65]. This version had the following characteristics: the first and the sixth forms of the characters were represented without vowels and the rest of them were used vowels but the 6th form has no relationship with the rest of the characters.

Later in 1949 E.C., the Amharic Braille was subject to improvement for the third times when Sir Cluth Mecanize, the chair person of the world blind society, came to Ethiopia for visit [64]. This time the improvement was made to further reduce the variant of Amharic characters. When we compare it with the second revision, this version had added vowel for the first form. The sixth form characters had similar form with the regular characters and had no vowels and changes the dot from three to five for the character "†". This version was used for a long time until another attempt was started to modify in 1980's E.C [62] [65].

Once again after an extended study for 12 years the 4th version of Amharic Braille was published in 1995 by Ethiopian Blind Society Association that was given the mandate to assess many issues to make the appropriate modification on the existing Braille code [62].

The Amharic Braille subjected to many improvements and change as it was designed by the American does not consider many issues set by the world blind association**.** Some of these rules and procedures which should be considered in adopting Braille to local languages are the following [64]:

- Different languages should have similar Braille code. This is not to destroy the cultural nature of the literature of the language and distort the sense and meaning, rather this is to simplify the international publication, to avoid difficulties for those blind who are interested to study different language, and to create common understanding among blind society in the world.

- As per the rule, different languages that are the same source or relatively have similar in sound, in character number and type should have similar Braille code structure. For instance, English and European languages like French, Latin, German, etc do have similar foundation in literature. Moreover, they are similar in the number of characters, sounds, thus, have similar Braille code. Likewise, the Amharic Braille code should have relatively similar with Arabic, and other Semetic and Kaham languages. However, this does not imply, those languages that are different should have similar Braille code.

As Nebyeleul [64], the Amharic language has its root from Sabaean and Arabic and also from different Sematic and Kahm language. Hence, according to the rule of the world blind society association, the Amharic Braille should consider those Braille code.

2.5 The Amharic Braille Code character

Braille has four parts [62]: Characters, punctuation marks, mathematical and Musical symbols. Braille by its behavior has many problems. Some of the problems of the Braille are: I) due to

limited Braille dots, there is similarity of representing characters, II) Braille does not use for legal purpose. Grade 2 Braille can solve the space requirement of writing (volume of the document) and also increases the speed of reading Braille. Most languages such as English, Arabic, etc has grade 2 Braille but the Ethiopic languages such as Amharic, Oromifa and Tigregna do not have it. The other problem of the Amharic Braille compare to other language is that Amharic Braille has more number of characters and also has characters with similar sounds.

The Amharic Braille was a problem to acknowledge and perform fast and easy reading. Because, the previous three version were focused only on the Amharic Braille characters without considering the other parts of the Braille, i.e. mathematical symbols, punctuation marks, musical symbols and other Ethiopic languages like Tigrigna, Oromiffa and others [65]. The last amendment, fourth version, was made in 1995 E.C. This version has been used for long period of time throughout the country. This part of the study gives detailed description of the fourth version of Amharic Braille code that is in use currently for writing, which is also the focus of the current study. The other three versions of Amharic Braille are given in appendix I, II, and III.

2.5.1 The Fourth version Amharic Braille

Discussion issues for the fourth version Amharic Braille committee were [62]: whether to create equal number of Braille code that match to printed character or not, means of substituting similar sound characters, means of preparing grade to Braille, Geez numbers, use of yard musical symbols, and punctuation marks. The committee discussed for a long period of time using the third version as a reference and then, they reached to the following conclusions: the number of Braille code should be equal to printed character, modify the punctuation by taking most from the English, in order to create grade 2 the grade 1 should first get legal recognition. Yared (Ethiopic) musical symbols will be discussed with professional, and for the case of mathematical symbols, it use either the Americans or English systems.

Accordingly, Braille dot combination has been approved and distributed for use by different blind institutions as shown in table 2.3. For examples, figure 2.3 present the appearance of character “h” in Braille:

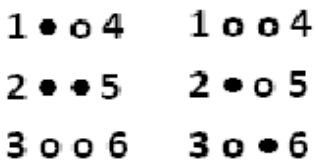


Figure 2.3 Braille code representation for character "h"

1st form (character)			6th form (character)		1st form (character)			6th form (character)	
		Vowel					vowel		
ሀ	1:2:5	2:6	ሀ	1:2:5	ኸ	2:3:6	2:6	ኽ	2:3:6
ለ	1:2:3	2:6	ል	1:2:3	ወ	2:4:5:6	2:6	ው	2:4:5:6
ሐ	*1:2:6	2:6	ሕ	*1:2:6	ዐ	*1:2:5:6	2:6	ዕ	*1:2:5:6
መ	1:3:4	2:6	ም	1:3:4	ዘ	1:3:5:6	2:6	ዝ	1:3:5:6
ሠ	2:3:4	2:6	ሥ	2:3:4	ዠ	3:5:6	2:6	ዥ	3:5:6
ረ	1:2:3:5	2:6	ር	1:2:3:5	የ	1:3:4:5:6	2:6	ይ	1:3:4:5:6
ሰ	*1:4:5:6	2:6	ስ	*1:4:5:6	ደ	1:4:5	2:6	ድ	1:4:5
ሸ	1:4:6	2:6	ሽ	1:4:6	ጀ	2:4:5	2:6	ጅ	2:4:5
ቀ	1:2:3:4:5	2:6	ቅ	1:2:3:4:5	ገ	1:2:4:5	2:6	ግ	1:2:4:5
በ	1:2	2:6	ብ	1:2	ጠ	2:3:4:5:6	2:6	ጥ	2:3:4:5:6
ተ	2:3:4:5	2:6	ት	2:3:4:5	ፈሐ	1:4	2:6	ፍ	1:4
ቸ	1:6	2:6	ች	1:6	ጸ	2:3:5	2:6	ጽ	2:3:5
ኀ	*1:5:6	2:6	ኅ	*1:5:6	ፀ	1:2:3:4:6	2:6	ፅ	1:2:3:4:6
ነ	1:3:4:5	2:6	ን	1:3:4:5	ጸ	*2:3:4:6	2:6	ጽ	*2:3:4:6
ኘ	3:4:6	2:6	ኝ	3:4:6	ፈ	1:2:4	2:6	ፍ	1:2:4
አ	1:2:3:5:6	2:6	አ	1:2:3:5:6	ፐ	1:2:3:4	2:6	ፕ	1:2:3:4
ከ	1:3	2:6	ክ	1:3	ቨ	*1:2:3:6	2:6	ቭ	*1:2:3:6

* Symbol indicates Braille code change made on this new version.

Table 2.3 Fourth Version Amharic Braille Characters

Vowel	2:6	1:3:6	2:4	1	1:5	Not apply	1:3:5
Character position	1st	2nd	3rd	4th	5th	6th	7th

Table 2.4 Fourth Version Amharic Braille Vowels

As shown in table 2.4 the sixth variant of the characters are decided to continue without vowel attachment as it has been used in the third version of the Braille. Regarding punctuation mark, codes in Table 2.5 are designed. The codes are designed using one, two or three cells. The complete list of punctuation mark is presented in appendix IV.

Punctuation	Braille code
.	3
:	6 and 3
:-	2:5
/	5 and 2
‘	4 and 1
-	3:6
½	2
¿	2:3
“	2:3:6
“	3:5:6

Punctuation	Braille code
'	3:5:6 and 3
?	2:3:6
=//=	6 and 3
X	1:3:4:6
$	4:5
::	2:5:6
!	2:3:5
…	3 , 3and 3
( )	2:3:5:6
[	6 and 2:3:5:6

Punctuation	Braille code
]	2:3:5:6 and 3
*	3:5 and 3:5
__	4:6 and 4:6
→	2:4:6, 2:5 and 2:5
←	2:5,2:5 and 1:3:5
↑	4:5:6 and 1:5
↓	4:5:6 and 3:5

Table 2.5 Fourth Version Amharic Braille code for punctuation Marks [51]

Most of the punctuation marks are not new, however much focus had not been given so far. Thus, in the new Braille they are included as most of these symbols feel new by many blind society, according to the revision committee report [62]**.**

2.5.2 Amharic Braille Numerals

The Braille revision committee also made change on the numerals part of the Braille code. The need to reexamine this part is that, unlike the Arabic numerals, in Ethiopia there are different publication that uses the Ethiopic numerals such as newspapers and religious books. So, in order to show the numerals are Ethiopic, changes were needed on this part of the Braille. Accordingly, it is designed to use the whole dot (1:2:3:4:5:6) to indicate the numbers are in Ethiopic. Then, this Ethiopic numerals indicator mode followed by the numerals which is used for Arabic numerals are adopted in the fourth version [62]. Except the numerals translation mode 3:4:5:6 (for Arabic) and 1:2:3:4:5:6 (for Ethiopic) the same dot combinations are used as given in Table 2.6.

፩	፪	፫	፬	፭	፮	፯	፰	፱	፲
1	2	3	4	5	6	7	8	9	0
⠁	⠁⠃	⠁⠉	⠁⠙⠑	⠁⠑	⠁⠋⠉	⠁⠋⠙⠑	⠁⠋⠑	⠃⠉	⠃⠙⠑

Table 2.6 Ethiopic Vs Arabic numerals Braille code

2.6 Braille System

Braille is a system of embossed (raised) signs, which are formed by six (or eight) dots arranged and numbered as in Figure 2.4. Eight dot Braille is limited in use for computer application area and is practically applied in the display of text attributes. Six dots Braille is the widely used one and hence it is also considered in this study [27].

To facilitate the description of individual symbols, the dots are conventionally numbered, those of the left-hand column being numbered 1-2-3 from top to bottom and those of the right-hand, 4-5-6. The six dot Braille cells are classified according to their dot patterns as upper cells (1-2 and 4-5), lower cells (2-3 and 5-6), bottom row cells, indicator cells, and full cells [27][34].

Braille is applied to various practical purposes some of these are the expression of music, mathematical and musical symbols, the making of faces of watches, meters, gauges, thermometers and playing cards, and adapted too, to the outlining of geographical maps and plans of cities and buildings [4].

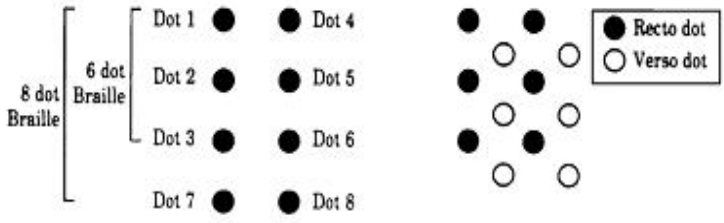


Figure 2.4 Representation of Braille cells with 6 and 8 dots, and recto and verso dot

2.6.1 Braille Writing

Braille is essentially a six-bit binary system – there are six dots, and a dot is either raised or not raised – meaning there are exactly 64 (2^6) possible characters, consisting of 63 dot combinations, and the blank space where no dots are raised. The 63 symbols derived from the six-dot Braille cell form the basis of all Braille writing systems worldwide, although the exact meaning of each symbol is language-specific [46].

Braille writing devices have evolved into some very sophisticated pieces of equipment [55]. Lois Braille's slate consisted of two parts; the top part was a two-line metal guide with the familiar cell openings, and the bottom part a very thick piece of wood with only horizontal grooves instead of the now familiar pits. The Braille slate- a writing device that allows an individual to press on paper with a sharp instrument to form the Braille dots-is practical and well suited to a variety of purposes as well as being inexpensive, portable, and requiring little, if any, repair. Because they are fairly simple to make, Braille slates are still the primary Braille-writing tools in the world today. Yet, writing with the standard Braille slate can be somewhat confining. It is not possible to vary the size of the letters, to write between the lines, or too scribble in the margins, as those who write print so often do [55]**.**

One way of embossing the sheet of paper is by hand using a Brailler, which resembles a typewriter with a key corresponding to each of the six points (plus a space key). To produce a dot, the user applies force to the appropriate key which embosses the paper. For mass production, plates are formed which are then pressed against the paper sheets to form pages of Braille. In this case, the size of the dot is varying in size due to the force they apply. The other means of writing are- the Braille embosser, and the Braille printers, connected as peripherals to computers, which emboss the dots on paper according to the information sent in electronic form.

The size of the page varies between documents produced by different means (hand, embosser or Braille printer). In addition, the color of the paper varies, as it does not play any role in

conveying the written information. The majority of Braille documents, however, are either buff-colored or white. In addition, their surface usually has a low gloss finish point [1].

The Braille characters are embossed in lines from the top of the document to the bottom much like printed documents. Braille documents can have the dots embossed on one side or on both. Braille paper must be much heavier to hold the dot formation and the dots themselves considerably increase the effective thickness of a page. The result is that embossed Braille is very bulky. To mitigate this problem somewhat, most larger Braille books are published in "inter-point", that is with the embossing done on both sides of each sheet, with a slight diagonal offset to prevent the dots on the two sides from interfering with each other [1] [27]. Inter-point Braille type rarely exists in Amharic Braille.

As stated the 64 distinct characters are insufficient to cover all possible print signs and their variants, it is necessary to use multi-character sequences for some purposes. Often this is accomplished by using certain characters as "prefixes" or "indicators" that affect the meaning of subsequent cells. For example, in English a dot 6 before a letter indicates that the letter is a capital; otherwise it is understood to be lower case.

A standard sheet of Braille has about 40 (forty) cells per line and may contain 20 or more lines, that is 1000 characters, can fit on a page of the usual size (11 inches wide by 11 to 12 inches deep). This contrast with the 3500 characters that will fit on a standard, smaller, typed page [27] [42].

The literary Braille codes for English and many other languages employ contractions that substitute shorter sequences for the full spelling of commonly-occurring letter groups [27]. This is partly because of the bulk problem, and partly to improve the speed of writing and reading.

When contractions are used, the Braille is usually called grade 2 in contrast to grade 1 transcription where all words are spelled out letter-by-letter. In English, which has 189 contractions, almost all Braille is grade 2 [27]. But the Ethiopic language such as Amharic does

not have grade 2 because as we have discussed in section 2.5.1, grade 1 Braille does not have legal recognition [62].

2.6.2 Braille Reading system

Braille is more complicated than sighted reading. Braille characters are formed by the presence or absence of dots arranged in a two by three matrix known as the Braille 'cell' [22]. A sheet of printed Braille will have a series of cells embossed on thick paper and passing one's forefinger over each line of embossed cells can sense the embossing [42].

Braille is a system of raised dots, designed to be read by scanning the ball of the fingertips across a line of embossed paper, or along the pins of a refreshable Braille display (an electro-mechanical device that serves as the output for a computer. Competent Braille readers may achieve reading speeds upwards of 190 words per minute, although there is of course considerable individual variation among reading speeds based on age and experience, proficiency, and the type of material being read[27] [46].

All dots on a Braille page should fall on an orthogonal grid. When texts are printed double sided (Inter-point), the grid of the inter-point text is shifted so that the dots fall in between the primary side dots. Braille readers can usually read contracted Braille faster than uncontracted Braille. This is not just a matter of familiarity and practice. Contracted Braille uses a smaller total number of cells but a greater number of different cells. It seems likely that it is the combination of fewer cells with more variation in dot patterns that makes reading faster.

According to American Council of the Blind, people who are fluent in Braille can typically read at a rate of 125 to 200 words per minute. On average, eighth graders read at a rate of 205 words per minute, and college students read at 280 words per minute as per University at Buffalo [63]. No documented information found in Amharic Braille user [51].

2.7 Computer Braille representation

Computer Braille refers to the use of a table that associates print characters one-to-one with Braille cells. The original purpose of computer Braille was as a digital format for representing Braille cells by using (the ASCII codes for) print characters. Computer Braille later became a way of representing individual print characters by Braille cells. Computer Braille was not designed with tactile readers in mind and is not a transcribing code since it doesn't encompass rules for markup, context, and formatting.

2.8 Representation of Amharic Braille character

Braille, like printed document, consists of different characters which is called Braille cell. Based on the rule set for its representation, the Braille cell may interpret to printed character in one-to-one way or one –to-many printed characters.

2.8.1 Braille characters

The dimensions of a Braille dot have strict regular structure. That is, in Braille, dot height, cell size and cell spacing are always uniform [27]**.** The horizontal and vertical distance between dots in a character, the distance between cells representing a word and the inter-line distance are also specified by the Library of Congress [2] [25].Theoretically, the dimension of Braille dots must fall within certain intervals as indicated in figure 2.5. Practically most instances of Braille are close to these standards. There are slight variations between Braille produced by different devices.

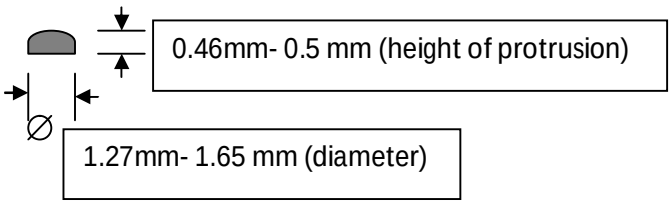


Figure 2.5 Braille Dot Dimension

Each Braille character is a unique combination of dots within the cell. Braille cells with six points is widespread; in Ethiopia also this is the Braille cell so far used [62] [65]. In this study Braille characters are assumed to be composed of six or twelve or eighteen points for Amharic Braille.

2.9 Interpretation of Amharic Braille character

The Amharic printed word is represented by a set of Braille characters forming a Braille word. The characters inside a Braille word are separated by a distance greater than the horizontal one between points (dots within a cell/character) and less than that between words [25] [51]. The distance between words in a line of Amharic Braille characters consists of one or more space characters. The space character is a Braille cell with no dots (all points are flat).

In Braille there is 63 (2^6 -1)(excluding one for space) possible combinations of dots which can be used to represent all Amharic printed symbols. A particular combination of dots may correspond to more than one different printed character (at different occasions). For example, 1, 1:2, and 1:4 are the same for Arabic and Ethiopic number one, two and three. On the other hand, a character may require more than one Braille cell. For example, all the Amharic characters except the 6th forms and Amharic numerals (like 1:2:5 and 2:6 for character h; 1:2:3 and 2:6 for character l; 1:2 and 2:6 for character b; 1:2:3:4:5:6 and 1 for number 1) [62] [66]. Furthermore, there are different representation rules for other types of documents, like those containing music or mathematical expressions [25] [62].

2.9.1 Amharic Grade **1** *Braille*

In grade 1 Braille, each printed letter corresponds to one Braille character (e.g. all the 6th forms of Amharic Braille characters like 1:4:6, 1:2:3:4:5 and 1:2 for characters >, Q, and B, respectively). Punctuation marks are also represented by one Braille characters (e.g. question mark, full stop, &, and @) [62] [67]. In this case, while reading, the interpretation is made according to the context. The combination of dots in each Braille character is designed in such a

way that symbols are distinguished from certain others by a translation of the same dot arrangement (e.g. Arabic and Ethiopic numerals) [25] [62] [67].

It is not possible to represent the large number of existing symbols, in Amharic language by a combination of dots in a single Braille cell. Therefore, all the Amharic symbols (characters) except some punctuation marks (such as →, ←, etc) are represented by two Braille characters [25] [62] [67]. Furthermore, the Arabic digits "1", "2",… "9", "0" are represented by the Braille characters corresponding to the range of English letters "a",...,"j"., but preceded by the numeral symbol(3:4:5:6) [25]. Still, grade 1 Braille is quite straightforward to interpret. The numeral symbol is an example of a mode symbol (such as 3:4:5:6. for Arabic numerals and 1:2:3:4:5:6 for Ethiopic numerals) [62] [67].

2.9.2 Grade 2 Braille/contraction

The letter-by-letter representation of a word as in grade 1 Braille leads to the production of quite voluminous Braille documents (a dictionary can occupy a bookcase). In English Braille therefore, attempts have been made to represent frequently occurring letter strings by one or two Braille characters [25] [62]. The rule for using them is referred to as grade 2 Braille (also called contracted or literary Braille) [25].

Grade 2 is the everyday form used for general purposes, such as Braille periodicals, books, and letter writing. It is used for the expression of conjunctions, prepositions, pronouns, prefixes, suffixes, frequently recurring groups of letters and common words. Its primary purpose is to reduce the bulkiness of Braille books which means a saving in the cost of production as well as in storage and cost of distribution. It also saves the braillist some of the effort involved in reading and writing.

The strings of characters that can be contracted have been selected according to the frequency that they occur and therefore, differ between languages. In English grade 2 Braille, for instance, there are over 200 contractions and short form words [25].

Regarding the Amharic Braille, as indicated by the Braille improvement committee (1995 E.C), the design of Amharic Braille contraction along with some other issues was deferred for some other time, for the reason that issues with the basic Amharic Braille code yet not finalized by the time [62] [66].

2.10 Features of Amharic Braille Code

The Amharic Braille character composed of three parts [62] [67]: Braille character for printed symbol, punctuation marks and numerals.

All basic Amharic characters are represented with a single Braille character (see Table 2.3). Each Amharic character has seven variant. Except the sixth variant, the remaining six sounds add the particular vowel code to get the required sound in a character [62]. For example, a Braille dot 2:3 is vowel for the first variant in all the character. To write first form of Fidel ሀ, for instance (1:2:3:5:6)→(1:2:3:5:6 and 2:3)

The Amharic Braille also consists of different punctuation marks. Most of the Braille code for this purpose is taken from English. For those, unique to Amharic language different dot published by the committee (see Table 2.5) are used [62] [67].

Amharic Braille uses two types of numerals: Arabic and Ethiopic. In Arabic Braille the numerals 1, 2… 0 uses the code for a, b…j. The same dot combination is used in Amharic Braille also. But the numerals translation mode is different with the Arabic Braille. For example: while Arabic Braille numerals mode is 3:4:5:6, and Ethiopic is 1:2:3:4:5:6. To write number 3 in Arabic 4:5:6 and 1:2, where as in Amharic 1:2:3:4:5:6 and 1:2 are used.

2.11 Features of embossed Amharic Braille

As describe before, the Amharic Braille, like any other Braille documents, are formed by embossing the dots on the back side of the medium sheet so that they can be read from the facing

side. There are also different sized Braille sheet; the size of the page varies between documents produced by different means and the color of the paper varies, as it does not play any role in conveying the written information. The only information recorded on a Braille page is in terms of the protrusion created by embossing the card/Braille paper (under uniform illumination a Braille document page appears black). The fact that Braille documents are not intended to convey any visual information.

The Braille characters are embossed in lines from the top of the document to the bottom much like printed documents. Braille documents can have the dots embossed single side or double side. As the volume of Braille documents is quite large, it is advantageous to use both sides of the sheet. However, this is not the practice with Amharic Braille [25] [62].

Chapter Three

Optical Braille Recognition systems

3.1 Introduction

Recognition of document images greatly affected by quality and degradation level of printed and Braille documents. Document images from printed documents, such as books, magazines, newspapers, etc. are extremely poor in quality. Popular artifacts in printed document images include: large ink-blobs joining disjoint characters or components, cuts at arbitrary direction due to paper quality or foreign materials, floating ink from facing pages etc. In addition the following artifacts are observed in Braille images: (i) Excessive dusty noise, (ii) Connected dots due to repetitive use, (iii) Vertical cuts due to folding of the paper, (iv) Degradation of dots due to the poor quality of paper and ways of writing, etc.

Further, noise is also created in handwritten Braille documents when the writer made a mistake. The mistake is corrected either by removing from the back side of the Braille or by putting full 6 dots to the line or the paragraph. However, the recognition systems consider the removed dot as it exit. This greatly affects the performance of the system.

To reduce the effect of degradation and improve the performance of the recognizer, there are many noise detection and removal techniques suggested in literature.

3.2 Phases of Braille recognition system

The process of identifying and recognizing Braille characters is to some extent of different nature than that of reading printed characters. Yet, there are many interesting parallels that can be drawn. Braille documents occupy considerably greater volume than printed ones (40 to 60 times) for recording the same information. This is because of the thickness of the material, the height of protruding dots and the low information recording density resulting from the relatively large

spatial distribution of Braille characters [57]. The process of Braille document recognition passes through many stages starting from Braille acquisition to recognition of Amharic characters.

3.2.1 Image acquisition

Acquiring images from Braille documents can be done using scanner or digital camera. The mechanism used for image acquisition has been a flat-bed scanner instead of a digital camera because it is a cheap alternative which can be used for so many other applications and it is easy and quick to use [40]. The system is able to work with images of different resolution ranges from 100 to 600 dpi.

3.2.2 Image preprocessing

Preprocessing is a technique used to prepare isolated character for recognition. It is an essential step during which errors that occurred while the images were taken are eliminated. Errors include noise, deformation, bad illumination or blurring. Image preprocessing can be used for image enhancement by reducing noise, sharpening images, or rotating a skewed page [2].

The preprocessing stage aims to make the image be suitable for recognition. The preprocessing stage which includes normalization, thresholding/binarizing, noise detection and removal, edge detection, de-skewing and so on can make the initial image more suitable for later computation.

3.2.2.1 Noise detection and removal

Noise is a common phenomenon in a scanned image especially in real life Braille documents. This noise greatly affects the recognition phase [20]. So, researchers attempts to design different types of noise detection and removal techniques in order to reduce the effect of degradation during the recognition process. The detail is discussed in section 3.3.

3.2.2.2 Image Normalization

Thus size normalization or scaling help to make all characters encountered in the scanned image to be of same width and height [36]. This preprocessing step helps to effect the same action on all isolated characters in the coming stages of the recognition process. This normalization is more crucial when the classifier is a size non-variant neural network. This type of classifier accept same size of input patterns for each character representation to classify the input characters according to the pattern observed at each selected area of the binarized image.

3.2.2.3 Image Thersholding/binarizatation

Image thresholding transforms a grayscale image into its binary version representing objects and background, respectively [2]. The quality of the binarization step is critical for subsequent analysis. If poorly binarized images are used, document understanding would be difficult or even impossible.

Thresholding or binarization of documents can be categorized into two main classes [51] [59]: global and local thresholding. Global thresholding techniques use a single threshold; on the other hand, local thresholding techniques compute a separate threshold based on the neighborhood of the pixels.

3.2.3 Segmentation

Segmentation is a process of separation of dots from Braille image that can further be grouped into a cell [51]. These cells further grouped into character, words of any strokes in Braille. The low level of abstraction to be extracted in the Braille recognition system is the single dot that could have six alternative positions in a cell as depicted in figure 3.1.

Column 1 and Row 1: Dot 1
Column 1 and Row 2: Dot 2
Column 1 and Row 3: Dot 3
Column 2 and Row 1: Dot 4
Column 2 and Row 2: Dot 5
Column 2 and Row 3: Dot 6

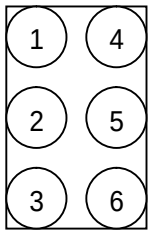


Figure 3.1 Braille Cell with Dots Position

There are different segmentation techniques particularly applied in Braille images such as mesh-grids, and local measure. Local measure works using the difference dot pixel intensity level as a threshold value. Because dots produce different color level at the top and bottom. On the other hand, mesh grid work based on constructing a vertical and horizontal grid on the strictly arranged Braille cells. It is applicable and widely used in Braille because Braille cells arranged a strictly following vertical and horizontal layout of the document [51].

3.2.4 Feature Extraction

Feature extraction is a representational mechanism of the Braille image. At this stage the identity of each of the Braille detected is recognized. Since each character is distinguished from the rest by its unique combination of dots, then, the character can be described in terms of the number of dot positions in a Braille cell. Checking the existence of a dot at each of the six positions of points in a Braille cell is straight forward because the positions of Braille characters in the image have already been identified. In order to do these, the first approach is to describe Braille characters as a region in the image; this region is divided into six equal compartments (two across, three down) in which the search for dots is performed. If the exact positions of dots in the image are known, it is easy to check if one of them is included in a compartment along specified directions following mesh grid that has been constructed. The six possible positions of the dots are known for each individual Braille character position in the image. Hence, all the possible character positions are examined. Within each one of them, the intersections of grid lines are considered [25].

3.2.5 Classification

Image classification is a fundamental problem in pattern recognition. Pattern recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest, make sound and reasonable decisions about the categories of the patterns [59].

In classification, the objective is to categorize objects in the scene from a set of measurements of the objects [7]. The measured values are the features of the pattern. A set of similar objects or patterns possessing more or less identical features are belongs to a certain category called classes.

3.2.5.1 Techniques in Pattern Recognition

Pattern recognition as a science is the study of how patterns in real world or in the brain of human being are processed and how to adopt the techniques to the machine. It encompasses a wide range of information processing problems of greater practical significance, from speech recognition and the classification of handwritten characters, to fault detection in machinery and medical diagnosis. However, to solve these problems using computers have, in many cases, proved to be immensely difficult. In order to have the best opportunity of developing effective solutions, it is important principled approach based on sound theoretical concepts [8].

The most general and most natural framework in which to formulate solutions to pattern recognition problems is a statistical one [37], which recognize the probabilistic nature of both the information process, and of the form in the result. Statistical pattern recognition is a well established field with a long history. Unlike more analytically based information processing methods, neural computation effectively explores the information contained within the input data, without further assumptions. Statistical methods are based on assumptions about input data. Neural networks, on the other hand, build relationships in the input datasets through the iterative presentation of the data and the intrinsic mapping characteristics of neural topologies, normally referred to as learning. It is well known that application of neural network is effective for

classification problems. Some studies reported that application of neural network to Braille recognition improves its accuracy [37].

3.2.5.2 Neural Networks

Neural networks were initially studied by computer and cognitive scientists in the late 1950s and early 1960s in an attempt to model sensory perception in biological organisms. Neural networks have been applied to many problems since they were first introduced, including pattern recognition, handwritten character recognition, speech recognition, financial and economic modeling, and next-generation computing models [17].

Artificial neural networks (ANN) have been developed as generalizations of mathematical models of biological nervous systems. The model of computing elements, called McCulloch – Pitts neurons, proposed by McCulloch and Pitts in 1943, which performs a weighted sum of the inputs to the elements, followed by threshold logic operation as a starting point for the development of ANN [31] [9].

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems. ANN's, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANN's as well [33].

Artificial Neural Networks (ANN) are usually formed from many hundreds or thousands of simple processing units, connected in parallel and feeding forward in several layers.

Because of the fast and inexpensive personal computers availability, the interest in ANN's has flourished in the current digital world. The basic motive of the development of the ANN was to

make computers do things, what a human being can do. The development of an ANN is an attempt to simulate a human brain [33].

The most common ANN model is the multilayer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. Multilayer neural networks are organized in layers of neurons and implement a feed-forward processing chain. Multilayer perceptrons are more flexible than single perceptrons. The most important learning rule for network training is the back-propagation algorithm. With back-propagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fedback (back-propagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output.

3.2.5.3 Design of ANN

Designing a neural network is a complex tasks and it is one of the major concerns of the system designers. Designing a neural network consist of the following set of decisions and activities [59].

- Deciding the number of layers in the network.
- Arranging neurons in various layers.
- Deciding the type of connections among neurons for different layers, as well as among the neurons within a layer.
- Deciding the way a neuron receives input and produces output.
- Determining the strength of connection within the network by allowing the network to come up with appropriate values of connection weights by using a training data set.

Designing artificial neural network is not a one shot process rather it is an iterative processes to achieve the optimal results by adjusting the parameters of the neural network depending on the proximity of the output with the desired target.

3.2.5.4 Architecture of ANN

Architecture of a neural network is the specific arrangement and connection of the neurons that are organized in the form of layers, make up the network. A layer is a collection of neurons that can be thought of as performing some type of common functions. All neurons nets have an input layer and have outputs to interface with the external environment. Each input layer and each output layer has at least one neuron. Any neuron that is not in an input layer or in an output layer is said to be in a hidden layer (shown in Fig. 3.2), because neither its input nor its output can be observed from outside [33]. The most common neural network architectures have three layers. The first layer is called the input layer and is one of the layers exposed to external signals. The input layer transmits signals to the neurons in the next layer, which is called a hidden layer. The hidden layer extracts relevant features or patterns from the received signals. Those features or patterns that are considered important are then directed to the output layer, the final layer of the network. Normally, there are two basic topologies of neural networks feed-forward networks and recurrent networks [60]. In a feed forward network, links are unidirectional and there are no cycles. Technically speaking, a feed forward network is a directed cyclic graph. In a layered feed forward network, each unit is linked only to units in the next layer; there are no links between units in the same layer, no links backward to a previous layer, and no links that skip a layer. The significance of the lack of cycles is that computation can proceed uniformly from input units to output units. The activation from the previous time step plays no part in the computation, because it is not fed back to earlier units. Hence, the feed forward network simply computes a function of the input values that depends on the weight setting-it has no internal state other than the weight themselves. A network with the input and output layer only is called single-layered network. Whereas, the multilayer feed forward networks or the multilayer preceptron, are generalizations of the single layer perceptron. It is the most popular neural network model. It includes one or more hidden layers each with their respective number of neurons as shown in

figure 3.2. The function of the hidden neurons is to intervene between the external input and the network output in some useful manner. On the other hand, in recurrent networks, the data flows to all adjacent connected units and circulates back and forth until the activation of the units is stabilized [33]. Recurrent can becomes unstable, or oscillate, or exhibit chaotic behavior. Recurrent networks can implement more complex agent designs and can model system with state.

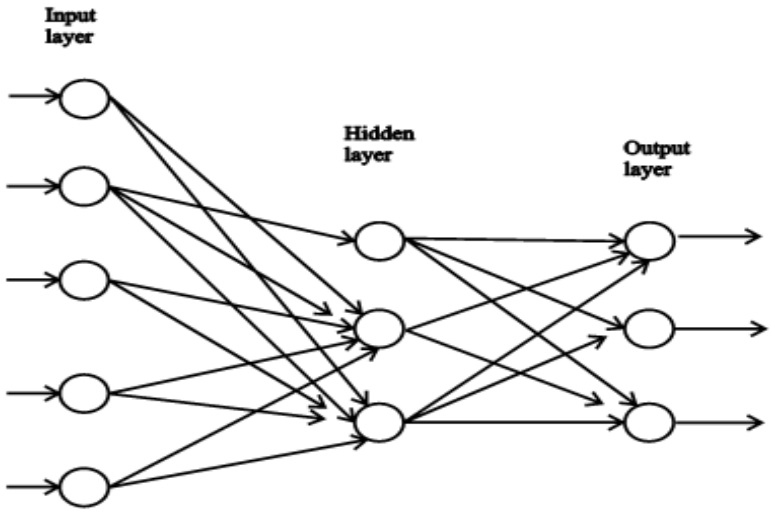


Figure 3.2 Architecture of feed-forward neural network

3.2.5.5 Learning ANN

A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to train the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. The learning situations in neural networks may be classified into two distinct sorts. These are supervised learning, and unsupervised learning. In supervised learning, an input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done, and the errors or discrepancies between the desired and actual

response for each node in the output layer are found. These are then used to determine weight changes in the net according to the prevailing learning rule. The term supervised originates from the fact that the desired signals on individual output nodes are provided by an external teacher [3].

Once the network has been initialized and the training input space prepared the network is ready to be trained. Some issues that need to be addressed upon training the network are [48]:

- How complex are the patterns for which we train the network? Complex patterns are usually characterized by feature overlap and high data size.
- What should be used for the values of:
    - Learning rate
    - Sigmoid slope; most common activation functions are the logarithmic and hyperbolic tangent sigmoid functions.
    - Weight bias
- How many Iterations (Epochs) are needed to train the network for a given number of input sets?
- What error threshold value must be used to compare against in order to prematurely stop iterations if the need arises?

3.2.5.6 Learning rate

The learning rates are the rates at which artificial neural networks learn how to deal with new patterns. The learning rate governs the rate at which the weights are allowed to change at any given presentation. This depends upon several controllable factors that have their own effect on the performance of the system. Learning rate effectively controls the size of the step that is taken in multidimensional weight space when each weight is modified. If the selected learning rate is too large, then the local minimum may be overstepped constantly, resulting in oscillations and slow convergence to the lower error state. If the learning rate is too low, the number of iterations required may be too large, resulting in slow performance. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained

system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly. Slower learning rates produce more reliable results at the expense of increased training time [33].

Usually the learning rate is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to exceed in correcting the weights, and the network will move back and forth. Small values of the learning rate will not correct the current error as quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence.

3.3 Noise Detection and removal

Digital image processing such as filtering was first developed in the 1960's. As computers became cheaper and faster, real-time image processing became available and its applications boomed. Digital filtering attempts to clear out noise, or useless and distracting information, in pictures. Examples of noise include missing pixels, Gaussian, salt and pepper, speckles and wrong pixels.

Image filtering is one of the most important operations used in image processing [30]. The importance of image filtering is constantly growing because of ever increasing use of television and video systems in consumer, commercial, medical, and communication applications. Image filtering is not only used to improve image quality but is also used as a preprocessing before most image processing operations such as encoding, recognition, compression, tracking, edge detection and noise reduction. In other words, without this preprocessing, the other processing would have inappropriate or even false results [30].

The application of noise detection and removal techniques not only increases the speed of processing, but also gives more accurate results for specific tasks. Braille images are mostly corrupted by noise due to repetitive use, bend, lightning during scanning and others. The impulse noise is the most frequently referred type of noise. These noises come in three flavors - patterned

noise, associated noise, and random noise. Patterned noise comes from graphical patterns, especially half part shaded areas, which appear on many scanned forms. Associated noise occurs when a scanned document is incorrectly thresholded and artifacts surrounding valid blobs appear in the image. Random noise comes from a bad global threshold or a dirty source document or scanner. In the case of random valued shot noise, the noise pixels have an arbitrary value [56] [58].

3.3.1 Gaussian Filtering

The essential idea in image smoothing is to filter noise present in the image signal without sacrificing the useful details. In contrast, image enhancement focuses on preferentially highlighting certain image features. Gaussian filtering is a well known technique for signal smoothing [16]. It is one of the most popular linear filters used to filter noise and smoothes the images. The Gaussian smoothing operator is a 2-D convolution operator used to remove detail and noise from images. For a Gaussian function with standard deviation σ, a 1-D Gaussian filter can be written as:

$$G_x = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$$

A 2-D Gaussian filter can be expressed as a product of two 1-D Gaussians:

$$G_{x,}G_y = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The effect of Gaussian smoothing is to blur an image, in a similar fashion to mean filter [56] [58]. The degree of smoothing is determined by the standard deviation of the Gaussian. Larger standard deviation Gaussians, of course, requires larger convolution kernels in order to be accurately represented [53].

The Gaussian smoothing method has a number of advantages [50]:

- It applies to piece-wise linear surfaces of arbitrary topological type, not only those that can be parameterized by functions defined on a rectangular domain.
- Since first order neighbors are defined implicitly in the list of edges or faces of the curve or surface, no storage is required to encode the neighborhood structures.
- The number of operations is a linear function of the total number of vertices, edges, and faces. However, since the method is very local, to obtain a long range smoothing effect, the Gaussian smoothing methods has to be applied iteratively a large number of times, producing significant shrinkage as a by-product.

3.3.2 Median Filter

The median filter is one of non-linear, low-pass filtering method, used to remove salt and pepper (speckle/spikes) noise from an image. A median filter can outperform linear, low-pass filters on this type of noisy image because it can potentially remove all the noises without affecting the "clean" pixels [25]. Median filters remove isolated pixels, whether they are bright or dark. The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.

Median filter is excellent and applicable to remove a noise from a scanned image than Gaussian and Morphing filters because the Gaussian filter blurs the image where as the morphing filter removes the pixels of the image and the image suffers a merge with close ones and also thin lines are removed[19]. Median filter also better than the non-linear mean filter because the mean filter cannot remove positive and negative impulse noises simultaneously but median can do well [25].

Median filter has played an important role in image preprocessing application with its fine detail preservation and impulse noise removal. However, the filter is constrained by a fixed input window size; this affects the quality of the filtered image. For example;

In an $N \times N$ window,

If $P > N(N+N)/2$, where P is the number of polluted pixels, then the noise will not be removed. In such case, we have to increase the size of the window to improve the filtering efficiency. On the other hand,

if $P < N(N+N)/2$, then the feature pixel will be replaced by other irrelevant pixels, and the features will be impaired. In such case, we have to reduce the size of the winnow to preserve image features. In order to strike a balance between noise removal and feature preservation, the size of the window has to be carefully chosen, and usually with contradictive effect [26].

3.3.3 Adaptive median Filtering

As we have discussed above the median filtering performs quite well, but it does not work when the probability of the impulse noise occurrence become high. To overcome this, Hwang and Hadded [25] propose a new algorithm called adaptive median filtering. It is an iterative order statistic filter. The iterative process was introduced to detect and replace corrupted pixels only. This filter is robust in removing mixed impulses with high probability of occurrence while preserving sharpness. It works by using variable window size. The window length is determined by the width of the impulsive noise presented in the input sample.

3.3.4 Morphological operations

Point and neighborhood operations are generally designed to alter the look or appearance of an image for visual considerations. Morphological operations are used to understand the structure or form of an image. This usually means identifying objects or boundaries within an image.

Morphological operations play a key role in applications such as machine vision and automatic object detection [24].

Morphology is a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull. It has different techniques for pre- or post-processing, such as morphological filtering, thinning, and pruning. The operations dilation and erosion are fundamental to morphological image processing. Others are special cases of these primary operations or are cascaded applications of them. Morphological operations are usually performed on binary images where the pixel values are either 0 (white) or 1 (black). While most morphological operations focus on binary images, some also can be applied to grayscale images [21].

3.3.4.1 Binary Erosion and Dilation

Erosion and dilation are related to convolution but are more for logical decision-making than numeric calculation. Like convolution, binary morphological operators such as erosion and dilation combine a local neighborhood of pixels with a pixel mask to achieve the result. The output pixel, 0, is set to either a hit (1) or a miss (0) based on the logical AND relationship.

Erosion shrinks or thins objects in a binary image [21]. As in dilation, the manner and extent of shrinking is controlled by a structuring element. Binary erosion uses the following for its mask:

1 1 1
1 1 1
1 1 1

This means that every pixel in the neighborhood must be 1 for the output pixel to be 1. Otherwise, the pixel will become 0. No matter what value the neighboring pixels have, if the central pixel is 0 the output pixel is 0. Just a single 0 pixel anywhere within the neighborhood will cause the output pixel to become 0. Erosion can be used to eliminate unwanted white noise pixels from an otherwise black area. The only condition in which a white pixel will remain white

in the output image is if all of its neighbors are white. The effect on a binary image is to diminish, or erode, the edges of a white area of pixels.

Dilation is an operation that grows or thickens objects in a binary image [21]. The specific manner and extent of this thickening is controlled by a shape referred to as a structuring element. Computationally, structuring elements typically are represented by a matrix of 0s and 1s. Sometimes it is convenient to show only the 1s. Dilation is the opposite of erosion. Its mask is:

0 0 0
0 0 0
0 0 0

This mask will make white areas grow, or dilate. The same rules that applied to erosion conditions apply to dilation, but the logic is inverted - use the NAND rather than the AND logical operation. Being the opposite of erosion, dilation will allow a black pixel to remain black only if all of its neighbors are black. This operator is useful for removing isolated black pixels from an image.

3.3.4.2 Lookup table

The lookup table is used to pre-compute the output pixels value for every possible neighborhood configuration and then store the answer in a table for later use. For instance, there are 2^9=512 different 3x3 configurations of pixel values in a binary image. To make the use of lookup table practical, we must assign a unique index to each possible configuration. A simple way to do this, for say, the 3x3 case, is to multiply each 3x3 configuration element-wise by the matrix.

3.4 Review of related research

This section covers a concise review on some of the Braille recognition systems being developed for various languages (English, Chinese, etc) and also tries to relate their applicability for Ethiopic script.

3.4.1 Optical Braille Recognition for Latin Based Characters

A number of researches are done towards the recognition of Latin Braille documents. We present two works that are relevant to our present study.

3.4.1.1 Analysis of Scanned Braille Documents

The objective of Ritchings [45] work is set based on Latin characters and Arabic numbers. This work tries to recognize double sided Braille documents. The documents are scanned using flatbed scanner to acquire a gray scale image at 100 dpi and 16 gray level images are produced. Under the scanning configuration used, the grey level values of the background pixels are from 10 to 12. Lighter areas have grey levels with values ranging from 13 to 15, and the darker areas corresponding to shadows have grey level values less than 10.

From the scanned images, Braille dots are then identified. The protrusions and the depressions on the surface of the Braille document give rise to differences in the grey levels in the image. In the scanning direction (vertical) the protrusions create a dark area first and then a lighter one, whereas, the depressions produce the opposite as shown in figure 3.3.

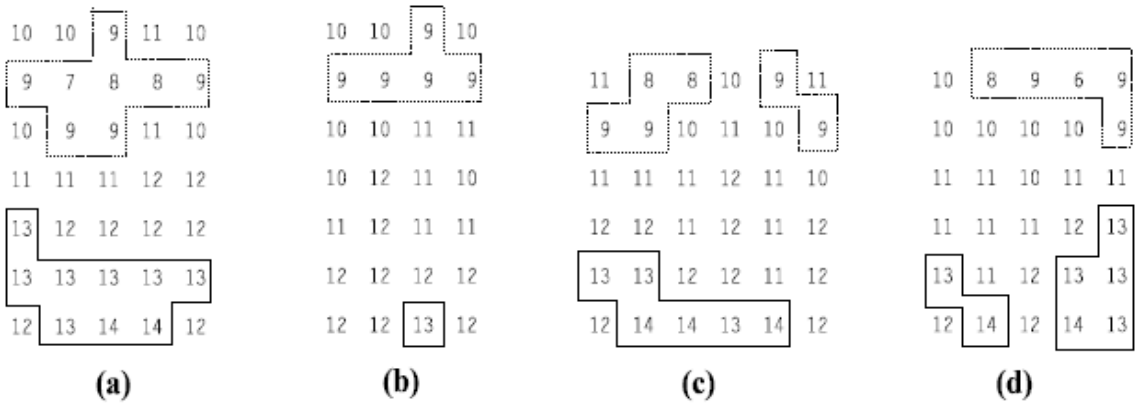


Figure 3.3 Image areas (grey levels) corresponding to protrusions [45]

At this stage the researcher attempts to apply segmentation to identify characters. The first action locates the lines of Braille characters then, knowing the average line height, the rest of the text lines together with the blank lines are also detected. Having located the lines of Braille, the characters in each line are next segmented. Character segmentation is based on the analysis of the horizontal spacing between columns with dots.

The segmented characters are then recognized based on the features extracted. To achieve this, the rectangular area in the image that corresponds to a segmented character is divided into six equal compartments. These are arranged in three rows and two columns and represent the bounds of the expected positions of dots in a Braille character. If there is a dot in any of the compartments then a bit is set in a binary number at the position that corresponds to the number of the compartment. Based on this binary number the Braille character identified from the character set used.

This paper concentrates on the application of OCR techniques to inter-point Braille documents. Preliminary results from images of inter-point buff-colored Braille documents seem promising. In the experimental stage, an average of just over 98.5% of the protrusions and 97.6% of the depressions are correctly recognized. The majority of the errors can be attributed to the quality of the image of the Braille document. Very old documents with some of the protrusions flattened due to heavy use will give rise to more incorrectly recognized characters.

3.4.1.2 Statistical template matching for Translation of Braille

This work [13] tries to recognize both single and double sided documents written in Latin characters and Arabic numbers.

The image is captured using a digital camera placed above the Braille page. The digitized images are preprocessed for the purpose of noise filtering and edge enhancement. Noise filtering is used to reduce the noise generated during the digitization process by applying a Gaussian filter whereas, the edge enhancement is used to sharpen the fine details of the image that had been blurred by applying convolution Sobel kernels X and Y. The resulting values are obtained by:

$$output = |convolute(input, X)| + |convolute(input, Y)|$$

The filtered Braille images are binarized using a gray level histogram of the page to select a threshold value and segmented to identify characters in the Braille image. The feature extraction stage detects the boundary to highlight the dots by using the chain code algorithm; back face removal-to identify the front side and back side dots and then remove the back side dots; centriod determination to locate the central point of the dots and; dot alignment to align the extracted dots with the coordinate system.

To translate the extracted features first the dots are grouped into cells based on the distance between the centriod and the four possible neighbors. Within each cell the dot pattern is determined and represented by bit string. Then search the bit string against the Braille dictionary. Finally group each character into word and check each word against English dictionary.

Using both single sided and double sided pages the system recognizes 100% and 97% accuracy, respectively. The system takes advantages of the regular spacing between Braille dots with in a cell and the regular spacing between cells.

3.4.2 Optical Braille Recognition for English and Chinese Characters

There are attempts to develop a recognition system for English/Chinese Braille documents. In this subsection, we present a notable work [12] that design regular feature extraction for recognition of Braille. This work tried to recognize both single side and double side documents with digital camera to acquire a gray scale image of a Braille documents.

Scene construction is used to maximize the use of prior knowledge of the scene, i.e. by exploiting existing knowledge; and to trivialize the problem of image analysis as far as possible, i.e. by effective imposition of constraints. Based on these, the researcher decided that the Braille pages should be illuminated with a yellow polarized light source placed at an angle about 45 degrees away from the page top.

The digitization process is performed by using digital camera, which is placed directly above the Braille page. The captured digitized image has a 512 * 512 pixel resolution with each pixel representing an 8-bit gray scale value. From the binarized images the researchers are apply a Gaussian noise filtering to filter the noise imposed on the gray scale images and used Sobel kernels to sharpen the fine details of the image that are blurred.

From the preprocessed the researchers considered a histogram made up of only those pixels that lie at or near the edges of the Braille dots to separate the background and foreground. To determine which pixels lie on or near an edge, the Laplacian of Gaussian operator $\nabla^2 G$ is applied to the entire image to identify the edge pixels. The average gray-level value of these edge pixels used as the global threshold value. Finally the researchers used a threshold value 100 for single-sided Braille pages and 85 for double-sided pages. Based on the boundary coordinates information and the illumination characteristics of the dots the front-faced and the back-faced dots are extracted.

The Braille dots are grouped into cells based on the distance between the centriod and the four possible neighbors of each dot. Within each cells, the dot pattern is also determined and represented by a bit string. The bit strings of the cells are searched against the Braille dictionary, and the retrieved characters are grouped into words. Each word is then check against an English dictionary. The system performs an accuracy of 100% and 97% for single sided and double sided Braille documents respectively.

3.4.3 Optical Braille Recognition for Arabic Characters

AbdulMalik et.al. [2], attempts to design Arabic optical Braille recognition system. The objective of the work is to build fully functional Arabic Braille Recognition system. This work tries to recognize both single side and double side documents with flat bed scanner. The researcher acquired colored Braille image and converted into gray level because dealing with gray level image is much easier and faster.

Cropping the image frame is used to minimize the effect of black or white frame in the image on thresholding. The researchers calculate the average gray level for the whole image and then for row and column separately. Finally finding a row or column average gray level that is above or below 15% of the whole image average gray level is an indication to delete it. Based on the thresholding value of the cropped image the Braille pages are classified into three i.e., dark, bright, and background. The tilted scanned images are corrected and de-skewed by applying a binary search algorithm. The maximum degree of recognizing a de-skewed image is 4 degrees from either the left or the right side.

The dots are identified based on the average spacing between the cells and among the cells. Local and global measure is used to correct the errors that occur during identification. However, global measure gave excellent results [2].

The researchers follow the following steps to detect the whole dots:

1. Check a detected dot has at least three columns, less than that is not counted in.
2. An empty array that has the same size of the one holding dots parts is constructed.
3. Perform a vertical search on the array starting from up to down, when three parts of a dot that are not more than 6 pixels apart are found then this is considered a point. This point is registered at the corresponding position in the array as 4x4 point.
4. The region holding the detected dot part in the previous step is deleted so that it will not be considered in future searches (Detect and Clean.)

Having identified all possible valid dots, the system defines the region containing all the dots such that no dots exist outside this region. To do this, the researcher has to add each row and column in the array separately then take the first and last positive value positions. After that one of the two following algorithms can be used:

Algorithm 1:

1. Determine the number of rows and columns in the defined region. The average line and column width have been identified.

2. The resulting value from the first step will take to reach the beginning of any cell by using the line and column number.

This algorithm does not depend on fixed lengths for cells heights and distances between them.

Algorithm 2:

The steps involved here are as follows:

1. Determine the horizontal projection for the array holding the dots and then determine the average distances between the rows holding the dots.
2. Determine the vertical projection for the array holding the dots and then determine the average distances between the columns holding the dots.
3. Using the horizontal and vertical projections reach any cell. This is because it consider that any consecutive 3 rows and 2 columns as a cell as long as the distance between the rows and columns does not exceed the averages.

Comparing the two algorithms, even though the latter one does not require fixed lengths to determine the cells, it is less accurate than the former one, especially which most Braille documents use the lengths and distances defined.

Having used either of the algorithms, convert the cell to binary code. To do that, first it divides the cell into six regions. Then take each region separately, if the sum of its elements is more than 8 pixels then it will be assigned the number 1, otherwise it will be assigned 0. Then convert the image to its decimal code representation:

$$decimal - code = b_1 + b_2 \times 2 + b_3 \times 4 + b_4 \times 8 + b_5 \times 16 + b_6 \times 32$$

The final step is to convert the decimal code to its corresponding Arabic letters to get the translation. The proposed system is tested using variation of Braille documents: skewed, reversed, or worn-out of both single and double sided documents and they found an overall accuracy of 99%. The researchers recommend that the system should develop for Arabic/English languages.

3.4.4 Image processing Techniques for Braille writing recognition

This system is designed for any language. In this system, the researchers [40] presented the development of BrailLector, a system able to speak from Braille writing. By means of dynamic thresholding, adaptive Braille grid, recovery dots techniques and TTS software (Text-To-Speech).

The researchers create a database contains both single and double-sided documents, which have dots in one or both sides of the sheet respectively for testing the developed system and analyzing of the error variance. The Braille image is acquired using a flat-bed scanner with any resolution since it uses interpolation methods to resize the input image [40]. During image preprocessing, the following functions are performed: dynamic thresholding, pattern detection, Braille grid creation and dots recovery using Braille grid.

The researchers apply iterative thresholding algorithms for identifying the black, white and gray (background) of the gray scale image. Based on this the protrusion and depression areas are identified. The protrusions on a Braille sheet have a brilliant zone above and a dark zone below. The depressions have exactly the opposite pattern so, this difference facilitate the separation between front and back side dots;

- Moving white “islands” 4 pixels downwards and doing a logical “and” with black spots extracts front side dots.
- Moving white “islands” 4 pixels upwards and doing a logical “and” with black spots extracts back side dots.

This algorithm consists of a “shift and overlap” process since it only moves the spots downwards or upwards and carries out a logical and.

Skew angle of the scanned document is detected by means of horizontal histogram and mass centers calculation and corrected rotating the original image. To identify dots that are missed in the identification of protrusion and depression the researchers apply an adaptive mesh grid

algorithm. This algorithm first builds the column and then rows based on the distance between dots. This method is suitable for coping Braille sheets without losing the format [40].

After mesh building, all valid Braille positions are known. Those intersections between rows and columns define a valid position for a Braille dot. In addition, the system checks the potential positions of dots and recover original dots that are belongs to correct Braille positions and discriminate false dots that are out of the valid places for a Braille dots. Based on the mesh the final image is analyzed and text is segmented in rows and characters. Every character converted into a binary number according to the active dots in the image. This way of Braille text binarization makes the global system independent of the language of the document and easily configurable for adding different alphabets [40]. This final output can be presented in different formats such as a text file, a new Braille printed copy, voice or even mp3 audio format. The global image processing algorithms (dynamic thresholding, pattern detection, Braille grid creation and dots recovery using Braille grid) is very fast and robust and perform an accuracy of 99.9% for double sided Braille documents.

3.4.5 Recognition of Amharic Braille

Teshome [51] attempt to develop Amharic Braille recognition system that enables to recognize optically scanned single-sided Amharic Braille and convert to its equivalent printed Amharic text characters. The researchers collected both manually and typewritten single sided Braille documents. These Braille documents are both plastic and paper with white and light yellow color, and its size is 11×11.5 inches. The Braille documents are digitized using a flat-bed scanner with 200 dots per inch (dpi) for both color and gray scale Braille images.

The researchers applied a global thresholding technique for image thresholding/binarization. A threshold value of less than or equal to 202 or greater than or equal to 247 color values used during binarization. Using two threshold values for a single dot is important for two sided Braille documents [51]. The result of binarization has the feature that the foreground (content of the

image) is represented by black color and background with white color. This output of the binarization module is fed to the next level of processing, which is image segmentation.

Segmenting the Braille image is performed in two steps: first, gridlines mesh is constructed; and then dots are searched following the grids mesh with threshold (see figure 3.4). Second, clearly display the dot with single white pixel on black background as shown in figure 3.5 below.

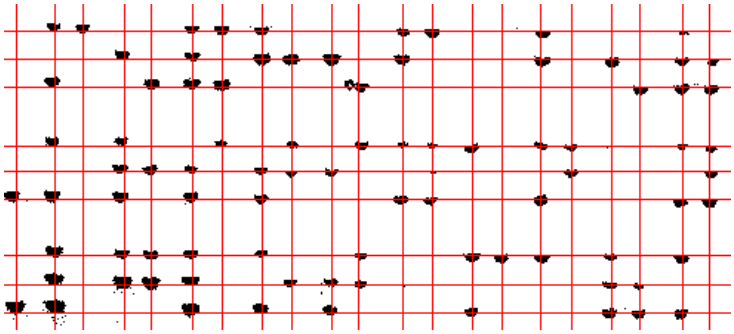


Figure.3.4 Braille image with Global grids (the result of the first phase)

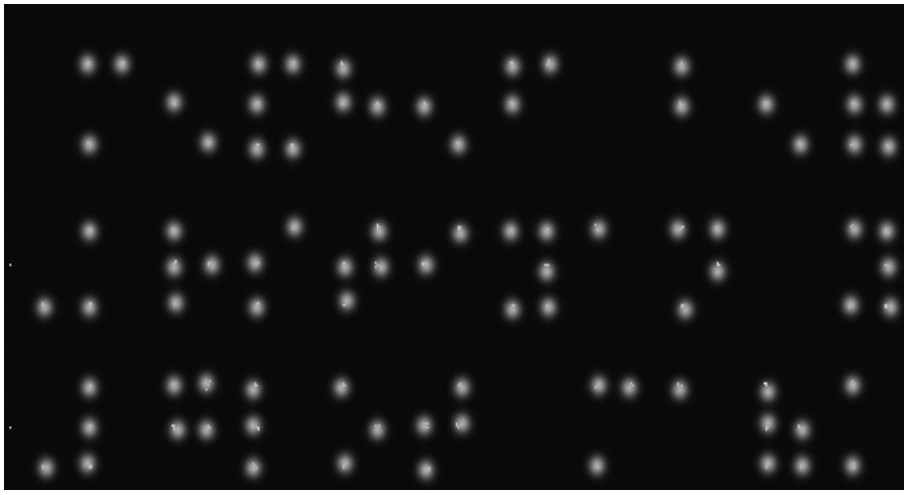


Figure 3.5 Segmented Braille image (the result of the second phase)

The result of this is used as an input for feature extraction to group Braille cells to draw patterns for Amharic Braille code. For feature extraction Teshome [51] adopted modification of region based approach. The modification is that the six compartment of the cell checked for the presence of a single pixel rather than all pixels in the area. Then, context analysis is performed to

recognize the cell as part of the Braille character or stand alone as a Braille character. Feed forward neural network is applied on the extracted features to recognize the characters.

In this research the proposed system is tested on single sided and clean Braille documents and found an accuracy of 92.5%. The performance of the network is decreased with the increase in test set. Global thresholding techniques is simple to use and computationally less expensive than local thresholding.

Currently, Braille recognition research is at a high level; most of the researches are not used image noise detection and removal techniques. However, in the practical image acquisition systems and conditions, noise is a common phenomenon in the acquisition process and also Braille noise is common due to repeated use, bend and scratch. As a result, these artifacts significantly affect the subsequent recognition process. So to address the problem noise detection and removal techniques need to be applied on the resulting image. Recognition of Braille images that are corrupted by noises has been the goal of the present research.

Chapter Four

Braille Recognition Techniques

4.1 Introduction

Text is extracted from a document image through optical character recognition. The accuracy of a recognition algorithm highly depends on the quality of the document image. In reality Braille documents are degraded due to different factors like repetitive use, bend due to storing inappropriate areas, lighting during scanning, etc. Segmenting a Braille document from the degraded documents for recognition is the most difficult work of an OCR system [14].

As indicated in sections 3.5.5 the algorithm suggested by Teshome [51], is not capable of recognizing Braille characters with noisy Braille documents. There are methods reported in the literature which enables OBR systems recognizes degraded Braille documents [12][13].

The main intension in conducting this study is to explore ways of improving the already adopted Amharic Braille recognition algorithm. In this chapter we have discussed the Braille recognition techniques that are used for preprocessing, segmentation, feature extraction and also the neural network techniques to classify the given Braille to print characters. Mainly the preprocessing algorithms available which could help the already adopted algorithms such as image noise detection and removal, and binarization are the main focus of this study.

4.2 Preprocessing of document images

For degraded and poor quality documents, a preprocessing stage of the grayscale source image is essential for the elimination of noisy areas and smoothing of background area. In image processing, it is usually necessary to perform high degree of noise reduction in an image before performing higher-level processing steps such as segmentation and feature extraction. Pre-processing operations are usually specialized image processing operations that transform the image into a better one with reduced noise and variation [47]. These operations include filtering

and smoothing, thinning, alignment, binarization, normalization, interpolation and baseline detection. Ideally, pre-processing should remove all variations and details from a Braille image that are meaningless to the recognition method [47].

The pre-processing operations that are considered for this study consist of four sub-operations: histogram equalization/interpolation, noise filtering, image binarization/ thresholding and morphological operations. The sequence of these operations affects the quality of removing noise in the image. Therefore, the study is used histogram equalization and filtering before binarization this is because the noise should first decrease before applying binarization to determine the threshold value. Then morphological operation follow the binarization, this is because the noises left after binarization should be removed to specifically to improve the accuracy of the segmentation phase. In general this phases increase the accuracy of the recognition.

4.2.1 Histogram Equalization

Degraded images have different pixel intensity values. The highly difference of its variation greatly affects the filtering process of the Braille image because the filtering operations are implemented equally or with different value based on the pixel intensity for each regions or the whole regions of the image. Due to this reason, the image requires adjusting pixel intensity before applying any filtering techniques. To achieve this histogram equalization is used. The histogram equalization minimizes/decreases the noise level of the Braille images.

The histogram of the processed Braille image will not be uniform, due to the discrete nature of the variables. The values in a normalized histogram are approximation to the probability of occurrence of each intensity level in the image [23]. It is represented by:-

$P_r(r_j)$, j=1,2,…,L (where, L is total number of possible intensity level)

For discrete quantities the equalization transformation becomes:

$s_k=T(r_k)$

$$= \sum_{j=1}^{k} P_r(r_j)$$

$$= \sum_{j=1}^{k} \frac{n_j}{n}$$

For k=1,2,…,L, where sk is the intensity value in the output (processed) image corresponding to value rk in the input image. Histogram equalization is implemented in the MATLAB toolbox by function histeq, which has the syntax

g=histeq(f, nlev)

Where *f* is the input image and *nlev* is the number of intensity levels specified for the output image. If *nlev* is equal to L (the total number of possible levels in the input image), then *histeq* implements the transformation function, $T(r_k)$, directly. If *nlevl* is less than L, then *histeq* attempts to distribute the levels so that they will approximate a flat histogram. However, the adaptive histogram equalization works by using bilinear interpolation instead of *nlevel*. Bilinear interpolation is often used in many image processing applications because it provides a compromise between computational efficiency and image quality [29] [32].

Interpolation works by using known data to estimate values at unknown points. Image interpolation works in two directions, and tries to achieve a best approximation of a pixel's color and intensity based on the values at surrounding pixels [32]. Bilinear interpolation is a non adaptive interpolation algorithm. It considers the closest 2×2 neighborhood of known pixel values surrounding the unknown pixel. It then takes a weighted average of these 4 pixels to arrive at its final interpolated value as shown in figure 4.1 [29].

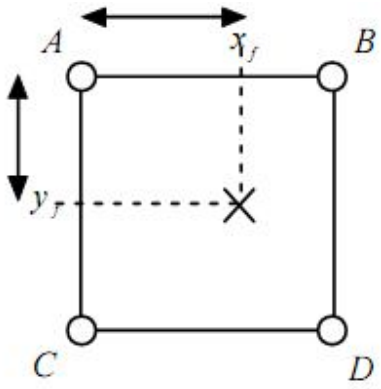


Figure 4.1 Bilinear interpolation neighborhoods

The equation used for bilinear interpolation is presented as follows

$$I'\left(\begin{bmatrix} x' \\ y' \end{bmatrix}\right) = I\left(f^{-1}\left(\begin{bmatrix} x' \\ y' \end{bmatrix}\right)\right) = I\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)$$
$$= (1-x_f)(1-y_f)A + x_f(1-y_f)B + y_f(1-x_f)C + x_f y_f D$$

Where,

$$\text{A}= I\left(\begin{bmatrix} x_i \\ y_i \end{bmatrix}\right) \quad \text{B}= I\left(\begin{bmatrix} x_i+1 \\ y_i \end{bmatrix}\right) \quad \text{C}= I\left(\begin{bmatrix} x_i \\ y_i+1 \end{bmatrix}\right) \quad \text{D}= I\left(\begin{bmatrix} x_i+1 \\ y_i+1 \end{bmatrix}\right)$$

And $x=x_i+x_f$ and $y=y_i+y_f$ with x_i and y_i being the integer components of x and y and x_f and y_f being the fractional components ($0 \leq x_f, y_f < 1$).

4.2.2 Image Filtering

In past years, linear filters became the most popular filters in image signal processing because they have the existence of robust mathematical models which can be used for their analysis and design. However, the nonlinear filters provide significantly better results in many areas [56] [39].

Traditionally, the impulse noise on the image is removed by a standard median filter which is the most popular nonlinear filter. Median filters are used mainly to remove salt-and pepper noise. Doing this, they preserve edges in the image (preserve their location and do not affect their steepness, unlike Gaussian filters), but unfortunately median filtering may destroy small features in the image and due to fixed window size in the filtering process, it will not filter correctly. For example, when the values in the interval of the window size have the same result, the median filter does not perform anything but all of the pixels may be a noise. In order to handle such type of problems the adaptive median filter is better, which varies the window size depending on pixels under considerations. For this study we compare the filtering performance of Gaussian and an adaptive median filtering. Because most researchers [12] [13] propose that Gaussian filtering performs well for Braille noise removal.

4.2.2.1 Gaussian Filtering

Gaussian smoothing is performed by convolving the vector function that parameterizes the curve with a Gaussian kernel. In the Gaussian smoothing method the new position of each vertex is computed as a weighted average of the current positions of the vertex itself, and its first order neighbors, those vertices that share an edge (or face) with the current vertex. This process is repeated a number of times [50].

In the Gaussian smoothing algorithm the position of each vertex is replaced by a convex combination of the positions of itself and its neighbors. Alternatively, Gaussian smoothing can also be reformulated as follows [50]. First, for each vertex v_i, a vector average is computed as a weighted average of the vectors v_j - v_i, that extend from the current vertex vi to a neighbor vertex v_j.

$$\Delta v_i = \sum_{j \in i^*} \omega_{ij} (v_j - v_i)$$

For each vertex v_i the weights w_{ij} are positive and add up to one, but otherwise they can be chosen in many different ways taking into consideration the neighborhood structures. One particularly simple choice that produces good results is to set w_{ij} equal to the inverse of the number of neighbors $1/|i^*|$ of vertex v_i, for each element j of i*. Once all the vector averages are computed, the vertices are updated by adding to each vertex current position v_i its corresponding displacement vector computed as the product of the vector average Δv_i and the scale factor λ, obtaining the new position v_i.

$$v_i' = v_i + \lambda v_i$$

The scale factor, which can be a common value for all the vertices or be vertex dependent, is a positive number $0<\lambda<1$.

4.2.2.2 Adaptive Median Filtering

The adaptive median filter (AMF) can be considered as an iterative order statistic filter. The iterative process was introduced in order to detect and replace corrupted pixels only. In each iteration, filtering window of different sizes is utilized. In order to simplify the description we will deal only with one filter window located at position (u,v). Let a two dimensional matrix $[X_{i,j}]$ describe the input image and W is the size of the filtered window. Let the sequence $[W_k,I]$ be the output of a local neighborhood function which contains just N=WxW samples of filtered window located at position (u,v) (assume that W is odd). Let $X_{u,v}$ denote the value of pixel $X_{u,v}$ which corresponds to the value of the a pixel at position (u,v) of the input image. Let $Y_{u,v}$ be of the output of AMF located at position (u,v). The algorithms of AMF are as follows [56]:

Step 1: Initialization

Start with the smallest window size W=3. Let the maximum window size be W_{max}(again, an odd number).

Step 2: Computation of order statistic

Let Xmin=S(0)([$W_{k,i}$]} be the output of the 0^{th} order statistic filter. Xmax=S(N)([$W_{k,i}$])is the output of the N^{th} order statistic filter and X_{med}=S((N+1)/2)($W_{k,i}$) is the output of the median filter.

Step 3: Evaluation of terminating condition

If the condition Xmin < Xmed < Xmax is satisfied then processing ends with the computation of the output value which is defined as follows [56]:

If Xmin <Xu,v < Xmax then the pixel is not corrupted by noise and the output value is the value of the original pixel. i.e, Yu,v= Xmed, if the condition is not satisfied then the computation continues.

Step 4: Increasing the window size

If the condition (Xmin < Xmed < Xmax) is not satisfy, it can be interpreted as follows. If many pixels have the same value then it is impossible to determine (with the current window size) whether the pixels are corrupted with high intensity noise or whether it is the constant area with the same color. This is the reason why the window to be increased.

If the window W is smaller than Wmax, increase the size of the window, i.e. W=W+2, and repeat the computation from step 2. If the size of the window W reaches the maximum value Wmax, the processing ends and the output value is defined as Yu,v=Xmed.

4.2.3 Image Binarization/Thresholding

Document image binarization is the initial step of most document image analysis and understanding systems that converts a grey scale image into a binary image aiming to distinguish text areas from background areas. Binarization plays a key role in document processing. Its performance affects quite critically the degree of success in subsequent character segmentation and recognition. In general, approaches that deal with document image binarization are categorized in two main classes: (i) global and (ii) local.

In a global approach, threshold selection results in a single threshold value for the entire image. Global thresholding [45] has a good performance in the case that there is a good separation between the foreground and the background. However, in the case of historical documents, there exist degradations that diminish robustness of this class of binarization. Examples of degradations include shadows and non-uniform illumination, deformed and connected dots, ink seeping, smear and strains. To deal with degradations, the preprocessing techniques are used to minimize the degraded quality of the Braille documents. Therefore, the Otsu's global threholding technique is used for this study. The Otsu's algorithm is presented below [41]:

Let the pixels of a given picture be represented in L gray levels [1, 2, ,L]. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + + n_L$*. In order to

simplify the discussion, the gray-level histogram is normalized and regarded as a probability distribution [41]:

$$P_i = n_i/N,\ p_i > 0,\ \sum_{i=1}^{L} p_i = 1 \qquad (1)$$

Now suppose that we dichotomize the pixels into two classes C_O and C_1 (background and objects, or vice versa) by a threshold at level k; C_O denotes pixels with levels [1, , .k], and C_1 denotes pixels with levels [k + 1, , L]. Then the probabilities of class occurrence and the class mean levels, respectively, are given by:-

$$\omega_o = P_r(C_O) = \sum_{i=1}^{k} p_i = (k) \qquad (2)$$

$$\omega_1 = P_r(C_i) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k) \qquad (3)$$

and

$$\mu_0 = \sum_{i=1}^{k} i\, p_r(i/C_O) = \sum_{i=1}^{k} ip_i/\omega_o = \mu(k)/\omega(k) \qquad (4)$$

$$\mu_1 = \sum_{i=k+1}^{L} i\, p_r(i/C_1) = \sum_{i=k+1}^{L} ip_i/\omega_1 = \mu_T - \mu(k)/1 - \omega(k) \qquad (5)$$

where:

$$\omega(k) = \sum_{i=1}^{k} p_i \quad \text{and} \qquad (6)$$

$$\mu(k) = \sum_{i=1}^{k} ip_i \qquad (7)$$

are the zeroth- and the first-order cumulative moments of the histogram up to the k^{th} level, respectively, and

$$\mu_T = \mu(L) = \sum_{i=1}^{L} ip_i \qquad (8)$$

is the total mean level of the original picture. We can easily verify the following relation for any choice of k:

$$\omega_o \mu_o + \omega_1 \mu_1 = \mu_T, \quad \text{where } \omega_o + \omega_1 = 1 \tag{9}$$

The class variances are given by

$$\sigma_o^2 = \sum_{i=1}^{k} (1-\mu_o)^2 P_r(1/\mathrm{C}_o) = \sum_{i=1}^{k} (1-\mu_o)^2 P_1 / \omega_o \tag{10}$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} (1-\mu_1)^2 P_r(1/c_1) = \sum_{i=1}^{k} (1-\mu_1)^2 P_1 / \omega_1 \tag{11}$$

In order to evaluate the "goodness" of the threshold (at level k), the following discriminant criterion measures (or measures of class separability) are used in the discriminant analysis

$$\lambda = \sigma_B^2 / \sigma_w^2, \qquad k = \sigma_T^2 / \sigma_w^2, \qquad \eta = \sigma_B^2 / \sigma_T^2, \tag{12}$$

Where

$$\sigma_w^2 = \omega_o \sigma_o^2 + \omega_1 \sigma_1^2 \tag{13}$$

$$\sigma_B^2 = \omega_o (\mu_o - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 \tag{14}$$

$$= \omega_o \omega_1 (\mu_1 - \mu_o)^2$$

(due to (9)) and

$$\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 p_i \tag{15}$$

are the within-class variance (equ.13), the between-class variance (equ.14), and the total variance of levels (equ.15), respectively. Then our problem is reduced to an optimization problem to search for a threshold k that maximizes one of the object functions (the criterion measures) in equation (12). This standpoint is motivated by a conjecture that well thresholded classes would be separated in gray levels, and conversely, a threshold giving the best separation of classes in gray levels would be the best threshold. The discriminant criteria maximizing λ, k, and η, respectively, for k are, however, equivalent to one another; e.g., k = λ + 1 and η=λ/(λ + 1) in terms of λ, because the following basic relation always holds:

$$\sigma_w^2 + \sigma_B^2 = \sigma_T^2 \qquad (16)$$

It is noticed that σ_w^2 and σ_T^2 are functions of threshold level k, but σ_T^2 is independent of k. It is also noted that σ_w^2 is based on the second-order statistics (class variances), while σ_B^2 is based on the first-order statistics (class means). Therefore, η is the simplest measure with respect to k. Thus we adopt η as the criterion measure to evaluate the "goodness" (or separability) of the threshold at level k.

The optimal threshold k* that maximizes t, or equivalently maximizes σ_B^2 is selected in the following sequential search by 6 using the simple cumulative quantities (6) and (7), or explicitly using (2)-(5):

$$\eta(k)=\sigma_B^2(k)/\sigma_T^2 \qquad (17)$$

$$\sigma_B^2(k)=[\mu_T\omega(k)-\mu(k)]2/\omega(k)[1-\omega(k)] \qquad (18)$$

And the optimal threshold k* is

$$\sigma_B^2(k^*)=\max_{1\le k<L} \sigma_B^2(k) \qquad (19)$$

From the problem, the range of K over which the maximum is sought can be restricted to

$S^*=\{k;\omega_o\omega_1=\omega(k)[1-\omega(k)]>0, \text{ or } 0<\omega(k)<1\}$.

We shall call it the effective range of the gray-level histogram. From the definition in (14), the criterion measure $\sigma_B 2$(or η) takes a minimum value of zero for such k as k ε S - S* = {k;ω(k) = 0 or 1} (i.e., making all pixels either C_l or C_O, which is, of course, not our concern) and takes a positive and bounded value for k ε S*. It is, therefore, obvious that the maximum always exists.

4.2.4 Morphological operations

Morphological operations are performed after thresholding. This is because the global thresholding operations will include some dots that are not part of the Braille content. So morphological operations remove these noise that are greatly affect the recognition process. Morphological operations are simple mathematical constructs which have led to effective solution for many problems in image processing and computer vision. Dilation and erosion operations are the base for other morphological operations by special case of these operations or cascading applications of them.

Mathematically, dilation is defined in terms of set operations. The dilation of A by B, denoted A⊕B, is defined as [24]

$A \oplus B = \{z | (\hat{B})z \cap A \neq \varnothing\}$

Where ∅ is the empty set and B is the structuring element. In other words, the dilation of A by B is the set consisting of all the structuring element origin locations where the reflected and translated B overlaps at least some portion of A. the translation of the structuring element in dilation is similar to the mechanics of spatial convolution.

The mathematical definition of erosion is similar to that of dilation. The erosion of A by B, denoted $A \Theta B$ is defined as [24]

$$A \Theta B = \{z | (B)_z \cap A^c \neq \varnothing\}$$

In other words, erosion of A by B is the set of all structuring elements origin locations where the translated B has no overlap with the background of A.

The MATLAB function *bwmorph* implements a variety of useful operations based on combination of dilation, erosions and look up table operations [21].

4.3 Image Segmentation

In the Braille Recognition System, the next step after binarization is segmentation. Segmentation refers to the process of separating dots from Braille image that can further be grouped into a cell. For the Braille recognition problem domain, the point of interest or the low level of abstraction to be extracted is the single dot that could have six alternative positions in a cell.

Touched, overlapped, separated, and broken characters are major factors for causing segmentation errors [49] [59]. To accomplish this, as we have discussed in section 4.2, we apply image preprocessing before it, and then the mesh grid segmentation adopted by Teshome [51] is used for this study. Grid Mesh can be global or adaptive. In global grid, the horizontal and vertical grids are constructed based on a fixed global threshold. Whereas, in adaptive mesh the threshold is determined locally for each dots that runs vertically and horizontally. To determine the size of dots, analysis of the dots profile is important. To this end, each logical line of the Braille is extracted by constructing the horizontal projection to determine the height of a single dot, which would be used while constructing the horizontal grids. After line segmentation, vertical projection is performed that result in half-character segmentation (one column of a cell). Then, based on the above information (spacing, dot height and width), a mesh is constructed.

The algorithm first finds optimal starting dot by searching black pixel column wise and start to count if black pixel is found until the next white pixel is found; and do the same for raster line horizontally. The resulting value checked for threshold and then used to complete the horizontal and vertical grids.

Once the mesh is constructed, segmentation extracts the dot point from the image. The segmentation procedure identifies the foreground, from the back ground based on previously constructed mesh. Finally, the algorithm convert the whole image into black color and set the dot coordinates value with one pixel of white color.

The technique is preferred since Braille cells follow strict layout in horizontal and vertical line and does not have unique structure to isolate one combination from the others. Besides segmentation with mesh-grids bring good result for most of the Braille research.

4.4 Feature Extraction

Feature extraction is a representational mechanism of the Braille image. At this stage the identity of each of the Braille detected is recognized. Since each character is distinguished from the rest by its unique combination of dots. Then, the character can be described in terms of the numbers of dot positions. The Amharic Braille characters are represented by one, two or three Braille cells; it means a cell can be part of a character or character alone. To this end context based analysis is used for this study that is adopted by Teshome [51].

The algorithm works by taking the advantages of the mesh constructed in the segmentation phase and it following the mesh access dots on the coordinate point and group into cell. Then, the resulting grouped dots of a cell are checked for its content against the rules defined. The context analysis is used to determine the status of the cell.

4.5 Classification

Image classification is a fundamental problem in pattern recognition. Pattern recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest, make sound and reasonable decisions about the categories of the patterns. Patterns are any entity or object.

Hence, the aim of pattern recognition is the design of a classifier, a mechanism which takes features of objects as its input and which results in a classification or a label or value indicating to which class the object belongs. This is done on the basis of the learning set- a set of objects with a known labeling. The classifiers performance is usually tested using a set of objects independent of the learning set, called the test set [52][7]. For classification purpose the neural network is used for this study.

4.5.1 Training a Neural Network

Training a neural network is the process of setting the weight on the inputs of each of the units in such a way that the network best approximates the underlying function. Training basically involves feeding training samples as input vectors through a neural network, calculating the error of the output layer, and then adjusting the weights of the network to minimize the error. Back-propagation is the most commonly used method for training multilayer feed forward neural network. This training scheme is used for adjusting the connection weight of each unit in such a way that the error between the desired output and the actual output is reduced. More clearly, the neural network adjusts the connection weights to each unit, beginning with the connection between the last hidden layer and the output layer. After the network has made adjustment to this set of connection, it calculates error values for the next pervious layer and makes adjustments.

To train the neural network the following algorithm is implemented:

1. Form network according to the specified topology parameters
2. Initialize weights with random values within the specified ± weight bias value
3. Load trainer set files (both input image and desired output text)
4. Analyze input image and map all detected symbols into linear arrays
5. Read desired output text from file and convert each character to a binary Unicode value to store separately
6. for each character:
    a. calculate the output of the feed forward network
    b. compare with the desired output corresponding to the symbol and compute error
    c. back propagate error across each link to adjust the weights
7. Move to the next character and repeat step 6 until all characters are visited
8. Compute the average error of all characters
9. Repeat steps 6 and 8 until the specified number of epochs
    a. Is error threshold reached? If so abort iteration
    b. If not continue iteration

4.5.2 Neural Network Learning Laws

As learning in artificial neural networks is the process of adjusting the connection strength between successive neurons, there should be some predefined law on how to adjust the connection weights. Learning laws describe the weight vector for the i^{th} processing unit at time instant (t+1) in terms of the weight vector at time instant (t) as follows [9]:

$W_i(t+1)=w_i(t)+\Delta w_i(t)$

There are different methods for implementing the learning features of the neural network. Some of the learning laws are discussed below. All of these learning laws use only local information for adjusting the weight of the connection between two units.

4.5.2.1 Hebb's Law

The law states that the weight increment is proportional to the product of the input data and the resulting output signal of the unit. This law requires weight initialization to small random values around $w_{i,j}=0$ prior to learning. This law represents an unsupervised learning.

4.5.2.2 Perceptron Learning Law

This law is applicable only for bipolar output function f(.). This is called discrete perceptron learning law. The expression for $\Delta w_{i,j}$ shows that the weights are adjusted only if the actual output s_i is incorrect. This is a supervised learning law, as the law requires a desired output for each input. In implementation, the weights can be initialized to any random initial values, as they are not crucial. The weights converge to the final value eventually by repeated use of the input-output pattern pairs, provided the pattern pair are represent able by the system.

4.5.2.3 Delta Learning Law

The law is valid only for a differentiable output function, as it depends on the derivative of the output function f(.). It is a supervised learning law since the change in the weight is based on the

error between the desired and the actual output values for a given input. Delta learning law can also be viewed as a continuous perceptron learning law.

In implementation, the weights can be initialized to any random values as the values are not very crucial. The weights converge to the final values eventually by repeated use of the input-output pattern pairs. The convergence can be more or less guaranteed by using more layers of processing units in between the input and output layers. The law can be generalized to the case of multiple layers of a feed-forward network.

Chapter Five

Experimentation

5.1 Introduction

The main aim of the present research is to enable the Braille recognition system recognize noisy Braille by applying different preprocessing techniques. For this purpose, a program is written for preprocessing (noise detection and removal, interpolation, and binarization) using MATLAB. Then, using different test cases, first, the efficiency of each algorithm is tested individually and then, both noise detection and removal, binarization, segmentation and feature extraction algorithms are integrated together and tested on actual data to evaluate the performance of the system on Amharic Braille.

5.2. General Design of Amharic OBR System

The overall architecture of the Amharic Braille recognition system is shown in figure 5.1. The double rectangle is the main focus of this study.

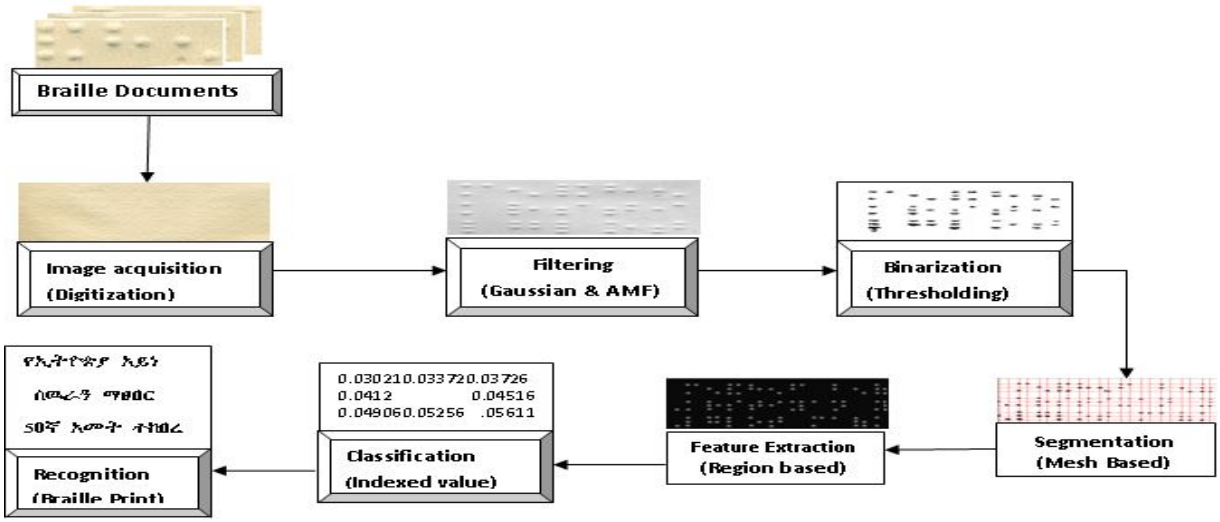


Figure 5.1 Block Diagram of the Amharic Braille Recognition system

The system is able to recognize single sided Amharic Braille documents. As shown in figure 5.1, the Amharic OBR passes through a series of steps. First, the real world Braille documents are collected and an image of the Braille document is acquired using a flatbed scanner with 200 dpi and then the image is converted to gray scale, if it is colored. Second, preprocessing techniques (like filtering , adaptive histogram equalization, morphological operations, thresholding) are applied in the gray scale image; both Gaussian and adaptive median filtering algorithms are tested. The Otsu's global thresholding techniques with some modification is applied to binarize the image. The binarized Braille images are feed to the previously developed segmentation and feature extraction techniques with some modification. Artificial neural network is used to classify the extracted Braille characters, based on which Amharic Braille images are converted to their equivalent textual format.

5.3 Data Collection

Braille documents have been collected mainly from Misrach center and AAU Kennedy library. The documents collected include manually produced, typewritten and using Braille printer and the Braille can be plastic and/or paper sheet, both are collected. In addition, the size of the page varies between documents produced by different means. For the purpose of this study the standard sized Braille, 11 x 11.5 inch, which is dominantly used at different center, is selected. As indicated in the literature, Braille can be produced on one side or double side of the document. For the reason that, most Amharic Braille documents are single-sided, the study is limited to investigate single-sided embossed Braille sheet. The details of sample Braille document collected for experimentation are given in Table 5.1.

Braille property	Description
Braille sheets	10 sheet
Total number of characters	4500
Mean number of character per sheet	450
Digital format	Gray scale/Color
Resolution (horizontal and vertical)	200 dpi
Image size Kbytes(on average)	2.75MB
Braille type	Single sided – grade 1
Image format Bitmap (‘bmp’)	Bmp
Document size(Horizontal x vertical)	11inches x 11.5inches

Table 5.1 Summary of Amharic Braille documents collected for the study

5.4 Braille Digitization

The current research used a flatbed scanner at 200 dpi resolutions for the digitization of the Braille documents. The 267 characters from the fourth version Amharic Braille are used for training and from the collected document, 200 characters for each image type (i.e. clean, small noise, medium and high noise) are digitized for testing purpose. The digitized image is saved in both gray and colored (RGB) window bitmap format. Sample digitized images are depicted in figure 5.2.

Scanning Braille is not a simple task. Above all, there is a need to take care in placing the hard copy document in the scanner for capturing an image. If the hard copy is placed carelessly, the digitized image may be tiled. Since the recognition of tiled image is a difficult task, tilt correction is necessary to make the image line horizontal. However, incorporating technique for tilt correction is not the scope of the present research; an effort is made to produce an image line which is strictly horizontal. In fact the scanner that is used in the present study allows to see the view of the image for further adjustment before obtaining the final output. In light of this, there is no problem of tiled image observed in the current study that affects the recognition process.

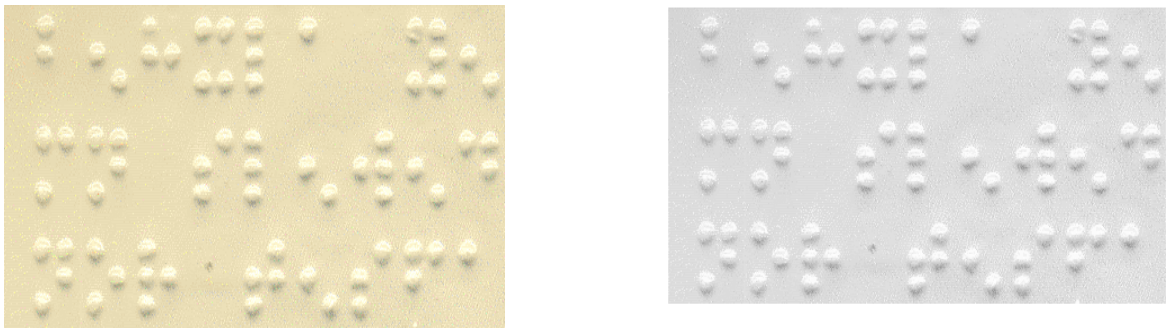


Figure 5.2 Braille document image (left with gray level scale and right with color)

The distribution of noise in the Braille images are not equal. Because some of the Braille images are stored properly and less used on the other hand some of the images are not properly sored, bend, and frequently used. Therefore, classifying the images based on the noise level is relevant to measure the performance of the system. Due to this reason the digitized Braille images are classified into four groups based on visual criteria of the noise level on the binarization stage Before applying any preprocessing technique (see figure 5.3).

a) clean image- image without noise,

b) low noise image- the images have slight distributed noise(like salt and pepper), the dots are not connected and we can clearly identify the dots from background, and the distribution of noise may or may not be equal,

c) medium noise image have the following characteristics: due to repetitive use and bend the pixels of the dots are connected, it consider the added noise as dots and have evenly or unevenly distribution of high noise but still we can identify dots from background,

d) high noise image have extremely used Braille and have unevenly distribution of high noise, due to this it is difficult to classify dots from background.

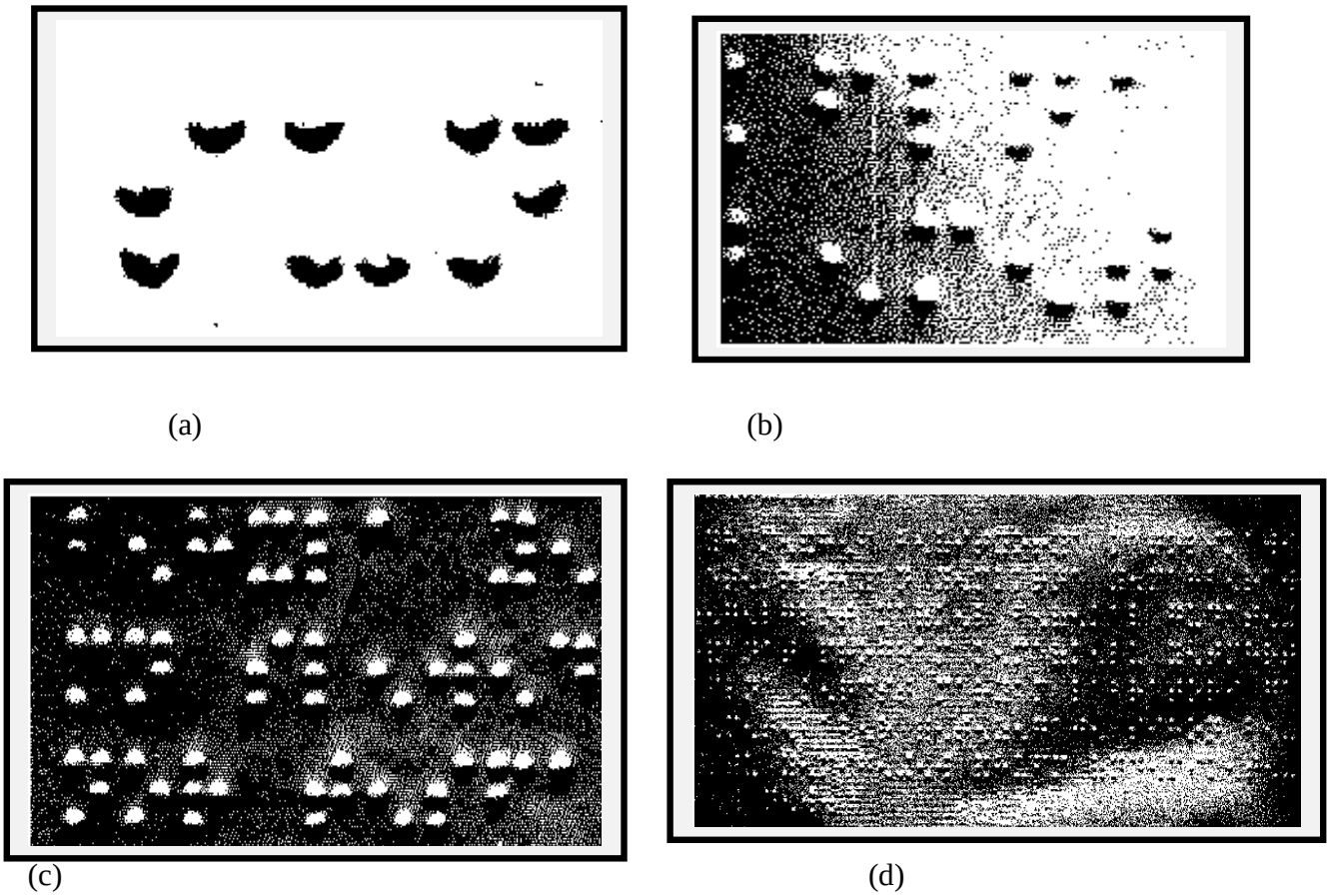


Figure 5.3 Image type by noise level: a) clean image b) low noise image c) medium noise image d) high noise image

After the required scanned bitmap image is produced, subsequent character recognition process such as preprocessing (like converting the image to gray color, noise detection and removal, binarization/threseholding) , segmentation, and feature extraction and detection are undertaken. To this end the digitized image is loaded into the memory of the computer. The program to load the image is written with MATLAB built in functions. The following is the source code to load it:

```
Brimg= imread('braille1.bmp');  %read and load the Braille image
```

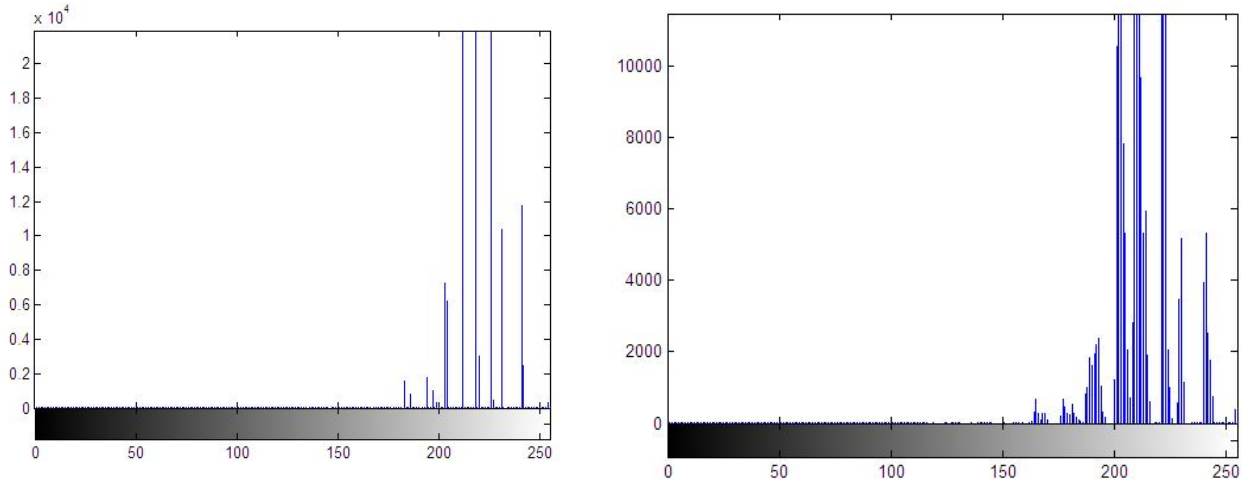
5.5 Preprocessing

Preprocessing helps to improve recognition accuracy of many optical Braille recognition systems because it reduces unwanted complexities of an image. In this work we apply histogram equalization, noise detection and removal, and binarization/thresholding preprocessing techniques. A pre-processing stage of the grey scale or colored source image is essential for historical and degraded documents for the elimination of noisy areas, smoothing of background texture as well as contrast enhancement between background and text areas.

The Braille image is noisy due to repetitive use, bend, resolution of the scanned image and paper quality, and in addition to this, during image acquisition impulse noise may be introduced in the image. These noises generally manifest itself a random fluctuation in gray-level values superimposed upon the ideal gray level value, and it usually of low amplitude, which nevertheless can degrade significantly the appearance of the image. Therefore, to reduce such noise we apply adaptive histogram equalization, a Gaussian and adaptive median filter, Binarization and morphological operations on the Braille image.

5.5.1 Adaptive histogram equalization

Adaptive histogram equalization enhances the contrast of the intensity image by transforming the values of the pixels. It operates on small regions, called tiles, in the image rather than the entire image (like histogram equalization). For each tile, the histogram is adjusted by using histogram equalization then the neighboring tiles are then combined using bilinear interpolation to eliminate artificially induced boundaries. The MATLAB implementation of the adaptive histogram equalization by using a built in function ‘adpthisteq’. This functions accepts the input image and generates the result image with the shape. The distribution of pixels in the original image and after applying adaptive histogram equalization is presented in figure 5.4 (a) and (b) respectively.



a) b)

Figure 5.4. Distribution of pixels using histogram a) original image b) the result of adaptive histogram equalization

The result shows that the high variation of neighborhood pixels in the image is adjusted. This greatly improves the performance of the filtering operations. The impact of adaptive histogram equalization on Gaussian filtering and AMF in is presented in section 5.5.4.

5.5.2 Gaussian filtering

The Gaussian smoothing method is widely used not only for smoothing signals of one independent variable but also for various image processing applications. It is a very good filter for smoothing signals or images. The amount of smoothing depends on the value of the spread parameter (i.e., the standard deviation) of the Gaussian function. It works by sampling a local neighborhood of pixels and producing a weighted average. The size of the neighborhood is called the kernel size. The Gaussian output a weighted average of each pixel's neighborhood, with the average weighted more towards the value of the central pixels.

The MATLAB implementation of Gaussian filtering by using a built in function 'fspecial'. This function accepts the image, a standard deviation and kernel value. The default value of 0.5

standard deviation and 3x3 matrix kernel. The command used to implement Gaussian filter and the result of the filtering is presented in figure 5.5 and 5.7(a) respectively.

```
gaussian=fspecial('gaussian',[3,3],0.5);

gaussianfilter=imfilter(img1,h,'replicate');
```

Figure 5.5 MATLAB code for Gaussian filter

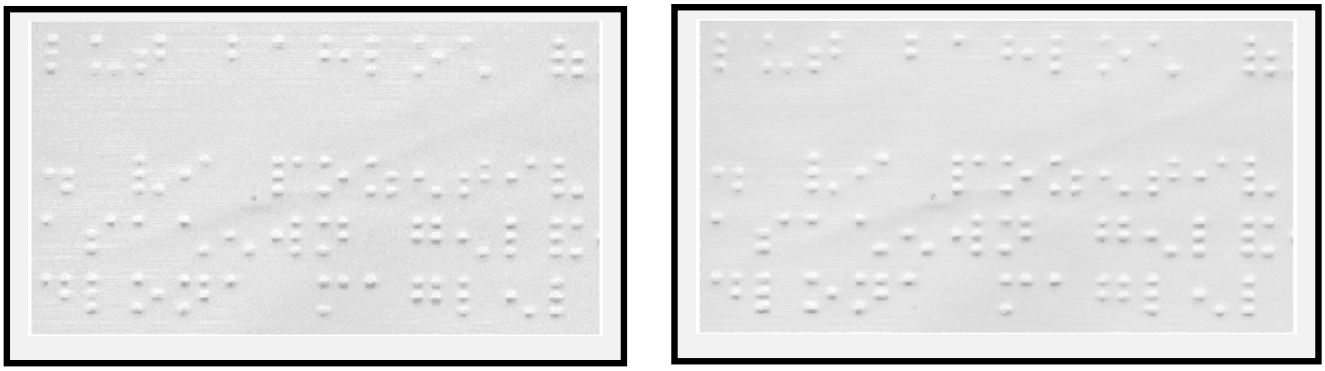
5.5.3 Adaptive Median Filtering (AMF)

Adaptive filtering has been applied widely as an advanced method compared with standard median filtering. Adaptive filter performs spatial processing to determine which pixels in an image have been affected by impulse noise. The Adaptive filter classifies pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable, as well as the threshold for the comparison. A pixel that is different from a majority of its neighbors, as well as being not structurally aligned with those pixels to which it is similar, is labeled as impulse noise. These noise pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test. Generally the purpose of adaptive filtering are removing impulse noise, reduce distortion, like excessive thinning or thickening of object boundaries and used to enhance the contrast and brightness of an image. For this study the Wiener2 adaptive MATLAB filtering method/function has been used. The filter has been proved efficient for the aforementioned goal. The Wiener filter is commonly used for gray scale image restoration. The MATLAB code for the Wiener2 adaptive filtering is shown in figure 5.6.

```
info=imfinfo('C:\\MATLAB7\\work\\test\\Br6.bmp');
ctype=[info.ColorType];  % To get information about the color type of image
if strcmp(ctype,'truecolor')  % compaire the image with color value
        img1=rgb2gray(img);  % convert the colored image into gray
else
        img1=img;
end;
img2=wiener2(img1); % applying adaptive median filtering from the gray scale image
```

Figure 5.6 The MATLAB code for adaptive median filtering

Our pre-processing module involves an adaptive Wiener method based on statistics estimated from a local neighborhood around each pixel figure 5.7(b) shows the results of applying a wiener filter to a document image.



(a) *(b)*

Figure 5.7 Results of the filtered image a) Gaussian filtering b) Adaptive median filtering

The performance evaluation of the filtering operation is quantified by the PSNR (peak signal to noise ratio) calculated using formula [28]:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - I'(i,j)]^2$$

and

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})$$

Where, M and N are the total number of pixels in the horizontal and the vertical dimensions of the image. I and I' denote the original and filtered image, respectively. A small value for MSE means less error in the new image. Larger PSNR values signify better signal restoration because it means that the ratio of signal to noise is higher. Here, the signal is the original image, and the noise is the error in reconstruction, so, if you find a filtered scheme having a lower MSE (and higher PSNR), you can recognize that it is better one [44].

The Gaussian and adaptive median filter is applied on the different noise level of Braille images (small, medium and high). The comparative results of AMF and Gaussian filtering with 3x3, 5x5 kernels and 0.5 standard deviation the results are presented in Table 5.2.

Image noise type	MSE		PSNR	
	AMF	Gaussian filter	AMF	Gaussian filter
Small	66	0	29.94	48.13
Medium	77	0	29.27	48.13
High	119	0	27.36	48.13

Table 5.2 PSNR and MSE measure of Braille images

From the above results we can say that the performance of Gaussian filtering is better than adaptive median filtering in terms of PSNR and MSE. However, when we visually investigate the results of the filtered Braille image, the Gaussian filtering algorithm blurs the image, where

as the adaptive median image filtering algorithms results in no blurring effect in the image and also preserves the edge.

5.5.4 Binarization/Thresholding

Image binarization converts the gray scale image into binary (black and white) image aiming to distinguish text areas from background areas. Binarization plays a key role in document processing and its performance affects the degree of success in the subsequent segmentation and recognition phases [6]. In setting threshold, a preliminary analysis of relevant picture statistics with a series of experiments is performed in which the threshold image is examined as the threshold is adjusted, and the best result is ascertained visually. Visual examination of the resulting image show that with color image there is a possibility of losing valid dot point as the dot size is relatively small and the noise level is relatively minimized. This requires advanced image preprocessing techniques [51]. However, image preprocessing (such as histogram equalization, Gaussian filtering, adaptive median filter) is applied before this phase (as we have discussed in section 4.2) to restore some of the dots by avoiding the noise. Hence, in this study both colored and gray-level images are considered for further processing.

The results of the thresholding after applying adaptive median filtering with out adaptive histogram equalization, as presented in figure 5.8 (a),shows that the image dot sizes have great variation and small in size, and also some of the dots are remove. However, the results of Gaussian filtering shows that, see figure 5.8(b), it adds more dots to the binarized images and has more noise, and the dots sizes are relatively better than AMF. Therefore, inorder to tackle these added dots and noise we used adaptive histogram equalization before applying filtering as we have discussed in section 5.5.1. Adaptive histogram equalization restores some of the deleted dots images and equalizes the dots for adaptive median filtering, however, in the case of Gaussian filter it greatly increases the noise level by restoring the deleted dots and also increase the dot sizes. To remove these (in the case of Gaussian filtering) noises (figure 5.8 (d)) we Perform morphological operations on images. The result of Gaussian and adaptive filter with histogram equalization is presented in figure 5.8 (c) and (d).

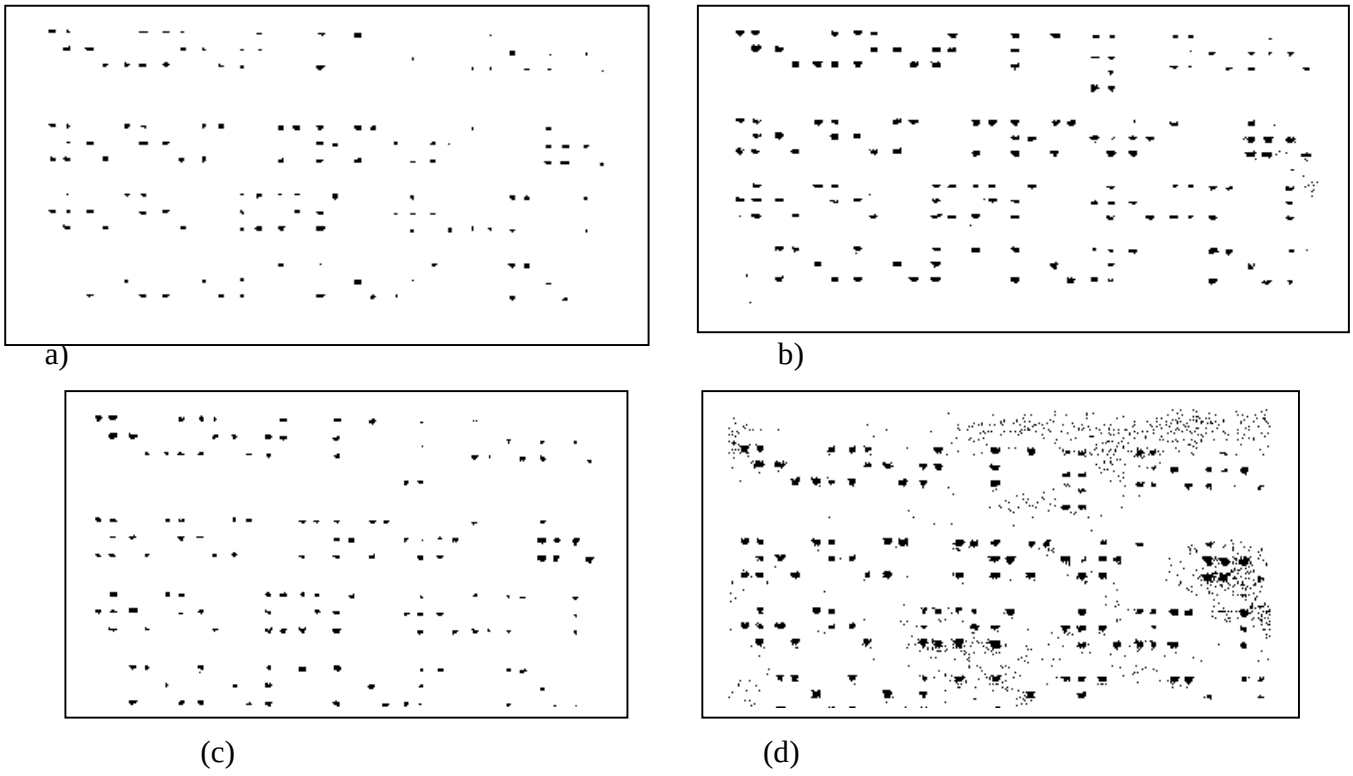


Figure 5.8 The result of adaptive histogram equalization a) thresholding AMF without histogram equalization b) thresholding Gaussian filtered image without adaptive histogram equalization c) thresholding AMF image with adaptive histogram equalization d) thresholding Gaussian filtered image without adaptive histogram equalization

For this study, we apply the Otsu's global binarization technique on the gray scale filtered image. As shown in figure 5.9 (a), some parts of the Braille image is difficult to identify the dots from the background and some of the dots also connected. This problem is attacked by using the Otsu's global thresholding technique. Increasing or decreasing the Ostu's algorithm diminish the generated noise. So after extensive experimentation decreasing the threshold value by 0.065 from the Otsu's global threshold value generates a better Braille image, as presented in figure 5.9 (b). This image also have some noise on the edges of the document. This is due to the fact that the edge of the Braille is the most touchable parts. To eliminate these edge noises we cut four pixels from the four sides of it. This techniques generates a better result without affecting the Braille dots (see figure 5.9 c).

(a) (b) (c)

Figure 5.9 Thresholding result: a) Otsu's global thresholding b) The new thresholding c) Edge enhancement

Part of the MATLAB code that performs the thresholding task to enhance the quality of image is presented in figure 5.10.

```
Threshold_level=graythresh(filtered_image);

Otsu_threshold=im2bw(filterd_image, Threshold_level);

Modify_threshval= Threshold_level -0.065;

New_thrshold=im2bw(filtered_image, modify_threshval);
```

Figure 5.10 MATLAB code for thresholding Braille image

The new global thrsholding techniques produces a satisfactory results for clean, small and medium level noise with some addition and deletion of dots. However, for high level noise, it generates the image by deleting some of the dots and also connecting the dots at the edge of the image. Generally, theresholding the Gaussian filtered image with additional preprocessing image performs better than thresholding the adaptive median filtering. Because thresholding the Gaussian filtered Braille images has good dot size and almost preserves all the dots for clean, small, and medium noise level images.

After the binarization, or sometimes called thresholding, the scanned image has the feature that the foreground (content of the image) is represented by black color and background with white color. The result of these image shows that still there is a noise in the document. To remove this

noise we analyze the pixel size of the dots in the image. Through experimentation we identify that a dot size less 20 pixels in the documents is a noise, therefore, we avoid the dots. The output of the binarization module is fed to the next level of processing, which is image segmentation.

5.5.5 Morphological operations

Morphology is a broad set of image processing operations that process images based on shapes, size, neighborhood. The morphological operation is performed based on a comparison of the corresponding pixel in the input image with its neighbors. The implementation of morphological operations is by using MATLAB built in function 'bwmorph' majority operations. The function sets a pixel value to 1 if five or more pixels in its 3x3 neighborhood are 1 otherwise sets the pixel value to 0. The complemented result of morphological operation on the noisy Gaussian filtered image is presented in 5.11.

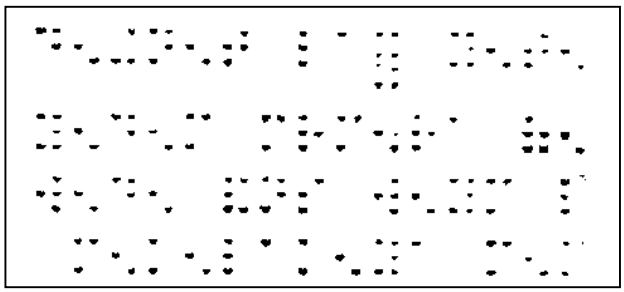


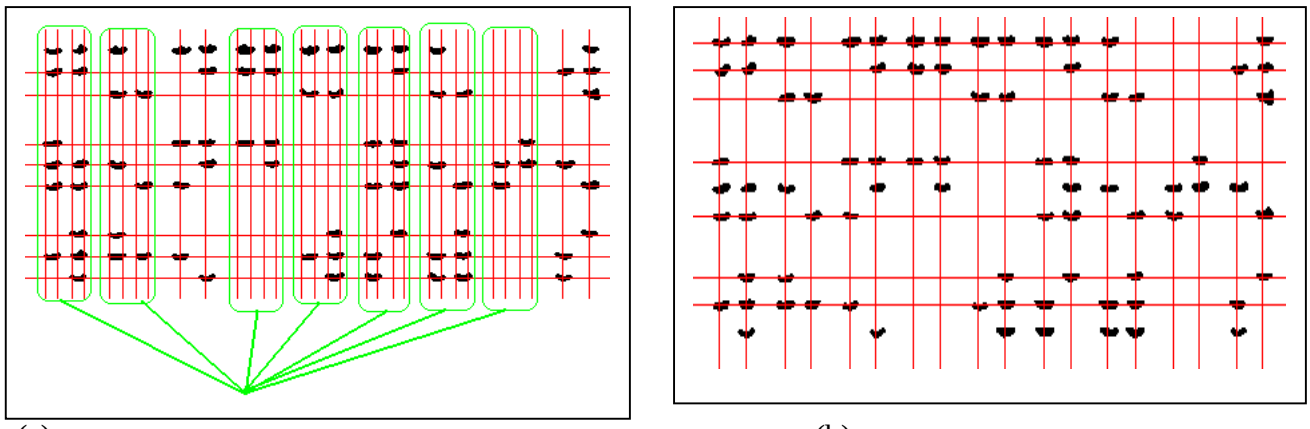
Figure 5.11 The result of morphological operation on the Gaussian binarized image

The result shows that the small noises in the images clearly removed. This image greatly improves the performance of the segemetation processes because the mesh grid segementation is constructed based on the positions of pixels in the image.

5.6 Segmentation

As indicated previously, segmentation refers to the process of separation of dots from Braille image that can further be grouped into a cell. These cells further grouped into character, words of any strokes in Braille. The output of this step is used as an input to feature extraction. Hence, an algorithm that correctly segments the Braille image into cells and among the cells is crucial in Braille recognition.

In the present research, mesh grid segmentation algorithms adopted by Teshome [51] is used. This algorithm is efficient in segmenting Braille image that are not connected or skewed with similar dot size. Segmenting real-life documents are done successfully since the current system is supported by noise filtering technique, which minimizes the noise and creates spacing between dots. However, the segmentation algorithm encounters challenges that are created by the real-life documents. One of the challenges is because of the fact that real-life documents have different Braille dot size due to repetitive use and the impact of filtering. The variation of the Braille dots greatly affects the construction of mesh grid (which is size dependent). Therefore, mesh grid segmentation algorithm is modified such that the mesh grid is constructed by finding the minimum and maximum dot size of the Braille dots.

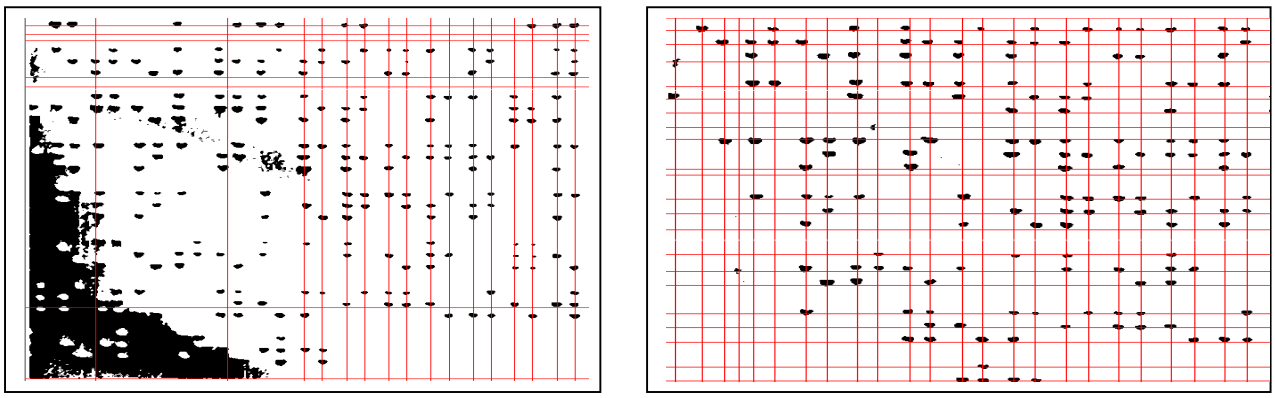


(a) (b)

Figure 5.12 Braille dots with a) Incorrectly segmented b) correctly segmented

The comparison of mesh grid constructed before and after modification of the algorithm is shown in figure 5.12. As shown in figure 5.12 (a) the algorithm adopted by Teshome [51] constructs two mesh grid for a single dot. By finding the minimum and maximum dot size the mesh grid depicted in figure 5.12 (b) is generated. The modification of the mesh grid is not work for all image therefore, the system to be efficient the mesh grid segmentation should be adaptive. A high level noisy image has a large noise at the edge of the image as shown in figure 5.13 (a). These noise is difficult to separate from the content. Constructing a mesh for such type of image is difficult because the mesh grid segmentation algorithm is a size dependent. Therefore to

handle such type of image, the first solution will be cutting such large dots from the document, the second solution will be using other noise filtering technique that is proposed by AbdulMalik et.al [2], the technique works by calculating the average gray level of the whole image, and then calculate for row and column separately. Finally finding a row or column average gray level that is above and below 15% of the whole image average gray level is an indication to delete it. From the two techniques we have use the first technique i.e. avoiding the large dots with contents and then construct the mesh grid for the image as shown in figure 5.13 (b).



(a) (b)

Figure 5.13 Mesh grid for high level noise a) with large noise b) after cutting the large noise

Cutting the large noise is not a best solution because it remove the Braille dots with noise. These greatly affects the performance of the system.

5.7 Feature Extraction

The main function of this phase is to extract the Braille dots from the segmented image and groups them into cells [12]. The extracted features are used for testing the developed neural network model. To this end, the present study is used context based feature extraction algorithm adopted by Teshome [51] with some modification. The modification is that the previous system store the extracted feature only for the current Braille image, however the current system appends the output from the previously extracted features. For this study all of the Amharic

Braille characters are not considered because except the basic Amharic characters with its six forms and Arabic numerals, most of the punctuation marks and Ethiopic numerals are rarely apply in the Braille writing [51]. The context analysis has performed to determine the status of dots in a cell. Based on the result, the cell can be recognized as Braille character alone or part of a Braille characters. This is because depending on the context, a Braille character may have one, two or three cells.

The failure of constructing vertical or horizontal mesh grid highly affects the feature extraction of the Braille document images. For example, adding or missing the construction of horizontal mesh grid affects a maximum of the whole features in the line. Therefore, the success of the feature extraction is highly depends on the segmentation phase. The context based feature extraction is successful for Amharic Braille recognition.

All the features of the Braille dots are extracted and represented with 6 and 12 bits that can turned on(1) or off(0) in any combination.

5.8 Braille Character Recognition

After extracting the features of the Braille dots they have been feed to the artificial neural network. This section describes how the artificial neural network is created, trained and refined to classify an input pattern(binary Braille characters) into one of the target output results (print characters). The overall task of the recognizer is to be able to classify new input to one of the possible output values. Since the neural network tool used for the recognizer is the MATLAB Neural Network tool Box, the discussion on how to design, create, train and test the network is presented below.

5.8.1 Artificial Neural Network Architecture

The Architecture of the neural network is the first task to create and train the neural network. In this study, the architecture of the neural network created for the recognition is a feed-forward, supervised, multilayer perceptron (MLP) network with three (3) layers – input layer, one hidden

layer and an output layer. Only one output neuron is used that accepts the binary representation of the segmented Braille character as an input vector. The binary representations of Braille character as input vector are created from Amharic Braille alphabet in two standards (6 bits and 12 bits) that have equal number of input nodes to the corresponding input layer of the network. The two inputs with two different standards are prepared representing the Braille character for training and testing the network.

The following graphical presentation gives the visual impression of the overall architecture of the recognition system starting from the Braille image input to the recognition phase.

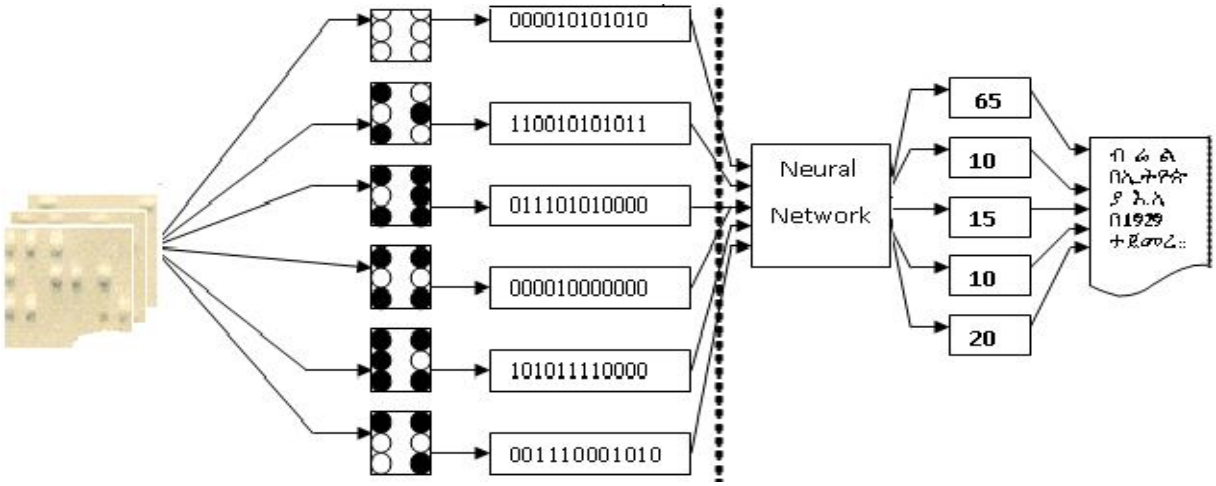


Figure 5.14 General Architecture of Braille Recognition (taken from Teshome[51])

5.8.2 Neural Network Classifier

A neural network is an adaptable system that can learn relationships through repeated presentation of data, and is capable of generalizing to new, previously unseen data. They are a large set of interconnected neurons, which execute in parallel to perform the task of learning. For this study we have used feed forward multilayer perceptron (MLP) model with back-propagation learning rule which is based on supervised learning. The approach adjusts the strength of connection between nodes at different layers after computing the error between the desired and the actual output of the network at each training iteration. The network is used a mean square

error to measure the errors of learning (i.e. the error between the desired and the actual output). If this value is found to be greater than the threshold the error will propagate back to adjust the weights of each connection in the course of finding the optimal weight for each connation.

The network is designed with two possibilities with respect to the number of nodes at the input layer: 6 and 12 nodes. The binary representation of the input character extracted and created in excel file is fed to the network after some rearrangement of the rows and columns of each patterns.

5.8.3 Braille Character Input Method for the Neural Network

To train the network the first task is rearranging the input pattern into column vector which contain elements equal to the number of neurons at the input layers. For this purpose, the binary representation of the characters extracted during the preprocessing stages, which is created in a file as 6 and 12 bits pattern of Braille character, is converted into one column vector where each preceding column of the character matrix are appended one after the other, see table 5.3.

1	1	1	1	0	1	1 . . .
1	1	1	0	1	1	0 . . .
0	1	0	1	1	1	0 . . .
0	0	0	1	1	0	1 . . .
1	0	0	0	0	1	1 . . .
0	0	1	0	0	0	1 . . .

(a)

1	1	1	1	0	1 . . .
1	1	1	0	1	0 . . .
0	1	0	1	1	0 . . .
0	0	0	1	1	1 . . .
1	0	0	0	0	1 . . .
0	0	1	0	0	1 . . .
0	1	0	1	1	0 . . .
1	0	1	0	0	0 . . .
0	1	0	0	0	0 . . .
0	0	1	0	0	0 . . .
0	0	0	1	0	0 . . .
1	1	0	0	0	0 . . .

(b)

Table 5.3 Sample input for the neural network (a) with 6 input (b) with 12 input node

The number of nodes in the network layer is decided to be one because each Braille character has a unique representation of binary values. However, preparing output for the network is a challenging task because the Amharic characters are not represented with a unique ASCII (see table 5.4), adding the values resulted in the same value for different characters, this creates confusion in the output. On the other hand, even if representing characters with Unicode generates a unique value but the value has a combination of numbers with characters, such type of representation does not handle by the network. So, to come up with uniform mapping, the character is assigned index starting from 1 to 34, and 1 to 267 for the four different input set and this value is converted between the value 0 to 1 to fit the network output with Min-max normalization formula.

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A$$

Amharic characters	Latin representation	Latin ASCII code		Amharic ASCII code
X¦	X@	88	64	152
~	~	152		152
u?	u?	117	63	180
´	´	180		180
%<	%<	37	60	97
a	a	97		97

Table 5.4 Sample Amharic characters with the same ASCII value

5.8.4 Creating the neural network

The network created is a feed-forward error back-propagation where the type of training is gradient decent with momentum and adaptive learning rate. These two parameters are important for the creation of the network because learning rate is used to specify how to make adjustment to the weights after calculating the errors on the output layer. While, the momentum parameter is used to adjust the speed of the network for faster convergence. The principle is that, once the direction of convergence of the learning model is known moving fast towards the minima point speedup the learning process which helps to minimize the total time required by the network to learn different patterns. In addition to fast convergence, the momentum constant help the

network not to get stuck at local minima. A local minimum is a point in the learning curve where the network error is the minimum only to points in its vicinity and not to the total learning curve.

Based on the above assumption, in MATLAB tool such facility to train the network is accessed with training function TRAINLM. This function updates weight and bias values according to Levenberg-Marquardt optimization. This algorithm appears to be the fastest method for training moderate sized feed-forward neural network (up to several hundred weights). In many cases, TRAINLM is able to obtain lower mean square errors than any of the other algorithms. This advantage is especially noticeable if very accurate training is required.

To create network based on TRAINLM using MATLAB, the command is presented in figure 5.15:

AmharicBrailleNetwork = newff(input_range,[12,1], {'tangsig', 'purelin'}, 'trainlm');

Where:

newff; is the neural network feed-forward function,
input_range: are the minimum and maximum values at the input nodes,
{'tansig', 'purelin'}; is the activation function at the hidden and output layers independently.
[12, 1]; is size of neurons at hidden and output layer respectively

Figure 5.15 MATLAB code for creating the neural network

4.8.5 Training the Neural Network

Once the network has been designed, the training input space prepared for the purpose is introduced to the network to achieve the intended target output. We train the network by optimizing the error function. This process determines the best set of weights and biases for our dataset. With the objective to consider different pattern (with one/two cells) Braille character and character type, four training dataset are prepared with the following proportion to train the networks:

Training dataset 1: contains 34 records for basic Amharic Braille character (that use one cell representation) and the corresponding print character code.

Training dataset 2: contains 238 records for basic Amharic character (including the variant 34 * 7), each character presented with 12 bit (2 cells) and the corresponding print character code.

Training dataset 3: contains 257 records for basic Amharic character (38 x 7) and punctuation marks (19), each with 12 bit (2 cells) and with the corresponding print character code.

Training dataset 4: contains 267 records for basic Amharic character (38 x 7), numerals and punctuation marks (19), each with 12 bit (2 cells) and with the corresponding print character code.

Once the network has been created and the training input space prepared the network is ready to be trained. Some issues that need to be addressed upon training the network are:

- How complex are the patterns for which we train the network? Complex patterns are usually characterized by feature overlap and high data size.
- What should be used for the values of learning rate and show
- How many Iterations (Epochs) are needed to train the network for a given number of input sets?
- What error threshold value must be used to compare against in order to prematurely stop iterations if the need arises?

For the purpose of this project the optimal parameters used are:

- Learning rate = 0.01
- Show=50
- Number of Epochs = 1500 (Maximum)
- Mean error threshold value = 0.0001

The line of instruction to perform the training is:

[AmharicBrailleNetwork,tr]=train(AmharicBrailleNetwork,TrainSet,TargetSet);

Where : AmharicBrailleNetwork is the neural network created in the previous section, ***TraingSet*** is matrix where each column represents one Braille character from the dataset and ***TargetSet*** is pattern of target output for each corresponding input.

Accordingly, to train the network, the four Braille character input file along with the corresponding target character file is given to the neural network. Then, it has been repeatedly trained with the original dataset for 50 iterations, which makes a total of 1500 for each character pattern in the training set.

4.8.6 Testing the Neural Network

After training the artificial neural network is accomplished, it is required to test the performance of the modeled network with different test cases. In testing the network, two approaches are implemented. The first one is testing the network with the pattern that the network is trained with. This testing scenario helps to see if the network can recognize the original training pattern correctly. The second approach is evaluating the network performance with new input patterns, i.e. extracted from the Braille document which prepared as 6 and 12 bits in files.

This approach is important to evaluate the network performance with new input patterns. The following line of instruction is used to test the network trained so far for the three training set.

Rst=sim(AmharicBrailleNetwork,TestSet);

Where: Rst is the output of the network, AmharicBrailleNetwork is the network created and trained for different patterns; TestSet is the testing input pattern

The two artificial neural networks created for two different numbers of input nodes with four training sets are tested for accuracy of classifying new character pattern to one of the 34 and 267 characters of the Amharic language. The output of the network could range between 0 and 1 because as we have described in section 5.8.3, the Amharic characters can't be represented by ASCII or the Unicode. Then, this value can be converted back into the indexed value in the range 1-34, and 1-267.

4.8.6.1 Test result using the training set

To see whether the network can recognize the character set used for training, all the four training dataset that contain 34, 238, 257 and 267 characters set is fed to the network independently. The

models are learn by using different learning rate, goal and epoch, and with the same show. The output of the two networks is presented in table 5.5.

Network	Performance/rate of recognition			
	Training set 1(34)	Training set 2(238)	Training set 3(257)	Training set 4(267)
With 6 input nodes	100%	-	-	-
With 12 input nodes	-	100%	97.67%	97.75%

Table 5.5 Network performance for the training set

The above table shows that training the network with first order basic characters sets i.e. the 34 basic Amharic characters results in 100% recognition accuracy. This means the network recognizes the basic characters without any miss. However, with this network we cannot test the other training sets (i.e. training set 238, 257, and 267) because there is an input node variation. Testing the network with 12 input nodes with the three training sets generates better results for training set 2. All basic characters with their variant characters are correctly recognized. For others (training set 3 and 4) almost the same level of performance is achieved. Characters that are not correctly recognized are shown in table 5.6.

Input pattern given												Expected Target	Result(error)
0	1	1	1	1	0	1	0	0	0	1	0	t	¬
1	0	1	0	1	1	1	0	1	0	0	1	z#	z!
1	1	1	1	0	1	1	0	0	0	0	0	É	É!
0	0	1	1	1	1	1	0	0	0	0	0	1	2
0	0	1	1	1	1	1	1	0	0	0	0	2	3
0	0	1	1	1	1	0	1	0	1	0	0	9	8

Table 5.6 Error type and list in validating the network

As presented in the table 5.5, the majority of the errors are punctuation marks. The reason behind is the high similarity of dot representation of punctuation marks.

From the above model 3, the model created using training set 4 is selected for further testing the system. The rationale is that this model is trained with training set contained in the other two models plus additional numerals set and also has the same number of errors with model training set 3.

This time the performance is tested with 12 node network because most of the Braille documents consist of a two cell Braille characters.

5.9 Performance evaluation of the Amharic OBR

To test the performance of the system, new testsets extracted from Braille document are used. This test helps to see the response of the system to new character input patterns that the classifier never faced during the training phase. As we have discussed in section 5.4 Braille documents are grouped into four noise levels: clean Braille, small noise Braille, medium noise Braille, and high noise level Braille. The model has been tested with a set of Braille characters for each of the four Braille types.

- Testset 1: contains a total of 200 characters with two cells for clean image.
- Testset 2: contains a total of 200 characters with two cells for small noise level Braille image.
- Testset 3: contains a total of 200 characters with two cells for medium noise level.
- Testset 4: contains a total of 200 characters with two cells for high noise level.

Accordingly, testset 1, testset 2, testset 3 and testset 4 have been submitted for recognition. The system is tested on the Braille images filtered with Gaussian filtering and adaptive median filtering. The performance of the neural network for the two filtering is presented in Table 5.7.

Test data set	Testset size	Performance	
		Gaussian	AMF
Testset1	200	95.5%	95.5%
Testset2	200	95.5%	95.5%
Testset3	200	90.5%	87.5%
Testset4	200	65%	58.5%

Table 5.7 Performance rates for test dataset

The performance of the system for testset 1 (clean Braille images) and testset 2 (small noise) is the same however, accuracy decreases from testset 2 (small noise Braille) to testset 4 (high noise Braille). This indicates that the system clearly removes small noise level from the Braille image like salt and pepper, Gaussian etc. The performance of the system decreases on testset 3 is because of the addition and deletion of Braille dots in the image. The reason for high performance reduction on testset 4 is addition and deletion of Braille dots, the impact of the repetitive use, bend and highly connected Braille dots of the Braille image. Due to this, both filtering techniques are ineffcient to restore the removed dots and separate the connected Braille dots. In addition to this, the mesh grid segmentation is not handle the large dots that are found at the edge of the Braille image.

In general, the decreasing performance of the system in both filtering techniques is that some characters found in the test document for which the network is not recognize in the learning process. This includes the substitution errors as presented in table 5.6.

Table 5.7 shows that the performance of the Gaussian is outperforms the adaptive median filtering. This is because Gaussian filtering is better to restore dots, and also the dots have high similarity of Braille dot sizes. Therefore, for real world Braille recognition system we recommend the use of Gaussian filtering with morphological operations.

There is a great variation of performance between high level noise with other (clean, small noise, and medium noise images). The reason is that the technique used to remove high connected noise with dots in the high level noisy image is cutting. This techniques removes the dots that are parts

of the image. Therefore, to improve the system the high level noise needs advanced noise removal techniques that clearly separate the noise with contents. In the Amharic Braille recognition system most of the errors are substitution and addition in the medium noise image because, the result of adding or deleting a single dot to the image results changing the characters to their seven forms or replaced by other characters however, in the case of high level noise there is deletion error in addition to addition and substitution because the connected noise dots in the image are removed by cutting.

During experimentation of the applicability of these algorithms and approaches, the adaptive histogram equalization method for the digitized image is found to work very well instead of global histogram equalization techniques/methods for the problem of interest.

Chapter Six

Conclusions and Recommendation

6.1 Conclusion

In this study an attempt has been made to recognize the noisy Amharic Braille document images. Braille has been a very effective means of written communication for the blind and the partially sighted people. The development of the Braille recognition system can solve the communication gap between sighted and visually impaired people.

Since most real life Braille document images are with poor quality, this research considers different preprocessing algorithms like interpolation, noise filtering, morphology operations, and global thresholding. The results of preprocessing stage enhances segmentation, feature extraction and the classification task.

When we compare the Braille documents with other vision society reading materials (documents), it is very exposed to noise due to figure reading system and poor quality of the paper. To handle such different types of noise, Gaussian filtering with morphological operation is better. Gaussian filter is robust in retaining the small Braille dots and remove Gaussian noise.

Test results show that there are problems of dot connectivity in some of the Braille documents. Hence, in attempt to develop generally applicable binarization algorithm, we considered by adjusting the Otsu's global thresholding algotitms. The newly developed algorithm is tested with different intensity noise level Braille in which case a remarkable result is obtained. The advantage of this algorithm is that it controls the connectivity of Braille dots.

The construction of the mesh grid segmentation depends greatly on the Braille dot size. The variation of the Braille dot size greatly affect the segmentation phase. Therefore, using similar dot size (normalized dots) or by finding minimum or maximum size dots (adaptive thresholding) gratly enhance the performance of the mesh grid segmentation algorithm.

For classifying different input patterns an artificial neural network classifier is used. The designed network is feed-forward back-propagation error correction scheme where desired and actual output errors are computed and used for refining the network.

The Amharic OBR system is tested to evaluate its performance using different noise level Braille image. The results vary from 96.5% (for clean image) to 65% (for highly noisy image) with the type of noise level. The severity of the noise increases the performance of the system is decrease. The most common error in Amharic Braille recognition system for medium noise level is addition and substitution where as in high level noise addition, substitution and deletion. However, in the case of clean and small noise image the main error is substitution due to the problem of neural network classifier.

6.2 Recommendations

This study attempts to develop an Amharic OBR that recognizes real life Braille documents. Future research in this area should look to the following issues,

1. Normalization techniques should be devised and integrated with present development in Amharic OBR, so that the system will be size independent.
2. This system uses properly scanned Braille images. The slanting of the Braille images affects the construction of the mesh grid segmentation. Therefore, incorporating skew/tilt detection and correction algorithm should be develop to enhance the performance of the system.
3. To enable the Amharic OBR system search for obvious errors and locate possible alternative for unrecognized words, there is a need to integrate post-processing techniques (with the help of spell checker, thesaurus, grammar, etc tools).
4. The Braille dot size can be affected by use, type of writing, and the preprocessing algorithms used. Since the previously adopted segmentation algorithms is not flexible enough, as expected, to make a generalized one for the recognition of different dot size, more flexible segmentation algorithm (such as adaptive mesh grid etc.) should be adopted.

5. Nowadays, the uses of double sided Braille are started due to the coming of Amharic Braille printer. The existence of this printer generates a huge amount of information. However, the previously developed OBR system only considers single side Braille documents. So, we propose to design an algorithm that recognizes a double sided Braille document.
6. In the Amharic Braille recognition system neural network has a substitution error in recognizing the characters. According to Million [35] the use of support vector machine (SVM) classifier is advantageous because of their generalization capability. Hence, using SVM is one of the recommendation to increases the performance of the system.
7. The Amharic Braille has a fixed features representation for each character. Therefore, using rule based classifier instead of neural network will improve the performance of the system.

Reference

1. A. Antonacopoulos and D. Karatzas, “Color text segmentation in web images based on human perception”, Image and Vision Computing, Vol. 25, PP: 564-577, 2007.
2. AbdulMalik Al-Salman, et. al., “An Arabic Optical Braille Recognition System”, Proceedings of the First International Conference in Information and Communication Technology & Accessibility. Hammamet, Tunisia, 2007.
3. Abraham A., “Artificial Neural Networks: Handbook of Measuring System Design”.,John Wiley & Sons, Ltd 2005.
4. Bender, M.L. et al., “Language in Ethiopia”, London, Oxford University. 1976.
5. Berihun Girma, The Educational Situation of the Blind: Center for Educational Staff Development (CESD), 1994.
6. B. Gatos, I. Pratikakis and S.J. Perantonis, “Efficient Binarization of Historical and Degraded Document Images”, IEEE computer society, The Eighth IAPR Workshop on Document Analysis Systems, pp 447-454, 2008
7. Bir Bhanu, Yingqiang Lin and Krzysztof Krawiec, “Evolutionary Synthesis of Pattern Recognition Systems”, Springer Science Business Media, Inc., 2005
8. Bishop C. M., Neural Network for Pattern recognition. Oxford University Press.
9. B.Yegnanarayana, “Artificial Neural Network”, Prentice Hall of India Private Limited New Delhi, 2006.
10. Chapman S.T., Essential of MATLAB Programming, Volume 10, U.S. Student Edition, USA, 2008
11. Clutha Mackenzie, “World braille usage: A survey of efforts towards Uniformity of Braille notaion”, Poor l’Oragnization des Nations Unies pour l’Education, la Science et la Culture, de l’Imprimerie, Pairs, 1953.
12. C. Ng, V. Ng and Y. Lau, “Regular feature extraction for recognition of Braille”, In Proceedings of Third International Conference on Computational Intelligence and Multimedia Applications, ICCIMA'99, pp. 302-306, 1999.

13. C. Ng, V. Ng and Y. Lau, “Statistical Template Matching for translation of Braille”, In Proceedings of the Spring Conference on Computer Graphics (SCCC 1999), pp. 197-200,1999.

14. Dereje Teferi, “Optical Character Recognition of Typewritten Amharic Text”, (MSc. Thesis). Addis Ababa, School of information Studies for Africa, Addis Ababa University, 1999.

15. Desalegn Abebe, “Educational Problems of Blind Students: The case of Minilik II and Nigustekelehaimanot Regular Primary Schools” (MA. Thesis). Addis Ababa University, Department of Psychology, Addis Ababa, 2007.

16. Ehquierdo and M.Ghanbari, “Nonlinear Gaussian filtering approach for object segmentation”, Proc.-vis. Image Signal Process, 146(3), 1999.

17. Encarta Encyclopedia 2001; “Optical Character Recognition”; Microsoft Encarta Encyclopedia, Microsoft Corporation, 2001.

18. English Braille: American edition 1994. Available at: http://www.brl.org/ebae/, accessed on October 10, 2009.

19. Ermias Ababe, “Recognition of Formatted Amharic Text Using Optical Character Recogniton”, (MSc. Thesis). Addis Ababa, School of Information Studies for Africa, Addis Ababa University, 1998.

20. Ersboll B.K. and Pedersen K.S., Image Analysis: 15th Scandinavian Conference, SCIA 2007, Aalborg, Denmark, June 10-24, 2007, Proceedings (Lecture Notes in Computer Science / Image ... Vision, Pattern Recognition, and Graphics) (Paperback).

21. Gonzalez, R.C, and Woods, R.E, Digital Image processing, 2nd ed., Prentice Hall, Upper Saddle River, NJ.

22. Greaney J. and Reason R., “Phonological Processing in Braille”, Dyslexia, Volume 5, 215-226, 1999.

23. H. Bunke and P.S.P. Wang, Handbook of character recognition and document image analysis, 1997.

24. H. E. Burdick, “Digital Imaging: Theory and Applications”, McGraw-Hill, Inc. New York, NY, USA 1997

25. H. Hwang and R. Haddad, “Adaptive median filters: new algorithms and results”, Image Processing, 4(4):499-502, 1995.
26. Hu Y. and Ji H., “Research on Image Median Filtering Algorithm and Its FPGA Implementation”, Global Congress on Intelligent Systems, 2009.
27. Iain Murray and Andrew Pasquale, “A Portable Device for the Translation of Braille to Literary Text”, 2006. Available at: http://www.cucat.org/general_accessibility/accessibility/Publications/Murray-Pasquale-Braille%20Translator%20Asset06.pdf.
28. Juneja M., and Mohana R., “An Improved Adaptive Median Filtering Method for Impulse Noise Detection”, International Journal of Recent Trends in Engineering, 1(1), 2009.
29. K.T. Gribbon and D.G. Bailey, “A Novel Approach to Real-time Bilinear Interpolation”, Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications, 2004.
30. Mahmoud Saeidi, M. Hasan Moradi and Fatemeh Sagafi, “Filtering Image Sequences Corrupted by Mixed Noise using a New Fuzzy Algorithm”, Iran Telecommunication Research Center, Amirkabir University of Technology, Tehran, Iran, 2006.
31. McCulloch,W.S. and Pitts,W.H., “ A Logical Calculus of the Ideas Immanent in Nervous Activity,” Bulletin of Mathematical Biophysics, Vol.5, PP:115-133, 1943.
32. Mchugh S. T., “Cambridge in Colour: Digital Image Interpolation”, Available at: http://www.cambridgeincolour.com/tutorials/image-interpolation.htm Accessed on 20 September 2009
33. Mehta. A.j. et.al., “A Multi-Layer Artificial Neural Network Architecture Design for Load Forecasting in Power Systems”, International Journal of Applied Mathematics & Computer Science 4(4), 2008.
34. Mennens, J., et al., “Optical Recognition of Braille Writing Using Standard Equipment”, IEEE transactions of rehabilitation engineering, 2(4): 207-212, 1994.
35. Million Meshesha, "A Generalized Approach to character Recognition of Amharic Texts", (MSc. Thesis). Addis Ababa Univerity, School of information Studies for Africa, Addis Ababa, 2000.

36. Miyoshi T, Nagasaki T, and Shinjo H., "Character Normalization Methods using Moments of Gradient Features" IEICE Tech. Rep., 108(432):187-192, 2009.
37. M.NambaandZ.Zhang, "Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition", International Joint Conference on Neural Networks, 2006.
38. Muhammad Abuzar Fahiem, "A detreminstic Turing Machine for Context Sensitive Translation of Braille Code to Urdu Text", IWCIA, 2008.
39. Nawaz T. et al., "Performance Evaluation of Noise Removal Algorithms for Scanned Images", International Journal of Computer Science and Security, (IJCSS) Vol. 3, 2009.
40. Néstor Falcón et. al., "Image Processing Techniques for Braille Writing Recognition", EUROCAST, 2005.
41. N. Otsu, "A threshold selection method from grey level histogram", IEEE Trans. Syst. Man Cybern, Vol. 9, 1979.
42. Omar Khan Durrani K C Shet, "A New Architecture for Brailee Transcription from Optically Recognized Indian Languages", 3rd International CALIBER, 2005.
43. Philip B. and Sudhaker S.R.D., "Human Machine Interface- A Smart OCR for Visually Challenged", International Journal of Applied mathematics and Computer Science 2(3):34-36, 2009.
44. R. Garnett, et.al., "A Universal Noise Removal Algorithm with an Impulse Detector", IEEE Transactions on Image Processing, 14(11):1747 – 1754, 2005.
45. Ritchings R.T et. al., "Analysis of scanned Braille document", Document Analysis system, A. Dengel and A.L. Spitz (eds.), World scientific Publishing co, 1995.
46. Robert Englebretson, "An overview of IPA Braille: An updated tactile representation of the International Phonetic Alphabet", Journal of the International Phonetic Association, United Kingdom, 2009.
47. Sarfraz, M., Zidouri, A., and Shahab, S.A., A Novel Approach for Skew Estimation of Document Images in OCR System, Computer Graphics, Imaging and Visualization – New Trends, Sarfraz, M., Wang, Y., and Banissi, E. (Eds.), ISBN: 3-7695-2392-7, IEEE Computer Society, USA, 175-180, 2005.

48. Shamsher et.al., “OCR for Printed Urdu Script Using Feed Forward Neural Network”, Proceedings of World Academy of Science, Engineering and Technology. Vol. 23, 2007.
49. S.-W. Lee, “Off-line recognition of totally unconstrainted handwritten numerals”, IEEE Trans. Pattern Anal. Machine Intell, Vol. 18, 1996.
50. Taubin G., “Curve and surface smoothing without shrink age”, IEEE Proceedings of the Fifth International Conference on Computer Vision,1995.
51. Teshome Alemu, “Recognition of Amharic Braille Recognition”, (MSc. Thesis). Addis Ababa University, Department of information science, Addis Ababa, 2009.
52. Tinku Acharya and Ajoy K. Ray, “Image Processing Principles and Applications”, Jhon Wiley, 2005.
53. Trupti Patil, “Evaluation of Multi-core Architectures for Image processing Algorithms”, Master Thesis the Graduate School of Clemson University, 2009.
54. Ullendorff, E., “The Ethiopians: An introduction to the Country and People”, 3rd ed., Oxford University press, London, 1973.
55. UNESCO, “World Braille usage: National Library Service for the Blind and Physically Handicapped Library of Congress”, Washington D.C., USA, 1990.
56. Vasicek Z. and Sekanina L., “Novel hardware implementation of Adaptive Median Filters”, in Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop. pp. 110–115, IEEE Computer Society, 2008.
57. Vidya Shankar G, Hemantha K, and Shivakumara P., “Rotation Invariant Histogram Feature for Recognition of Braille Symbols”, University of Mystore, Karnataka, India, 2004.
58. White paper, “Image Preprocessing for OCR and Data Extraction Environments: OCR Pre-Processing”, Available at: http://www.neurogy.com/ocrpreproc.html Retrieved on September 27,2009.
59. Wondwossen Mulugeta, “OCR for Special type of Handwritten Amharic text (“YEKUM TSIFET”),” (MSc. Thesis). Addis Ababa University, Department of information science, Addis Ababa, 2004.

60. Worku Kelemework, “Automatic Amharic Text News Classification: A Neural Network Approach”, (MSc. Thesis). Addis Ababa University, Department of information science, Addis Ababa, 2009.
61. Yonas, A. et al., x¥R¾: l÷l@J dr© ytzUj. College of Social Science: Addis Ababa University. 1996 E.C.
62. bx!T×ùÃ xYnSW‰N B/@‰ê¦ ¥HbR 12¾ z#R S‰ xSfÉ¸ ÷¸t¦ ytÌÌmW yx¥R¾ BÊL xššY ÷¸t¦ÝÝ yx¥R¾ BÊL xššY ÷¸t¦ zgÆÝÝ ¬HúS 19 xÂ 2; qN 1995 ›.M. lt-‰W xgR xqF g#Æx@ yqrbÝÝ 1995
63. bL† TMHRT b#DN |¼T¼Z¼_ÂTÂ MRMR x!NSt¦T×TÝÝ yBÊL ¥St¥¶Ã ymMH„ mm¶ÃÝÝ xÄ!S xbÆ 1991ÝÝ
64. nByL;#L ×¦NS. zmn BR¦N xÄ!S xbÆ BR¦NÂ s§M ¥t¸Ã b@T 1954ÝÝ
65. tcm wÄËÝÝ yBÊL XDgT bx!T×ÉÃÝÝ xÄ!S xbÆÝÝ 1993ÝÝ
66. g@¬nH xbbÝÝ yx¥r¾ BÊL knhc? 24-28/1991 ucu¬ XW‰N T¼b@T xWd _ÂT yqrbÝÝ bxid ?É¼L×Â m¼t¼|¼T¼Z¼êÂ KFLÝÝ |¼TM¼Z¼K¼M¼x!NSt¦T×TÝÝ 1984
67. g@¬nH xbbÝÝ yx¥r¾ BÊL NÆBÂ {?fT m¥¶Ã ymMH„ mm¶ÃÝÝ bxid ?É¼L×Â m¼t¼|¼T¼Z¼êÂ KFLÝÝ |¼TM¼Z¼K¼M¼x!NSt¦T×TÝÝ 1984

Appendix

I. The first version of Amharic Braille

Table I.1 *List of vowels for first version Braille*

Vowels	1:4	2:5	3:6	1:5	2:4	2:6	2:6
Variant	1st	2nd	3rd	4th	5th	6th	7th

Table I.2 *List of first version Basic Amharic Braille Characters*

ሀ	1:2:4	ሸ	2:3:4
ለ	1:4:5	መ	2:3:4:5
ሐ	1:2:4:5	ዐ	1:6
መ	1:2:5	ዘ	1:4:6
ሠ	2:4:5	ዠ	1:3:4:6
ረ	1:3	የ	1:4:5:6
ሰ	1:2:3	ደ	1:5:6
ሸ	1:2:3:6	ጀ	1:2:4;6
ቀ	1:3:4	ገ	1:2:4:5:6
በ	1:3:4:5	ጠ	1:2:5:6
ተ	1:3:5	ጨ	1:2:3:5:6
ቸ	1:3:5:6	ጰ	2:4;5;6
ኀ	1:2:3:4	ጠ	2:4:5:6
ነ	1:2:3:4:5	ጸ	1:3:6
ኘ	1:2:3:4:5:6	ፈ	1:3:4:5:6
አ	1:2:3:5	ፐ	2:3:4:5:6
ከ	1:2:6		

Table I.3 *List of first Version Extended Amharic Braille Characters*

ቈ	1:2:3:4:6 2:3:4
ኰ	1:2:3:4:6 1:3:4
ጎ	1:2:3:4:6 1:2:3:4
ጐ	1:2:3:4:6 1:2:4:5:6

II. The second version Amharic Braille (1945)

Table II.1 *List of second version Basic Amharic Braille character*

	1st		7th		1st		7th
ሀ	1:2:5	ሁ	1:3:4:6	ኸ	2:3:6	ኽ	5 2:3:6
ለ	4:5:6	ሉ	1:2:3	ወ	2:4:5:6	ዉ	2:4:6
ሐ	Not apply			ዐ	Not apply		
መ	1:3:4	ም	2:3	ዘ	1:3:5:6	ዝ	2:3:4:6
ሠ	2:3:4	ሥ	5	ዠ	3:5:6	ዟ	5:6
ረ	1:2:3:5	ር	1:2:5	የ	1:3:4:5:6	ይ	1:4:5:6
ሰ	Not apply			ደ	1:4:5	ድ	1:4:5:6
ሸ	1:4:6	ሽ	1:5:6	ጀ	2:4:5	ጅ	1:2:6
ቀ	1:2:3:4:5	ቅ	4:6	ገ	1:2:4:5	ግ	2:3:5:6
በ	1:2	ብ	4:5	ጠ	2:3:4:5:6	ጥ	1:2:3:5:6
ተ	2:3:4:5	ት	123456	ጨ	1:4	ጭ	3:6
ቸ	1:6	ች	2:5	ጸ	2:3:5	ጽ	3:4:5:6
ኀ	Not apply			ፀ	1:2:3:4:6	ፅ	3:4:5:6
ነ	1:3:4:5	ን	1:2:4:6	ጰ	Not apply		
ፐ	3:4:6	ፕ	2:6	ፈ	1:2:4	ፍ	5 1:2:4
አ	3	ኦ	3:4	ፐ			
ከ	1:3	ክ	3:5				

Table II.2 *List of vowels for second version Braille*

Vowel	none	1:3:6	2:4	1	1:5	none	1:3:5
Variant	1st	2nd	3rd	4th	5th	6th	7th

III. The third version Amharic Braille (1949E.C)

Table III.1 *List of vowels for third version Braille*

Vowel	2:6	1:3:6	2:4	1	1:5	Independent	1:3:5
Character Variant	1st	2nd	3rd	4th	5th	6th	7th

Table III.2 *List of third version Basic Amharic Braille character*

6th		6th	
ሀ	1:2:5	ኸ	2:3:6
ለ	1:2:3	ወ	2:4:5:6
ሐ	Not apply	ዐ	Not apply
መ	1:3:4	ዘ	1:3:5:6
ሠ	2:3:4	ዠ	3:5:6
ረ	1:2:3:5	ደ	1:3:4:5:6
ሰ	Not apply	ጀ	1:4:5
ሸ	1:4:6	ገ	2:4:5
ቀ	1:2:3:4:5	ጠ	1:2:4:5
በ	1:2	ጨ	2:3:4:5:6
ተ	2:3:4:5	ጰ	1:4
ቸ	1:6	ጸ	2:3:5
ኀ	Not apply	ፀ	1:2:3:4:6
ነ	1:3:4:5	ፈ	Not apply
ኘ	3:4:6	ፍ	1:2:4
አ	*1:2:3:5:6	ፐ	1:2:3:4
ከ	1:3		

*** Symbol indicates Braille code change made on the new version.**

Table III.3 *List of third Version Extended Amharic Braille Character*

Amharic irregular character		
ቈ	1:3	2:4:5:6
ኰ	1:2:3:4:5	2:4:5
ጎ	1:2:5	2:4:5:6
ጐ	1:2:4:5	2:4:5:6

IV. The fourth version Amharic Braille (1995E.C)

Table IV.1 *List of fourth Version Amharic Braille punctuation marks*

Punctu.	Braille code
.	3
:	6 and 3
:-	2:5
/	5 and 2
'	4 and 1
-	3:6
––	3:6 and 3:6
	5 and 2:3:5
'	2
&	2:3
#	2:3:6
$	3:5:6
	6 and 2:3:6
'	3:5:6 and 3
?	2:3:6
::	2:5:6
!	2:3:5
…	3 , 3and 3
( )	2:3:5:6
[	6 and 2:3:5:6
]	2:3:5:6 and 3
*	3:5 and 3:5

__	4:6 and 4:6
→	2:4:6, 2:5 and 2:5
←	2:5,2:5 and 1:3:5
↑	4:5:6 and 1:5
↓	4:5:6 and 3:5
	5:6
✔	4 and 3:4:5
¨YU	3:4
	3:4:5 and 3:4:5
	6 and 3
	2:6 and 3:5
	3:6
	4
	5:6
	5
	2:5 and 2:5
=//=	6 and 3
	4:5:6 and 4:6
	4:5:6, 4:6 and 46
X	1:3:4:6
$	4:5
	3:6, and 3:6

V. MATLAB code for preprocessing

%% Noise detection and Removal, and Binarization

```
%(1)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Read Image
clear, close all
clc
I=imread('C:\\MATLAB7\\work\\test\\Br11.bmp');
%(2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Convert to gray scale
info=imfinfo('C:\\MATLAB7\\work\\test\\Br32.bmp');
ctype=[info.ColorType];
if strcmp(ctype,'truecolor')
   img1=rgb2gray(I);
else
   img1=I;
end;
%(3)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Noise detection and removal
img2 = adapthisteq(img1); %contrast-limited adaptive histogram equalization
h=fspecial('gaussian',[3,3],0.5);
gauf=imfilter(img2,h,'replicate');
amf=wiener2(img2);
%(4)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% Binarization

level=graythresh(amf);

levelg=graythresh(gauf);

level1=level-0.065;

levelg1=levelg-0.065;

modifiedg=im2bw(gauf, levelg1);

modified=im2bw(amf, level1);

modified(1:end,1:4)=1; modified(1:4,1:end)=1; modified(end-4:end,1:end)=1; modified(1:end,end-4:end)=1;

modifiedg(1:end,1:4)=1; modifiedg(1:4,1:end)=1; modifiedg(end-4:end,1:end)=1; modifiedg(1:end,end-4:end)=1;

%(5)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Morphological Operations

bwgau=bwmorph(~modifiedg,'majority');

%bwamf=bwmorph(~modified,'majority');

imshow(~bwgau),title('origonal image');

figure,imshow(modified),title('binarized imageamf');

morphimg=~bwgau;

%(6)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Write the result to file

imwrite(morphimg,'c:\\Braille-DBase\\abay.bmp');

imwrite(modified,'c:\\Braille-DBase\\abay1.bmp');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

VI. MATLAB Code for Recognition

```
%neural network implementation to create model for 12 bit Braille characters
input=xlsread('input_267.xls');
input=input';
input_range=[0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1];
AmharicBrailleNetwork=newff(input_range,[12 1],{'tansig','purelin'},'trainlm');
Time_Start=clock;
AmharicBrailleNetwork.performFcn='mse';
AmharicBrailleNetwork.trainparam.lr=0.01;
AmharicBrailleNetwork.trainparam.goal=0.0001;
AmharicBrailleNetwork.trainparam.show=50;
AmharicBrailleNetwork.trainparam.epochs=1500;
AmharicBrailleNetwork.trainparam.mc=0.8;
AmharicBrailleNetwork.trainParam.mu_max = 1e10;
AmharicBrailleNetwork.trainparam.min_grad=1.0000e-010;
TrainSet=[input];
target=xlsread('output_267.xls');
target=target';
TargetSet=[target];
test=xlsread('features32.xls');
test=test';
testset=[test];
[AmharicBrailleNetwork,tr]=train(AmharicBrailleNetwork,TrainSet,TargetSet);
Rst=sim(AmharicBrailleNetwork,TrainSet);
%convert the output normalized value to index value
Rst=Rst*266+1;
Rst=round(Rst);
 x=xlswrite('result.xls',Rst');
      time=etime(clock,Time_Start);
         disp(time);
```

Declaration

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

_____

Ebrahim Chekol Jibril

June, 2010

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

_____

Dr. Million Meshesha

June, 2010