

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

RECOGNITION OF AMHARIC BRAILLE

**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**

BY

TESHOME ALEMU

MARCH 2009

Dedicated to my family & SubRush!

ACKNOWLEDGEMENT

Beyond and above all I would like to take this chance to praise God for everything that strengthens me even in the dying second to make a restless effort. Next, my sincere thank goes to my advisor Dr. Million Meshesha for his unreserved and constructive comment and ideas to encourage my work.

I would take this chance also to thank German school, particularly Ato Hailue E. and Ato Teshome M. who have been sharing valuable material resources and for their regular collaboration to exchange ideas.

My appreciation goes to my family, Dangnachew A., Wendweson S. Jack A. and Abey(China) A. for their continuous encouragement and support.

Dear friend, Hussien Y., I would like to appreciate all your efforts and encouragement, to reach to the end.

I would like to thank Department of ICT and BUED staffs, who have been helping me in idea and material.

I would like to thank, all my class mate, for their help.

Teshome Alemu

With Blessing!

Table of Contents

List of Figures	x
List of Tables	xii
List of Equations	xiii
Acronyms	xiv
Abstract	xv
Chapter One	1
1. Introduction.....	1
1.1 Background	1
1.2 Braille Writing System	1
1.3 Braille Recognition	4
1.4 Statement of the Problem	6
1.5 Objectives	8
1.5.1 <i>General objective</i>	8
1.5.2 <i>Specific Objectives</i>	8
1.6 Justification of the study	9
1.7 Methods.....	10
1.7.1 <i>Literature Review</i>	10
1.7.2 <i>Implementation Tools</i>	11
1.7.3 <i>Testing procedure</i>	11
1.8 Scope and Limitation of the Study.....	11
1.9 Application of result	12
1.10 Organizations of the Study	12
Chapter Two.....	14
2. Review Literature	14
2.1 Introduction	14
2.2 Amharic Writing System.....	15

2.2.1	<i>The Amharic Characters (Fidel)</i>	15
2.3	Evolution of the Amharic Braille	17
2.4	The Amharic Braille Code character	19
2.4.1	<i>The Fourth version Amharic Braille</i>	19
2.5	Amharic Braille Writing and Reading system	22
2.5.1	<i>Braille writing</i>	22
2.5.2	<i>Braille reading</i>	23
2.6	Braille computer representation	23
2.7	Representation of Amharic Braille character	24
2.7.1	<i>Braille character</i>	24
2.7.2	<i>Interpretation of Amharic Braille character</i>	24
2.8	Features of Amharic Braille Code	26
2.9	Nature of Amharic Braille embossed	27
2.10	The Braille OCR system	28
2.10.1	<i>Characteristics of Automatic Braille Reading system</i>	28
2.10.2	<i>Optical Braille recognition system Overview</i>	29
2.10.2.1	Image Acquisition/Digitization	29
2.10.2.2	Image Pre-Processing	30
2.10.2.3	Image Segmentation	30
2.10.2.4	Feature Extraction and Recognition	31
2.11	Review of Braille optical recognition work	32
2.11.1	<i>Optical Recognition of Braille Writing Using Standard Equipment</i>	34
2.11.1.1	Image preprocessing	34
2.11.1.2	Segmentation	35
2.11.1.3	Feature Extraction and Recognition	35
2.11.1.4	Test Results and Summary	36
2.11.2	<i>Analysis of scanned Braille documents</i>	37

2.11.2.1	Image Acquisition.....	37
2.11.2.2	Identification of Braille Dots	38
2.11.2.3	Character Segmentation	38
2.11.2.4	Character Recognition.....	39
2.11.2.5	Results and Discussion	39
2.11.3	<i>A Braille OCR for Blind people</i>	40
2.11.3.1	Point Localization	40
2.11.3.2	Mesh detection	41
2.11.3.3	Translation into ASCII.....	42
2.11.4	<i>Regular Feature Extraction for Recognition of Braille</i>	42
2.11.4.1	Scene Constraints.....	43
2.11.4.2	Image Acquisition.....	43
2.11.4.3	Pre-processing.....	43
2.11.4.4	Segmentation	44
2.11.4.5	Feature extraction.....	44
2.11.4.6	Interpretation.....	44
2.11.5	<i>A Software Algorithm Prototype for Optical Recognition of Embossed Braille</i>	45
2.11.5.1	Half-Character Detection	46
2.11.5.2	Half-Character Recognition.....	47
2.11.5.3	Text File Transcript.....	47
2.11.6	<i>Image Processing Techniques for Braille Writing Recognition</i>	47
2.11.6.1	Database	48
2.11.6.2	Thresholding	48
2.11.6.3	Pattern detection block	49
2.11.6.4	Braille grid.	49

2.11.6.5	Dots recovery using Braille grid	50
2.11.6.6	From Braille to Normal Text.....	50
2.11.7	<i>An Arabic Optical Braille Recognition System</i>	51
2.11.7.1	Image acquisition/Converting the Image to Gray Level.....	51
2.11.7.2	Image pre-processing.....	51
2.11.7.3	Segmentation	53
2.11.7.4	Braille Cells Recognition.....	54
Chapter Three	57
3.	Braille Recognition Technique	57
3.1	Introduction	57
3.2	Digitization/Image Acquisition	57
3.3	Binarization/Image Thresholding	57
3.4	Image Segmentation.....	59
3.5	Feature Extraction and Recognition	63
3.6	Artificial neural network.....	64
3.6.1	<i>Introduction</i>	64
3.6.2	<i>Basic structure of a neuron</i>	65
3.6.2.1	The Human brain.....	65
3.6.3	<i>Component of Neural Network</i>	67
3.6.3.1.1	Weighting Factors	68
3.6.3.1.2	Summation function	68
3.6.3.1.3	Transfer Function	68
3.6.3.1.4	Scaling and Limiting	69
3.6.3.1.5	Output Function.....	69
3.6.3.1.6	Error function and Back-propagated value.....	69
3.6.3.2	Learning functions.....	70
3.6.3.3	Type of Activation Function	70

3.6.3.3.1	The activation function	70
3.6.3.4	The network Architecture	71
3.6.3.4.1	Single layer feed forward networks	72
3.6.3.4.2	Multi layer feed forward networks	72
3.6.3.4.3	Recurrent networks	73
3.6.3.5	The learning process	74
3.6.3.5.1	Supervised learning	75
3.6.3.5.2	Unsupervised learning	75
3.6.3.6	Learning laws.....	76
3.6.3.6.1	Hebb's Rule	76
3.6.3.6.2	Hopfield Law	76
3.6.3.6.3	The Delta Rule	76
3.6.3.6.4	The Gradient Descent Rule	77
3.6.3.7	Multilayer Feed forward neural networks	77
3.6.3.8	Back-propagation	77
3.6.3.9	Application of neural networks.....	79
Chapter Four	80
4.	Experimentation.....	80
4.1	Introduction	80
4.2	Data Collection	82
4.3	Digitization/Image Acquisition	83
4.4	Binarization/Image Thresholding	84
4.4.1	<i>Binarizing Gray Image</i>	87
4.4.2	<i>Binarizing Color image</i>	88
4.5	Image Segmentation.....	91
4.6	Amharic Braille Feature Extraction and Recognition.....	101
4.7	Amharic Braille-print character recognition using ANN	106

4.7.1	<i>Artificial Neural Network Architecture</i>	106
4.7.2	<i>The Neural Network</i>	108
4.7.3	<i>Braille Character Input Method for the Neural Network</i>	108
4.7.4	<i>Creating the Neural Network</i>	109
4.7.5	<i>Training the Neural Network</i>	110
4.7.6	<i>Testing the Neural Network</i>	112
4.7.6.1	Performance Evaluation with one and two cell test dataset.....	114
Chapter Five		116
5.	Conclusions and Recommendation.....	116
5.1	Introduction	116
5.2	Conclusion	116
5.3	Recommendation	118
Reference		121
Appendix		126
I.	The first version of Amharic Braille.....	126
II.	The second version Amharic Braille (1945)	127
III.	The third version Amharic Braille (1949E.C).....	128
IV.	The fourth version Amharic Braille (1949E.C)	129
V.	Feature extraction Algorithm	129
VI.	Matlab code for Recognition.....	132
VII.	Sample Braille Image	133
Declaration		134

List of Figures

<i>Figure 1.1 Dot positions in a Braille cell</i>	2
<i>Figure 2.1 Amharic Core Characters</i>	16
<i>Figure 2.1 sample representation of Braille code</i>	20
<i>Figure 2.2 Braille dot dimension specified by library of Congress</i>	24
<i>Figure 2.3 Correlation mask for an image with a distance of five pixels between the center of the dark and light zone[left]and Neighboring dots introduce false core regions or ghost-point[right].</i>	35
<i>Figure 2.4 Analysis of scanned Braille documents.</i>	38
<i>Figure 2.5 Braille image showing left raised dots (protrusions) and right valley (depressions).</i>	40
<i>Figure 2.6 A Software Algorithm Prototype for Optical Recognition of Embossed Braille.</i>	46
<i>Figure 2.7 Image Processing Techniques for Braille Writing Recognition</i>	48
<i>Figure 2.8 Arabic Braille Recognition proceeding</i>	53
<i>Figure 2.9 Arabic Braille document recognition system</i>	54
<i>Figure 3.1 Global-thresholding algorithm for Braille image</i>	58
<i>Figure 3.2 Horizontal Mesh-grids algorithm</i>	61
<i>Figure 3.3 Vertical Mesh-grids algorithm</i>	62
<i>Figure 3.4 The Biological Neuron Model</i>	66
<i>Figure 3.5 Artificial Neural Network Model</i>	67
<i>Figure 3.6 Feed Forward Neural Network</i>	73
<i>Figure 3.7 Simple Networks with Feedback and Competition.</i>	74
<i>Figure 3.8 Feed-forward algorithms for neural network.</i>	79
<i>Figure 4.1 Block diagram of the Amharic Braille Recognizer.</i>	82
<i>Figure 4.2 Braille document image (left with gray level scale and right with color)</i>	84
<i>Figure 4.3 Frequency distributions of colors for Braille image (left- Gray-level image, and right- color Braille image)</i>	86
<i>Figure 4.4 Gray-level image after threshold (color value ≤ 202 or color value ≥ 247)</i>	87

<i>Figure 4.5 Binarized gray-level image (left- Color value less than 202, and right- with Color value greater than 247)</i>	87
<i>Figure 4.6 Result of Binarization with two threshold value.</i>	88
<i>Figure 4.7 Color-level images after threshold</i>	88
<i>Figure 4.8 Ms-visual C++ code to extract the RGB of image pixel and convert to gray-level</i>	89
<i>Figure 4.9 Noisy image after thresholding ((a) repeatedly used Braille image, (b) uniform color and (c) light effect).</i>	90
<i>Figure 4.10 Result of Binarized image after reducing the threshold (Image with reduced threshold)</i>	90
<i>Figure 4.11 Sample binarized which is used for further recognition.</i>	91
<i>Figure 4.12 Braille cell with dots position</i>	91
<i>Figure 4.13 Horizontal and vertical projection for Braille image.</i>	92
<i>Figure 4.14 Inter and intra dot spacing in Braille</i>	93
<i>Figure 4.15 Vertical grids line based on threshold value (10 pixels for dot within a cell and 14 pixels for dots between cells).</i>	94
<i>Figure 4.16 Horizontal grids line based on threshold value (10 pixels for dot within a cell and 18 pixels for dots between cells).</i>	94
<i>Figure 4.17 Braille image with Global grids</i>	95
<i>Figure 4.18 Deviation of grids line towards the right ends</i>	95
<i>Figure 4.19 A scenario that shows grids for overlapping dots considering the threshold dot width</i>	96
<i>Figure 4.20 Horizontal mesh implementation</i>	97
<i>Figure 4.21 Horizontal gridline for Braille image</i>	98
<i>Figure 4.23 Mesh grids line with ideal vertical line.</i>	99
<i>Figure 4.22 Visual C++ code to draw hypothetical vertical grids line</i>	98
<i>Figure 4.24 Implementation of mesh based segmentation</i>	99
<i>Figure 4.25 Segmented Braille image</i>	100
<i>Figure 4.26 Amharic Braille-print recognition network architecture</i>	107
<i>Figure V.1 Feature extraction algorithm based on grids-mesh and contextual approach</i>	131
<i>Figure VII.1 Sample Braille image with Gray-color</i>	133

List of Tables

<i>Table 2.1 The total number of Amharic characters by type</i>	17
<i>Table 2.1Forth version Amharic Braille character.....</i>	20
<i>Table 2.2 Fourth version Amharic Braille vowels</i>	21
<i>Table 2.3 Amharic Braille punctuation of the last version.....</i>	21
<i>Table 2.4 Ethiopic, Arabic-Braille characters</i>	22
<i>Table 2.5 Summary of Braille optical recognition research work</i>	33
<i>Table 4.1 Summary of Amharic Braille documents collected for the study</i>	83
<i>Table 4.2 Frequency distribution of colors (left for gray-level image and right for color image)</i>	84
<i>Table 4.3 Basic Amharic Braille character with one cell</i>	102
<i>Table 4.4 First variant of basic Amharic Braille character with tow cells.</i>	102
<i>Table 4.5 Arabic Braille numerals with two cells.</i>	102
<i>Table 4.6 Amharic Braille numerals with two cells.</i>	103
<i>Table 4.7 Lists of punctuation marks with one and two cell.</i>	103
<i>Table 4.8 punctuation marks that defined in three Braille cell.</i>	104
<i>Table 4.9 Two cell Braille-print character representation.....</i>	105
<i>Table 4.10 Sample Amharic characters with same ASCII value</i>	108
<i>Table 4.11 Training parameters for the proposed network</i>	111
<i>Table 4.12 Error type and list in validating the network.....</i>	113
<i>Table 4.13 Performance rates for test dataset</i>	114
<i>Table I.1 List of vowels for first version Braille</i>	126
<i>Table I.2 List of vowels for first version Braille</i>	126
<i>Table I.3 List of first Version Extended Amharic Braille Character</i>	126
<i>Table II.1 List of second version Basic Amharic Braille character</i>	127
<i>Table II.2 List of vowels for second version Braille</i>	127
<i>Table III.1 List of vowels for third version Braille</i>	128

<i>Table III.2 List of third version Basic Amharic Braille character</i>	128
<i>Table III.3 List of third Version Extended Amharic Braille Character</i>	128
<i>Table IV.1 List of fourth Version Amharic Braille punctuation marks</i>	129

List of Equations

<i>Equation 3.1 Weight Summation function for Neural Network</i>	67
<i>Equation 3.2 Threshold Function</i>	70
<i>Equation 3.3 Piecewise Linear Function</i>	71
<i>Equation 3.4 The Sigmoid Function</i>	71
<i>Equation 4.1 Min-max normalization formula</i>	109

Acronyms

EABS Ethiopian Association for Blind society

IEF International Eye Foundation

OBR optical Braille recognition

OCR optical character recognition

Abstract

According to the International Eye Foundation (IEF) reports, there are currently about 45 million visually impaired people in the world, the vast majority of which has been living in Africa. In Ethiopia, the latest census indicates that there are well over half a million visually impaired individuals including: students, lawyers, teachers, researchers, artists etc. So far, Braille has been the most invaluable means for visually impaired individuals to communicate to the world. Braille called after its inventor Braille Luis is a tactile writing means that consists of six dots arrangement in 2-by-3 matrix in a cell. Societal support is important to rebuild lives devastated by sight loss. These days technology has contributed a paramount value to the society in facilitating two way communications. Recently, optical character recognition (OCR) technique has been implemented for the recognition of Braille documents scanned with standard scanning device.

In this study, an attempt has been made in Amharic Braille-to-print documents recognition. To achieve this, various techniques has been reviewed, developed and adopted. The proposed system performs the required recognition in two phases: these are recognition of Braille character and Braille-to-print character. The first phase involves four steps such as thresholding/Binarization, image-segmentation, and feature extraction and recognition. Global-threshold has been implemented to binarize the foreground (content) from the background. As Braille cell are strictly arranged horizontally and vertically, mesh-grid technique has been adopted for segmentation process. With mesh, dots are extracted following the vertical and horizontal grid line. Having done this, in feature extraction the system once again takes advantage of the mesh. So, during this steps dots are further grouped in to cell, which would then recognized with context analysis based on rules defined. To accomplish the first phase Microsoft Visual C++ programming tools has been used.

The second phase, deals with classification of Braille-to-print. To this end MATLAB's implementation of the feed forward artificial neural network has been utilized.

The neural network classifier has been trained on Amharic Braille with Amharic print as the target character. Moreover, the performance of the model has been evaluated with test sets that are prepared from the Braille document.

Eventually, the study has shown better performance with all training and test set, with 92.5% accurate.

Chapter One

1. Introduction

1.1 Background

“Every day 100 people will start to lose their sight offering practical support and information to anyone with a sight problem. Your support helps us rebuild lives devastated by sight loss.”[2].

According to the International Eye Foundation (IEF) [5] reports, there are currently about 45 million blind people in the world, the vast majority of which has been living in Africa. In Ethiopia, the latest census indicates that there are well over 500,000 blind people in the country.

It is undeniable fact that visually impaired people are part and parcel of the society and play significant role in the society [6] where they live in. So far there are different touch and/or hear means and system created for visually impaired people as a means to reach what the world has in printed document [6]. One of the most valuable and indispensable system is through the use of Braille.

1.2 Braille Writing System

Braille is a code which enables visually impaired people to read and write. It was invented by a visually impaired Frenchman, Louis Braille, in 1821 [33][39] [47]. Braille is comprised of a rectangular cell and each cell is made to consist up to six-dots, that makeup up to 63 possible combinations of different characters set or sequences of characters using 1 to 6 dots [6][29][41]. The different arrangement of raised dots in a 3-by-2 Braille is presented in *Figure 1.1*.

Currently there are two types of Braille styles that are commonly used in the world from the English character set, i.e., American Braille and France Braille style [33] [47] . However, most countries adopt

and/or define their Braille code so as to fit their local language character and have their own set of defined dot pattern. Not standing with the above fact, in general Braille codes that are used worldwide currently are categorized in to two levels [13][21][47]. The first, *Grade 1 Braille* represents each *print character* as one Braille cell, but many experienced Braille users read and write grade 2 Braille [5][6][13], which is a form of shorthand where group of letters may be combined into a single Braille cell. The Grade 2 Braille, which is also known as contractions, is embossed (process of writing Braille code) by hand (manual) and/or with a machine (Braille embosser) onto a thick paper, and is read with fingers moving across on top of the dots [43]. A standard Braille page size is 11 inches by 11.5 inches and typically has a maximum of 40 to 43 Braille cells per line which comprise a total of 25 lines per Braille page [6].

“...All kinds of material can be put into Braille, from bank statements to bus timetables, maps to music and mathematical notation.” [6]

So Braille has been the major reliable method of literacy for blind people and a means to communicate with the world [1][43].

The basic Braille symbol or cell that is composed of 6 dots arranged in 2 vertical columns, each column being 3 dots high are numbered as shown in *Figure 1.1*: [6][43][20]

1 0 0 4
2 0 0 5
3 0 0 6

Figure 1.1 Dot positions in a Braille cell

The Braille sheet is a thick material designed to withstand the pressure while embossing the dots. While writing the character, it can be embossed in one side or both side of the sheet [21].

In Ethiopia the introduction of Braille has more than half a century history. According to [47], modern touch education for visually impaired people started in Ethiopia by the end of August 1924 by Ruthle who

was one of the missionary who came to Ethiopia for the same purpose.

Nebyeluel [47] further describe in his book that the American press missionary could bring Braille education in Ethiopia to keep Mr. Matiwos' promises. Matiew, who was one of the rich individual in America by the time, promised to donate all his wealth for the establishment of Braille education in Ethiopia as he reached to death. His motive as described by Nebyelul was [47]:

"...Visually impaired individual in my country (American) help themselves in any aspect and ... have access to learn different handcraft and education, so this chance should reach to blind people who lived in Ethiopia..."

Accordingly, based on Mathews promises, the missionary [47] established Braille education in Dembidolo (the former welega awraja) with the name of the donor "Kedus Mathewos Yayne Seweran Temihrt Bet" and commence teaching handcraft and the Amharic writing system "U" to "T" which was created on wooden board with nails. This was because there was no readily made Braille code so far developed for Amharic character [47].

Mr. Giddada, the former president Dr. Negasso's father, was among the pioneer learner who found to study both Amharic and English Braille within a short period of time and got respect from teachers and people by the time [47]. Soon with the missionary's good will to help the visually impaired people to access education and learn handcraft, they established the second Braille education in *Debre Markos* in 1933. However they would not get acceptance from the society. As a result they soon came to Addis Ababa to establish the third Braille education which was located around Ras Mekonnen bridge, in 1934 [47].

This continued and in 1934 "Merha eweran" visually impaired education center was established in Addis Ababa, by the good will of the majesty, Haile Silasse [47]. Later in 1942 the Sweden missionary, which was found around Entoto established another Braille school to provide Braille education [47]. This way

the expansion of Braille education increased from time to time in Ethiopia.

According to Nebyluel [47], it was in 1934 that Amharic Braille code was introduced in Ethiopia. The Amharic Braille had subjected for improvement in 1954, 1956 and 1958 before it takes the current Braille dot pattern [20][47]. This Amharic Braille was designed by taking the 26 English characters pattern and with 8 newly added patterns. Thus, it exclude Amharic “fidel” that have different shape but same sound (as they are redundant) considering one pattern representation for same sound character.

Thus, for the last 40 years 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 and generation gaps between the vision and visually impaired people. The condition even would be very serious as the world trend is towards information society where information is a vital resource for advancement.

This research would have significant contribution in facilitating smooth communication between visually impaired and vision society in every dimension of life improving basically the life of visually impaired society.

1.3 Braille Recognition

One of the responsibilities of information scientist is making information item in any format accessible and usable by the society. Over the period, there are many research conducted in order to exploit the power of information technology (hardware and software) to solve societal problem such as information accessibility and computability which has been particularly increasingly sophisticated in the information era because of the immense increase in printed publications. Accordingly, Optical Character Recognition (OCR) technology is becoming a field of research in pattern recognition, artificial intelligence and machine vision. OCR system involves reading scanned bitmap images of machine-printed or hand

written text and translating the images into a form that the computer can manipulate (for example, into ASCII codes), i.e. OCR examines and translates the characters into ASCII text files that can be edited [13][26].

There are two types of OCR systems with respect to the way the input is provided to the system: on-line OCR where the recognition task is performed parallel with the writing process and off-line OCR where the recognition process is performed after the whole text entered to the OCR engine [40].

Recently, OCR technology has been the focus of many researchers in an attempt to recognize Braille in the process of making it accessible and computable to support visually impaired people [6][41].

According to [14], the Braille recognition system consists of six operations, such as Scene constraints, Image acquisition, Pre-processing, Segmentation, Feature extraction and Interpretation.

The first phase, *scene constraint/initial preparation*, has two principal aims: maximizing the use of prior knowledge of the scene, through exploiting existing knowledge, and to lower (play down) the problem of image analysis in any possible way. This can be achieved by imposition of effective constraints such as illumination [14].

Image acquisition is the process of capturing a Braille page and converts it into a digitized image array using a capturing device such as digital camera and scanner [14][26].

During pre-processing the major operations are noise filtering, edge enhancement and skew correction. This is because during image acquisition, impulse noise may be introduced in the image. Thus, this operation requires contrast enhancement and adjustment, noise removal and edges sharpening, and tilt image correction so as to modify and prepare the pixel values in the digitized image for subsequent operations [14][26].

Segmentation separates the acquired image into foreground (dots) and background regions. This operation also separates the front-faced dots and back-faced dots for double-sided Braille. As the Braille page images are noisy the selection of threshold value is problematic so considering a histogram made up of only those pixels that lie at or near the edges of the Braille dots improve the segmentation process [14].

Feature extraction is a representational mechanism of the Braille image. The main function of *feature extraction* is to extract the Braille dots from the binary image and grouping them into cells. In this case, 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 that detects the boundary coordinates of the Braille dots. So once the coordinates of the dots are detected, the diameter of the dot can also be deduced [14][26].

Interpretation/Classification: is the final operation that converts the Braille cells into the respective text (character). It involves grouping the Braille dots into cells and converts them into the text by determining the centroid between dots and the four possible neighbors. Each word is then checked against 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 [14].

1.4 Statement of the Problem

Since the introduction of Amharic Braille in Ethiopia there has been massive Braille documents that have been produced and found at different part of the country: typically AAU Kennedy library at Braille documentation, 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 (outside Addis Ababa) including the Ethiopian Orthodox Church are worth to mention. Moreover, for a long period of time Braille had been and is an invaluable means for visually impaired individuals [43][47] to document ideas, concepts, knowledge and so on. There are well more than half a million visually impaired individuals [41] that span students to teachers, artists to lawyer, thinker to employee

who have and play significant contribution [47] in political, religious, economic and social affair of the society.

There are many visually impaired teachers from elementary to high schools and colleges to universities who have used their Braille as the only means to codify their knowledge. There are visually impaired students in higher education. There are also many visually impaired artists and writer who had and have a lot to share for the societies they live in. However most of their work still remained in their Braille [47] and of course accessed by chance by those who can know, read and write Braille. There are many visually impaired employees' who have been working in governmental and nongovernmental organizations. There are attorney who work around the court area. *But how many of them are really reached to the vision society? How much effort is expected to convert Braille to printed format?* In fact there might be very few attempts that have done in this area to produce the Braille document in printed format either through oral dictation or by vision individuals who have the skill to read and write both Braille and the printed character. Even this is hardly reached to success as the transcription is very cumbersome. According to [36], until recently, creating book in Braille was time taking process. Which would in some cases could take hundreds of hours.

As Nebyuluel [47] notes, 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 people. And the work of many visually impaired people would have remained buried [47]. This would create a wide generation gap between the visually impaired and sighted society.

Nebyalule [47], also discuss many issues concerning the visually impaired individuals including: societies' attitude towards them, their competence to learn and accomplish different responsibility citing different talented and competent visually impaired individual from abroad and local, what visually impaired people expect from the society. All the issues discussed forward significant message for the sighted society to give respect, understand their capabilities, and share their feeling creating awareness among society for

the wellbeing of the visually impaired and sighted society in general. *In fact there is no one who knows and describe about visually impaired individuals more than them.* So we have to look for means by which the vision society can access their work.

Since 1997, there are a number of research works in the application of OCR technology in one of Ethiopian major language, Amharic. These include the work of worku [42], Ermias [18], Dereje [16], Berhanu [11], Million [31], and Wondwesson [40]. However, all of these research works were conducted on Amharic printed text documents that are either produced manually or typewritten or computer printout.

Literature disclose that different attempt have been done to develop Braille OCR system that work on some language including English, Arabic, Chinese and Indian to facilitate two way communication and knowledge transfer between visually impaired and vision society. However, none of the OCR Braille system developed abroad supports the Amharic Braille.

To the best of the researcher's knowledge, none of the preceding or any other works attempted on Amharic Braille OCR. Hence this study investigates the possibility of developing Braille recognition system for Amharic Language.

1.5 Objectives

The general and list of specific objectives of this study are presented below.

1.5.1 General objective

The general objective of this research is to develop Amharic Braille OCR that enable to recognize optically scanned Amharic Braille and convert to the equivalent printed Amharic text character.

1.5.2 Specific Objectives

In attempt to accomplish the aforementioned general objective, the study accomplishes the following specific objectives.

- Review previous related work in the area of Braille recognition so as to understand the domain area and assess different algorithms for Braille image preprocessing, segmentation, feature extraction, classification, and character mapping technique.
- Analyze the features and pattern of Amharic Braille code style for visually impaired individual.
- Understand the application of Neural Network approaches for Braille code recognition.
- Design algorithms required for the various steps (preprocessing, segmentation, feature extraction and classification) involved towards Amharic Braille recognition.
- Test and evaluate the performance of the algorithms developed by collecting sample Amharic Braille documents.
- Make conclusions and forward recommendations based on the test results.

1.6 Justification of the study

Awareness among the society about visually impaired people would be more promising if and only if there is two way communication between the visually impaired and vision society. Experience of others country shows that optical Braille recognition has many benefits to Braille users and those who work with them and/or would need to see their work in facilitating communication. It serves as a bridge between visually impaired and vision society. Besides, the outcome of the research would significantly reduce storage space [6][41] as the size and related volume of Braille documents is relatively large as compared to the print document. OCR is an important technology to convert the information contained in Braille into electronic format which would be more useful in information era [6][40] as it preserve Braille out-of-print. Moreover as the reproduction of Braille is cumbersome this research would lay a foundation for future

work in order to come up with an applicable Braille OCR system.

So far there are many attempts that have been done to support visually impaired individuals to access many of the printed documents. This attempt is successful in other language in different country through computer application. To the researcher knowledge, though there are different project started to develop Amharic Embosser, i.e. to transcribe print text to Braille, there is *no full-fledged* program that has successfully completed and working in Amharic Braille. This research is aimed at investigating the possibility of converting optically recognized Braille in to computer understandable format.

Thus, the objective of this research would have significant contribution in making accessible what visually impaired society had and do have for the sighted society creating awareness. This would be the first attempt that would bridge the gap between the two scripturally departed societies: vision and Blind.

1.7 Methods

Braille recognition involves various steps of image processing. Research makes available various techniques towards such a problem. Since the main aim of the thesis is to adopt and/or devise algorithms that perform Amharic Braille recognition, various methods of research are reviewed. The following part has been describing research methodologies used.

1.7.1 Literature Review

Related literature on the area of OCR Braille recognition and conversion to print has been thoroughly reviewed from various sources including: articles, books, journals, magazines, proceedings and the Internet. This is done to understand different OCR recognition technology, to map techniques and algorithms so far identified so as to address the related problem domain.

The researcher would believe that this has a paramount significant as the aim of this research is to adapt

and/or develop an algorithm that perform the required Braille recognition for Amharic Braille and convert to the corresponding text representation.

1.7.2 Implementation Tools

For the purpose of developing Braille OCR that converts to the equivalent Amharic text, Microsoft visual C++ programming language is used. This is because of the fact that it minimizes the effort required to structure source code which is time-consuming, and delivers increased performance and productivity by enabling developers to leverage existing skills.

The researcher has also used neural network for the purpose of classification of Braille dot for the very reason that it is designed to address problems that are often complex, such as pattern recognition and has the ability to learn from example.

1.7.3 Testing procedure

Testing is an important step to measure the performance of Amharic Braille recognizer that is designed in the present research work. Test is carried out using patterns extracted from the Braille document. Once the network has been trained for Amharic Braille characters, numerals and punctuation marks, the performance of the network tested with total of 550 Braille character.

1.8 Scope and Limitation of the Study

The aim of this study is assessing the potential application of OCR technologies to recognize Amharic Braille, so as to help improve the information accessibility between vision and visually impaired society. The scope of the research is limited to converting Amharic Braille character to print character for Single-sided Braille, that are embossed on one side of Braille paper

Given an image of Braille, an attempt has been made in this research work to adopt OCR algorithms for Amharic Braille recognition and apply MATLAB neural network tool to build a model that classify Braille dot into print character. The study has made experiment on the applicability of selected preprocessing, segmentation, feature extraction, and classification techniques for the recognition of Amharic Braille.

The study planned to perform on all Amharic Braille character, numerals and punctuation marks. However, because of time constraints the current study could not conducted in extended Amharic Braille characters, Ethiopic numerals and some punctuation marks. Besides, Braille character defined with one, two and three cells. Braille character with three cells is not included in the study for the same reason. Moreover, many sample Braille document, with variety of content could not be found as reproduction of Braille is cumbersome.

1.9 Application of result

The outcome of the study will have significant contribution to bridge communication gap between the two scripturally separated societies: sighted and visually impaired in teaching-learning process, work area, justice, and art. In many cases different organizations particularly refrain to hire visually impaired individuals as they would have no means to communicate *formally* through written document unless there is someone who mediates in between to transcribe what the visually impaired people produced in to printed format. Student at different level: elementary, high school and higher-education will benefit from the outcome. The outcome also benefit lawyer visually impaired individual in their work environment.

Accordingly, the government and the non-government organizations, and the community at large, will benefit from the research, as the output of the research contributes for developing a practical application.

1.10 Organizations of the Study

The paper is organized into five chapters. Chapter one introduces the present research problem, objectives of the study and methodologies adopted. In chapter two an attempt has been made to review literatures in Amharic Braille system, features of Braille character and characteristics of Braille recognition. Review of related research work in Braille problem domain along with the techniques used is also discussed. Chapter three discusses basic components and issues in application of artificial neural network. Chapter four reports experimental results of the research. The experiment basically comprises of two parts: the first part involves mainly the image preprocessing, segmentation and feature extraction, and the second part consists of the recognition phase which involve mainly training; building and validation of the models in addition to analysis and interpretation of the results. Finally, concluding remarks and recommendations for future research are forwarded.

Chapter Two

2. Review Literature

2.1 Introduction

Since its invention by French blind man, Louis Braille, Braille has been adapted to nearly every language on earth and remains the major medium of literacy for blind people everywhere [47][33]. For the last many years, restless efforts have also been made to improve the Braille system in every dimension. Their efforts expanded over the last decade with the coming of the computer age.

Literature revealed that since the introduction of OCR technology there have been different attempts to exploit the potential applicability of the technology to Braille document in different part of the world [21]. Though, there has been a great deal of research work conducted on the application of OCR to Amharic printed document [18][31][40], nothing is so far started to exploit the potential applicability of OCR to Amharic Braille documents. Accordingly the work of different researchers in an attempt to Braille recognition from abroad are reviewed in this study. An attempt was made as early as in 1988, by Dubus et al. [41] to design an algorithm which translates relief Braille into an equivalent print character version on paper. And a recent attempt was also made in New Zealand [6], Spain [32] and Tunisia [41], respectively towards the recognition of Braille documents. In this chapter, techniques and approach to Braille OCR is reviewed to show how it is done and what is to be done there by enlighten the direction of the research underway.

To provide an overview of the problem domain, the discussion of Amharic Braille writing system is also presented before the review of Braille OCR system. In addition, Amharic Braille code representations are discussed. This is important in the subsequent discussion of feature extraction of the Braille dot recognition.

2.2 Amharic Writing System

According to [10][31], the present writing system of Amharic is taken from Geez (evolved out of Sabaean language) which was brought to the highlands by immigrant from south Arabia in the first century A.D . Geez, which was remained the medium of religious and literary expression in Ethiopia until the 16th century gradually gave way to Amharic.

Before the Amharic characters (Fidel) have got their existing shape and number of symbols, they have gone through a serious of changes by adopting and using the Sabaean and Geez scripts over a long period of time [10][18][31].

By the time Geez was replaced by Amharic, Amharic took, for the purpose of writing, all the 26 symbols which were used in the Geez language. In addition, Amharic has created eight additional symbols to represent sounds not found in Geez [10][31]. This increased the total basic symbols used in Amharic writing system to 34; out of which 33 are core character and 1 is a special character.

2.2.1 The Amharic Characters (Fidel)

Currently, Amharic language consists of 33 core characters each of which occurs in seven orders (one basic form and six non-basic forms), representing syllable combinations consisting of a consonant and following vowel [10][47]. This brought the total number of core characters to 231(33x7). List of these basic characters are shown in *Figure 2.1*.

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሐ	ሑ
መ	ሙ	ሚ	ማ	ሚ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሠ	ሡ
ረ	ሩ	ሪ	ራ	ሪ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ሰ	ሱ
ሸ	ሹ	ሺ	ሻ	ሼ	ሸ	ሹ
ቀ	ቁ	ቂ	ቃ	ቄ	ቀ	ቁ
በ	ቡ	ቢ	ባ	ቤ	በ	ቡ
ተ	ቱ	ቲ	ታ	ቴ	ተ	ቱ
ቸ	ቹ	ቺ	ቻ	ቼ	ቸ	ቹ
ኀ	ኁ	ኂ	ኃ	ኄ	ኀ	ኁ
ነ	ኑ	ኒ	ና	ኔ	ነ	ኑ
ኘ	ኙ	ኚ	ኛ	ኜ	ኘ	ኙ
አ	አ	አ	አ	አ	አ	አ
ወ	ወ	ወ	ወ	ወ	ወ	ወ
ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ
ከ	ከ	ከ	ከ	ከ	ከ	ከ
ኸ	ኸ	ኸ	ኸ	ኸ	ኸ	ኸ
ዘ	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ
ዠ	ዠ	ዠ	ዠ	ዠ	ዠ	ዠ
የ	የ	የ	የ	የ	የ	የ
ገ	ገ	ገ	ገ	ገ	ገ	ገ
ደ	ደ	ደ	ደ	ደ	ደ	ደ
ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ
ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ
ጨ	ጨ	ጨ	ጨ	ጨ	ጨ	ጨ
ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ
ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ
ጳ	ጳ	ጳ	ጳ	ጳ	ጳ	ጳ
ፈ	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ
ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ

Figure 2.1 Amharic Core Characters

In addition to the 231 core characters, there are others, which are indicated below [44][31]:

- Special character (ቬ), which has also seven forms, and has used to represent the ‘v’ sound of words in Latin-based languages.
- Labialization character, a total of 44 symbols to represent special features. Such symbols include ሊ ሚ ረ ሳ ቆ ሳ etc.
- Punctuation consisting of word-divider(:), end of sentence indicator(::), comma(፣), semi-colon(፥)

and other Latin-borrowed symbols like question mark(?), exclamation point(!) , quotes(“ ”), and parenthesis() [44][31]

- Numerals which consists of symbols for 1 to 9 (፩,፪,፫,፬,፭,፮,፯,፰,፱), for multiple of ten(፳,፴,፵,፶,፷,፸,፹,፺) for 100 (፻) and for 1000(፳፻) [10] [44] [31].

Hence, the total number of symbols used in Amharic writing system rises to 310, the composition being as presented in *Table 2.1* [44][31].

Table 2.1 The total number of Amharic characters by type

No	Type of Amharic characters	No. of characters
1	Core characters(33x7)	231
2	Special characters(1x7)	7
3	Labialized characters	44
4	Punctuation marks	8
5	Numerals	20
	Total	310

Besides, as noted by [10][31] the original Amharic character set have no symbol for representing zero, negative numbers, decimal points, and mathematical operators for performing arithmetic competitions.

2.3 Evolution of the Amharic Braille

The history that would give birth to Braille is rooted deep in times long past after King Louis the Ninth of France suffered a crushing defeat in the Sixth Crusade [39].

As described in section 1.2 of the first chapter, Brail as modern touch education for visually impaired people started in 1917 E.C. Since then, it expands to different parts of the country. In between, many improvements have been made before it takes the current shape of Amharic Braille [44][47].

The first Amharic Braille introduced in Ethiopia was comprehensive and complete. It consider all characters that have similar sound such as (ሀ, ሐ, ኅ), (ሰ, ሠ), (ፀ, ኸ), (ፁ, ኹ) including punctuation marks which are used while reading, and all variants, except the Geez characters (like . Some part of the holy

bible: Lukas, Mark gospel, Jhone gospel, Solomon, Mezmure Dave were published by this Amharic Braille and reached to the blind society by the time [47].

In 1944 E.C, the world association of blind society made some change on the earlier Amharic Braille code. The change was made in two parts: 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 [44][47].

Later in 1948 E.C, the Amharic Braille was subjected for improvement for the third times when Sir Cluth Mecanize, the chair person of the world blind society, came to Ethiopia for visit. This time the improvement was made to further reduce the variant of Amharic characters [44][47].

After the 1948 amendment, the 3rd version of Amharic Braille was used for a long time until another attempt was started to modify in the 1980's E.C.

Once again after 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 [44].

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 [47]. Some of these rules and procedures which should be considered in adopting Braille to local language are the following [47]:

- 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, language that have the same sound or related language that are similar in sound, in character number and type should have similar Braille code structure. However, this does not imply, those languages that are different should have similar Braille code. 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.

As Nebyeleul [47] notes, the Amharic language has its root from Seabian 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.

The early Amharic characters adopt the English character which does not represent the right sound. And for the variant of Amharic character assigned the five English vowels such as ‘H’ to represent the three Amharic character of same sound (**ሀ**, **ሐ**, **ኸ**) [47].

2.4 The Amharic Braille Code character

Amharic Braille has been subjected for different improvement since its introduction in Ethiopia. After the last amendment, which was made in 1993E.C, the fourth version of Amharic Braille has been used for a long period of time throughout the county. This part of the study gives detailed description of the fourth version of Amharic Braille code that is in use currently and the focus of the current study. The other three versions of Amharic Braille are given in appendix I, 0, III and IV

2.4.1 The Fourth version Amharic Braille

After 12 years extended study on the third version Amharic Braille code, once again the Braille was revised for the fourth times in 1993E.C. The justification was come from the fact that blind kid should get the Braille simple to learn. Besides, the volume of Braille document, limited combination of dots for

characters, and difficulties to present as a legal document was the rational to reconsider for amendment. However, so far nothing is found best for blind than the Braille. The task force which was appointed by the blind association conduct thorough examination on the third Braille version and made improvement around three aspects [44][45]:

- Create equal number of Braille code that match to printed character.
- Modify the punctuation by taking most from the English.
- Include the Amharic numerals.

Accordingly, Braille dot combination has been approved and distributed to be used by different blind institutions has shown in

Table 2.1, Table 2.3 and

Table 2.2. To make the representation clear a character “U” in Braille would have the following appearance (see Figure 2.1):

```

1 ● ○ 4   1 ○ ○ 4
2 ● ● 5   2 ● ○ 5
3 ○ ○ 6   3 ○ ● 6

```

Figure 2.1 sample representation of Braille code

Table 2.1Forth version Amharic Braille character

1 st form (character)		6 th form (character)		1 st form (character)		6 th form (character)	
	Vowel				vowel		
ሀ	1:2:5 2:6	ሀ	1:2:5	ኸ	2:3:6 2:6	ኸ	2:3:6
ሐ	1:2:3 2:6	ሐ	1:2:3	ወ	2:4:5:6 2:6	ወ	2:4:5:6
ሐ	*1:2:6 2:6	ሐ	*1:2:6	ዐ	*1:2:5:6 2:6	ዐ	*1:2:5:6
መ	1:3:4 2:6	ም	1:3:4	ዘ	1:3:5:6 2:6	ዘ	1:3:5:6
ሠ	2:3:4 2:6	ሥ	2:3:4	ዠ	3:5:6 2:6	ዠ	3:5:6
ረ	1:2:3:5 2:6	ር	1:2:3:5	የ	1:3:4:5:6 2:6	የ	1:3:4:5:6
ሰ	*1:4:5:6 2:6	ሰ	*1:4:5:6	ደ	1:4:5 2:6	ደ	1:4:5
ሸ	1:4:6 2:6	ሸ	1:4:6	ጀ	2:4:5 2:6	ጀ	2:4:5
ቀ	1:2:3:4:5 2:6	ቀ	1:2:3:4:5	ገ	1:2:4:5 2:6	ግ	1:2:4:5
ቦ	1:2 2:6	ቦ	1:2	ጠ	2:3:4:5:6 2:6	ጥ	2:3:4:5:6
ተ	2:3:4:5 2:6	ተ	2:3:4:5	ጨ	1:4 2:6	ጭ	1:4
ቸ	1:6 2:6	ቸ	1:6	ጸ	2:3:5 2:6	ጸ	2:3:5

ሃ	*1:5:6	2:6	ህ	*1:5:6		ፀ	1:2:3:4:6	2:6	ፐ	1:2:3:4:6
ነ	1:3:4:5	2:6	ን	1:3:4:5		ጸ	*2:3:4:6	2:6	ጐ	*2:3:4:6
ፕ	3:4:6	2:6	ኝ	3:4:6		ፈ	1:2:4	2:6	ፑ	1:2:4
አ	1:2:3:5:6	2:6	ኦ	1:2:3:5:6		ፐ	1:2:3:4	2:6	ፕ	1:2:3:4
ከ	1:3	2:6	ከ	1:3		ቨ	*1:2:3:6	2:6	ቨ	*1:2:3:6

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

Table 2.2 Fourth version Amharic Braille vowels

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

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 2.3** are designed.

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 [44].

Table 2.3 Amharic Braille punctuation of the last version

Punctu.	Braille code	?	2:3:6	_	4:6 and 4:6
.	3	=//=	6 and 3	→	2:4:6, 2:5 and 2:5
:	6 and 3	X	1:3:4:6	←	2:5,2:5 and 1:3:5
÷	2:5	\$	4:5	↑	4:5:6 and 1:5
/	5 and 2	::	2:5:6	↓	4:5:6 and 3:5
-	4 and 1	!	2:3:5	እና ወይም	3:4
:-	3:6	...	3, 3and 3		
፤	2	()	2:3:5:6		
፤	2:3	[6 and 2:3:5:6		
«	2:3:6]	2:3:5:6 and 3		
»	3:5:6	*	3:5 and 3:5		
'	3:5:6 and 3				

Amharic 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 newspaper 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 was used for Arabic numerals are used in the fourth version [44]. Except the numerals translation mode 3:4:5:6 (for Arabic) and 1:2:3:4:5:6 (for Ethiopic) the same dot combinations were used as given in *Table 2.4*.

Table 2.4 Ethiopic, Arabic-Braille characters

$\underline{\underline{1}}$	$\underline{\underline{2}}$	$\underline{\underline{3}}$	$\underline{\underline{4}}$	$\underline{\underline{5}}$	$\underline{\underline{6}}$	$\underline{\underline{7}}$	$\underline{\underline{8}}$	$\underline{\underline{9}}$	$\underline{\underline{0}}$
1	2	3	4	5	6	7	8	9	0
1	1:2	1:4	1:4:5	1:5	1:2:4	1:2:4:5	1:2:5	2:4	2:4:5

2.5 Amharic Braille Writing and Reading system

Braille sheet is a thick paper or plastic material designed to withstand the pressure while one write and read the Braille code. The size of the page varies between documents produced by different means. In addition, the color of the paper varies, as it does not play any role in conveying the written information. The majority of Braille sheets, however, are either buff colored or white [6][14][21].

2.5.1 Braille writing

Braille is designed to be read by moving fingertips from left to right across the lines of dots. While writing Braille, move fingertips from right to left instead (on the reverse side), by physically pressing the dots into the paper so that they show up on the other side of the Braille sheet. The different methods used to write Braille, (including the manual) are listed as follows [21][36]:

- Manually, using a handheld stylus (to make the impressions) and a slate (to hold the paper) by physically pressing each dot into Braille sheet.
- With a Braille typewriter, this has one key for each of the six dots in a Braille cell (Mechanical).

- Using Braille printer that attached to computer system (electronic).

Until recently, creating books in Braille was time taking process. This is because translating a book from print to Braille required sighted transcribers to translate the book by hand. Later, improved optical character recognition (OCR) technology with computerized Braille printers has improved this process significantly. In OCR, rather than copying a book by hand, the book can be scanned, and then the result translate to the equivalent Braille code to print an embossed copy [21][36].

2.5.2 *Braille reading*

As indicated earlier, though there is no best system that can replace Braille, it has many difficulties as the number of possible combination is limited to represent all the print symbols. So, unlike print, reading Braille can be a little slower [11][44][21].

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 [43] . No documented information found in Amharic Braille user.

2.6 Braille computer representation

Braille has been extended to an 8-dot code, particularly for use with Braille embossers and refreshable Braille displays. In 8-dot Braille the additional dots are added at the bottom of the cell, giving matrix 4 dots high by 2 dots wide. The additional dots are given the numbers 7 (for the lower-left dot) and 8 (for the lower-right dot). Eight-dot Braille has the advantages that the case of an individual letter is directly coded in the cell containing the letter and that all the printable ASCII characters can be represented in a single cell. All 256 (2^8) possible combinations of 8 dots are encoded by the Unicode standard. Braille with six dots is frequently stored as Braille ASCII [39].

2.7 Representation of Amharic Braille character

Braille, like printed document, consists of different character 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 for a collection of printed character. This section of the study discusses the issue in detail.

2.7.1 Braille character

In Braille, printed text is represented by Braille characters. Each character is constructed as a set of six points arranged in two columns of three called a Braille cell. The cell has been set according to the tactile resolution of the fingertips of person. Each point position in a cell is identified by a number and can be either raised (a protrusion) or flat. A raised point is also called a Braille dot [44][21].

The dimensions of a Braille dot have strict regular structure. 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 [6][21]. In theory, the dimensions of Braille dots must fall within certain bounds as indicated in *Figure 2.2*. In practice, although most instances of Braille are close to these standards, there are slight variations between Braille produced by different devices [21].

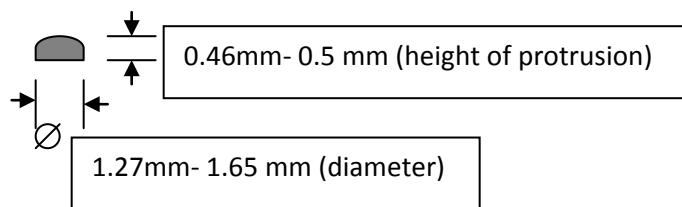


Figure 2.2 Braille dot dimension specified by library of Congress

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 [44][47]. In this study Braille characters are assumed to be composed of six points for Amharic Braille.

2.7.2 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 [21]. 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 (excluding one for space) possible combinations of dots which can be used to represent all Amharic printed symbols. According to the context and to the set of representation rules followed in a given Braille document; a particular combination of dots may correspond to more than one different printed character (at different occasions) (example 1, 1:2, 1:4 same for Arabic and Ethiopic number one, two and three). On the other hand, a printed symbol may require more than one Braille character for its representation (1:2:5 and 2:6 for **ሀ**, 1:2:3 and 2:6 for **ለ**, 1:2:6 and 2:6 for **ሐ**) [44][45]. Furthermore, there are different representation rules for other types of documents, like those containing music or mathematical expressions [44][21]. The following discussion gives overview of the most commonly used Braille rule and applied in Amharic: Grade 1 and Grade 2 Braille.

Amharic Grade 1 Braille

In *grade 1* Braille, each printed letter corresponds to one Braille character (e.g. 1:4:6, 1:2:3:4:5 and 1:2 for **ሸ**, **ቀ**, and **ብ** respectively) [44][47]. 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) [44][21][47].

It is not possible to represent the large number of existing symbols, (particularly language like Amharic character) by a combination of dots in a single Braille character. Therefore, some symbols (the 34 x 6

form) are represented by two Braille characters [44][21][47]. Furthermore, the digits "1", "2",... "9", "0" are represented by the Braille characters corresponding to the range of letters "a",..., "j", but preceded by the numeral symbol (3:4:5:6) [13][21]. 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) [44][47].

In English language, another mode symbol is used to indicate the capitalization of a letter. Similarly, mode symbols are used in many languages to denote that a particular accent should be placed on the character that follows [21].

Grade 2 Braille/contraction

The letter-by-letter representation of a word as in grade I 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 [44][21]. The rule for using them is referred to as grade-2 Braille (also called contracted or literary Braille) [21].

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 [21].

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 [44][45]. Except for “ena”/”weyem” there is no grade 2 Braille in Amharic.

2.8 Features of Amharic Braille Code

The Amharic Braille character composed of three parts: Braille character for printed symbol, punctuation marks and numerals [44][47].

All basic Amharic character represented with a single Braille character (see *Table 2.1*). Each Amharic character has seven variant. Except the sixth variant, the remaining sound six sounds add the particular vowel code to get the required sound in a character [44]. For example:

- Braille dot 2:3 is vowel for the first variant in all the character. so to write first form of Fidel ብ (1:2)→ብ(1:2, 2:3)

The Amharic Braille also consists of different punctuation mark. 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 2.3*). The majority of the punctuation marks represent by a grade 1 Braille [44][47].

Amharic Braille use two type of numerals: Arabic and Ethiopic. In English Braille the numerals 1, 2... 0 uses the code for a, b...j. The same dot combination is used in Amharic Braille also. The numerals translation mode remains the same in Amharic Braille. For example:

- Arabic Braille numerals mode: 3:4:5:6, and Ethiopic →1:2:3:4:5:6,
- To write 2: 3:4:5:6 and 1:2
- To write ፩: 1:2:3:4:5:6 and 1:2

2.9 Nature of Amharic Braille embossed

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. In addition, the color of the paper varies, as it does not play any role in conveying the written information

[21].

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. 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 [21][41].

2.10 The Braille OCR system

As pointed out earlier, there has been a great deal of research work done in the field of optical recognition of Braille for automatic reading in different parts of the world. In order to show issues considered in connection to Braille, the next section of the studies discuss important features of embossed Braille document that will be helpful in the subsequent recognition process.

2.10.1 Characteristics of Automatic Braille Reading system

As the Braille documents are not designed to convey any visual information, their color is uniform or not significant. The information is not printed in a contrasting color as does with printed document (ordinary document) [21].

Accordingly, in an attempt to design automatic reading for Braille document the first problem is to identify the tactile information visually. The issue in this case is the separation of the protrusions (called raised dot) on the paper sheet from the flat background [21][36].

Second, the identification of Braille dots from the differences in the intensity of reflected light is not straightforward. As the Braille paper is only used for tactile purposes, the quality of its appearance is not important. Hence, there are sometimes visual imperfections due to small dark fiber usually associated with

cardboard and dirty marks that may also be present due to repeated use [21][36].

Moreover, damaged Braille documents with punched holes and less prominent dots due to heavy or repeated use compound the problem and negatively influence, the effect produced by the lighting [21][36].

Notwithstanding the above fact, many of the Braille documents are produced as inter-point (double-sided), which posing another problem on the process of recognition. This complicates the dot extraction stage because shadows and bright areas are produced from both sides [6][21][36].

Once the Braille dots have been located in the image, the next task is to identify the characters. In a Braille system a character is represented with six dots, which is called cell, arranged in two columns. The problem is to identify which dots belong to which Braille cell. In theory, Braille cells are aligned in the horizontal and vertical directions in the form of a grid. However, noticeable deviations are there due to the production process or distortions introduced in the image of the document [21][26].

Eventually, Braille characters have a more regular structure than printed ones. Their size and style is fixed throughout a document, although there are certain variations between different documents. The spacing between Braille dots is quite regular and can be used to segment characters [6][21]. Nevertheless, Braille characters are not single connected entities and do not have distinguishing structural differences between them (they are all combinations of dots) which affect the recognition and verification process [13][26].

2.10.2 Optical Braille recognition system Overview

For better analysis of the different aspects in existing solutions, as a starting point the basic steps in Braille recognition process considered in five stages including: *Image Acquisition/Digitization, Pre-Processing, Segmentation, feature extraction and recognition* [21]

2.10.2.1 Image Acquisition/Digitization

The first step towards digital document image processing is obtaining an electronic representation of a document. It is a process of capturing a Braille page and converts into a digitized image array. As it was mentioned above, the nature of this kind of document is quite different from that of a printed one. The protrusions (raised dots) create a light area next to a dark one (shadow) while the gray-level value of the background is somewhere in-between, this is visually examined in single-sided Braille document. This day's, the increasing popularity of flat-bed document scanners has provided a cost-effective and practical alternative to camera for digitization purpose [6][21][23].

2.10.2.2 Image Pre-Processing

This stage modify gray-level image of a Braille document and prepare the pixel values of the digitized image for subsequent operations. This includes contrast enhancement and adjustment, skew and noise removal, and edge enhancement (edges sharpening) of the Braille [14][21].

Noises generally manifest themselves as random fluctuations in gray-level values superimposed upon the “ideal” gray-level value, and it usually has a high spatial frequency. Skew may be introduced during image acquisition by not carefully aligning the document or it may arise due to a production fault. Large skew angle negatively influence the recognition process, as the dots will not be aligned in rows as expected [14][21].

Another form of pre-processing is edge enhancement to sharpen the fine details of the image that has been blurred in order to improve the separability between the regions corresponding to dots and the background. The solution include: subtract a locally averaged image from the original in order to eliminate this problem, applying second degree convex two-dimensional function based on a second-degree polynomial function, and median filtering adaptive thresholding process [17][21].

2.10.2.3 Image Segmentation

At this stage, the regions in the image corresponding to Braille dots are identified. In a binary image, dots can be identified as small connected components [26]. It should be noted that this situation is valid only when single-sided Braille documents are encountered. In this case, the above approaches are able to identify dots from either the dark (shadow) or the light areas in the image which, after thresholding, are represented as connected components [21].

In a grey-level image, the identification of dot regions needs more careful consideration. It is necessary to have a specification of the image area corresponding to a dot. A dot is represented by a combination of a dark and a light region. The intermediate grey-level pixels of the background surround and, possibly, also separate the pair. Naturally, the orientation and order of the light-dark pair of regions depends on the direction of the illumination. In images of inter-point Braille documents, however, the protrusions give rise to region pairs in which the order is opposite to that of the region pairs produced by depressions. The consistency of the appearance of gray-levels at the positions of depressions and protrusions renders the use of template matching a possible option [21][36].

2.10.2.4 Feature Extraction and Recognition

Feature extraction is a representational mechanism of the Braille image. So, the function of feature extraction is to extract the Braille dots from the binary image and grouping them into cells. Then the identity of each of the Braille detected is recognized, but no attempt is made to interpret the Braille character. Since each character is distinguished from the rest by its unique combination of dots, it will suffice to identify the (relative) positions of its dots. Then, the character can be described in terms of the numbers of dot positions.

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.

Alternatively, the region in the image corresponding to a compartment is searched and the number of pixels corresponding to dot components is counted. If the number exceeds a pre-specified threshold then the compartment is considered to contain dots. In any case, the recognized character can now be represented as a binary number with six bits.

In other approaches the Braille characters identified in the image are described explicitly or implicitly by the positions of the rows and the columns of a cell. In these cases it is necessary to check along specified directions for the existence of dots. This is accomplished with gridlines construction. So the possible positions of Braille dots are determined by the intersections of horizontal and vertical grid lines. 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.

2.11 Review of Braille optical recognition work

This section of the study discusses the work of different researcher in the area of Braille recognition problem domain as it is summarized in Table 2.5.

Table 2.5 Summary of Braille optical recognition research work

Year	Title	Country	Main Approach used	Result	
				Single	Double
1994	Optical Recognition of Braille Writing Using Standard Equipment.	Belgium	Digitization-mask method Segmentation-Gridlines Recognition-region based(local correlation)	Test-1 no-error Test-2 99.75% Test-3 unsatisfied	
1995	A System for Converting Braille into Print. (English language)	England	Recognition-context analysis Based on Finite state approach	-	-
1995	Analysis of Scanned Braille Documents (English language)	Manchester	Binarization- threshold Dot identification- template matching Segmentation- analysis of character line and horizontal spacing b/n columns of dots Recognition- region based (divided into six compartment)	-	98.5% 97.6%
1996	A Braille OCR for Blind People (English language)	Spain	adaptive thresholding Mesh-grids Interpretation-Braille Alphabet	-	-
1999	Regular feature extraction for recognition of Braille (English and Chinese language)		Binarization-Global threshold Feature extraction- Chain Code algorithm Interpretation-centriod based on boundary coordinates Braille dictionary	-	-
2004	A Software Algorithm Prototype for Optical Recognition of Embossed Braille (English language)	New Zealand	Binarization- thresholding Recognition-probabilistic neural network Transcription- gridlines	99%	-

2005	<i>Image Processing Techniques for Braille Writing Recognition. (English character)</i>	<i>Spain</i>	<i>Tool- MATLAB programming Binarization-dynamic thresholding Segmentation-adaptive Braille grid</i>	-	99.9%
2007	<i>An Arabic Optical Braille Recognition System. (Arabic language)</i>	<i>Tunisia</i>	<i>Binarization-global threshold Dot Parts Detection-thresholding Recognition-defined region and horizontal projection</i>	99%	99%

2.11.1 Optical Recognition of Braille Writing Using Standard Equipment.

This study is an early attempt to recognition of Braille using a commercially available scanner. The steps followed in the study are summarized as follows.

2.11.1.1 Image preprocessing

Braille pages are never perfectly flat. This is due to the tension in the paper caused by the production of the Braille dots. It introduces low frequency distortions (shadows) that can be eliminated by subtracting a local averaged image from the original. The image will be divided in to sub images to minimize the influence of deformation of grids (Local linearization) [21][28]. If the grid is very distorted, the sub images will become small. The calculated grid must be aligned with the main axes of the text. It is possible to work with a rotated grid or to align the complete image to orthogonal axes first. The latter has certain advantages in the organization of the data so this solution was chosen [28].

The Discrete Fourier Transform (DFT) can be used to calculate the rotation angle, but the image's structure causes important and recognizable data peaks in the DFT image to lie too close to the origin. This gives an error on the calculated rotation angle, and a further refinement is necessary. They followed a different approach. Using the deviation over the sum of the rows, the image was slanted over an angle [17][28].

2.11.1.2 Segmentation

In the digitized Braille page, dots are represented with light and dark areas. They classify these areas by making a three-value image in which all pixels within a threshold value are made zero. Pixel values above this range are made +1 and below this range -1 (n is typically between 1 and 2, and allows adaptation to different illumination conditions) [28].

The actual center of the dots can be found by correlating the image with a mask as shown in *Figure 2.3*. This mask is an absolute simplification of a Braille dot. The vertical size of the mask depends on the size of the Braille dots and equals the distance between the centers of the dark and light area of an average dot. After correlation, we obtain an image with five values. Regions with value -2 (recto) and +2 (verso) are called core regions, regions with value +1 (recto) and -1 (verso) are called side-lobes, and 0 is background.

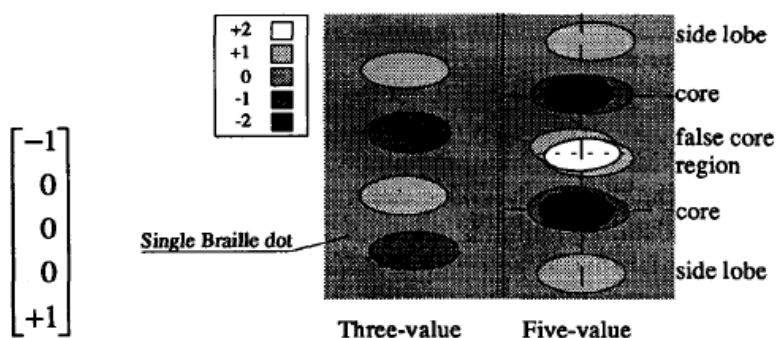


Figure 2.3 Correlation mask for an image with a distance of five pixels between the center of the dark and light zone[left]and Neighboring dots introduce false core regions or ghost-point[right].

(source: Mennens, J., et al, "Optical Recognition of Braille Writing Using Standard Equipment", IEEE transactions of rehabilitation engineering, Vol. 2, No. 4, pp. 210, December 1994. Fig. 6)

2.11.1.3 Feature Extraction and Recognition

Grid lines are searched by making histograms of rows and columns in the five-value image. These histograms have five bins (-2, -1, 0, 1, 2) [28].

- Firstly, the histograms for the rows are calculated. When a plot is made of the

number of pixels in bin (+2) and bin (-2) (bin-curves), they show maxima where rows of recto and verso dots reside.

- Next, the histograms of the columns are calculated. Because the previously found horizontal grid lines are eligible to contain Braille dots, it is sufficient to consider these lines only in creating the histograms.
- Another improvement was made by making combinations of values found in different bins.
- In the case of recto dots, the presence of pixels with a value -2 was rewarded and the presence of pixels with a value -1 was penalized.
- These pixels are always found around the center of a Braille dot. This procedure will result in narrower peaks.
- After the vertical grid lines are found, the horizontal lines searched again, but in a reduced image that consists of the vertical grid lines only. Because grid lines were calculated separately for each sub image, the data needs to be regrouped in strips, to get an overview of the complete image.

Before we can do any interpretation of the Braille dots, it is a must to find all atoms in the vertical and horizontal grid lines. Distances and tolerances on line positions are always calculated on a local basis, i.e. the routine learns from local conditions to decide where candidate lines can be found [28].

2.11.1.4 Test Results and Summary

When all atoms are found, a small test (a minimal local correlation) will check for the presence of a dot on the intersection of two lines. These dots are grouped in a binary Braille character set (Binary Braille: each dot is represented by a bit position), which can be translated using an appropriate translation table to any other code [28].

The algorithm was tested on various types of Braille: single-sided and double-sided and every set contained an equal amount of samples. An average error rate of 0.25% on a character basis was reported

[28].

Using a commercially available scanner and a set of fairly basic calculation methods to obtain a maximum conversion speed, a text containing Braille characters can be converted to any other character code. The most important characteristic of the routines is that on a local basis, without losing contact with the global context, and try to postpone complex calculations until the data set has been significantly reduced [28].

2.11.2 Analysis of scanned Braille documents

This paper attempted the use of commercially available flat-bed document scanner for the first time to Braille document recognition. It argues that, the use of the camera introduced complexities such as aberrations, considerable noise and the need for image registration. So to overcome these problems, earlier study used computationally expensive procedures [36].

After the Braille document is scanned, the system proceeds to identify the locations of the protrusions and the depressions. To do this it exploits the differences in grey levels in the image. It uses simple rules to distinguish between Braille dots on the two different sides. Then, the dots of each side are grouped together to form characters which are then recognized and encoded [36].

The system implemented in four steps: image acquisition, identification of Braille dots, segmentation and recognition of the characters.

2.11.2.1 Image Acquisition

The Braille document was scanned at 100 dpi and a 16 grey level image. Under the scanning configuration used, the grey level values of the background pixels were from 10 to 12. Lighter areas have grey levels with values ranging from 13 to 15, and the darker areas corresponding to shadows had grey level values

less than 10 [21][36].

2.11.2.2 Identification of Braille Dots

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. The depressions produce the opposite. (see *Figure 2.4*)

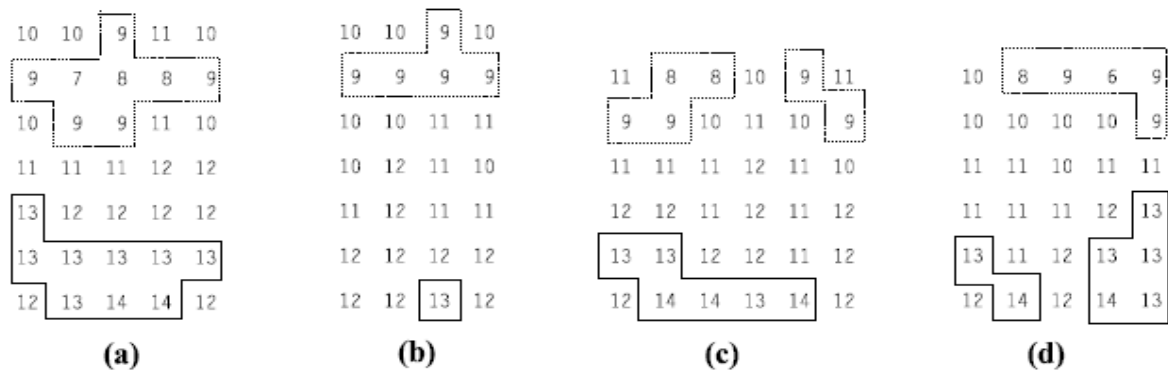


Figure 2.4 Analysis of scanned Braille documents.

Since there is such a consistency one could perform a template match to identify the locations of the depressions and the protrusions [21][36].

The algorithm to identify protrusions and depressions continues as follows. Each of the dark regions is considered in turn and a check is made to determine whether a light region exists above it. If it exists within the limits of the expected Braille character height then a depression is found. Otherwise, a check is made to determine whether a light area exists below the dark one. All protrusions and depressions identified are then written onto a new blank image independently. Then, the two images can be analyzed, each as containing only protrusions which form Braille characters [36].

2.11.2.3 Character Segmentation

The first action to analyze the resulting image from the earlier step is to locate the lines of Braille characters. Each character in a line has three rows where dots may be present. The distance between dots in consecutive rows of the same character line is determined and all the text lines that have dots on all three rows are detected. Then, knowing the average line height, the rest of the text lines together with the blank lines are also detected [21][28].

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 fact that a Braille character may have dots in only one column also contributes to a variety of horizontal spacing in a text line [36].

2.11.2.4 Character Recognition

Once the position of each Braille character in the image is known, this step is concerned with the recognition of each character and encodes it in a suitable form for further processing. 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. The binary number can then be used to identify the Braille character in the character set used [36].

2.11.2.5 Results and Discussion

A preliminary result from images of inter-point buff-colored Braille documents was promising. At this experimental stage, an average of just over 98.5% of the protrusions and 97.6% of the depressions are correctly recognized [36].

This paper concentrates on the application of OCR techniques to inter-point Braille documents. As a whole, the approach proves the feasibility of a cost-effective, fast and easy to use Braille reading system [36].

2.11.3 A Braille OCR for Blind people

The purpose of this work was to develop an optical character recognizer (OCR) for the Braille code. The research gone through three main stages, including [23]:

- Processing of the scanned images to locate the positions of the Braille dot.
- Processing of the resulting dot from the previous process for Braille mesh determination.
- Application of the Braille alphabet to convert the located points into an ASCII text.

2.11.3.1 Point Localization

This stage starts with an observed model of the pattern to be recognized. That is, in a double-sided two kinds of points distinguished: mountains (protrusion) and the valleys (depressions). This pattern can be observed in *Figure 2.5* [23][21].

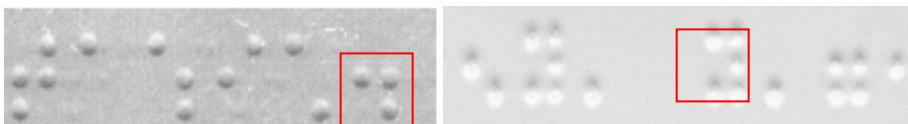


Figure 2.5 Braille image showing left raised dots (protrusions) and right valley (depressions).

So at this point in order to convert the zone in to black and white spots, the research applies two adaptive thresholds which are computed from luminance histogram.

Thresholding is an image processing technique for converting a grayscale or color image to a binary

image based upon a threshold value. Image thresholding is very useful for keeping the significant part of an image and getting rid of the unimportant part or noise. If a pixel in the image has an intensity value less than the threshold value, the corresponding pixel in the resultant image is set to black. Otherwise, if the pixel intensity value is greater than or equal to the threshold intensity, the resulting pixel is set to white. Thus, creating a binarized image or an image with only 2 colors (black (0), and white (255)). However, this holds true under the assumption that a reasonable threshold value is chosen [15][21].

In the research, the thresholds are computed from the luminance histogram. They are found as the points that leave above/below a given percent of the total area of the histogram [21][23].

Afterwards, compute the centers of the black and white spots and consider them as black and white points. Then, look for couples formed by a white point above and a black point below. And also mark the couples formed by a black point above and a white point below as valley-points. It adjusted the thresholds in the distances necessary for two points to be considered as a couple [23].

The method had two problems: first, some points are lost and second, some false points are produced. The errors are due to spots that are very near to each other and that get joined after the thresholding process (this happens especially when a mountain and a valley are very near). The solution was to put all the intelligence in the next process (the mesh detection), which was able to remove the false points and to find where the points lost [23].

2.11.3.2 *Mesh detection*

This process detects the positions and directions of the lines and columns of the text that make up the Braille mesh. A Braille character has a mesh of 6 points. Each point may or may not be active. So having the entire mesh, it is possible to go over it and inspect the presence of spots near each potential point. This

algorithm recovers the lost points. However the false points are the one that do not observe the distance rules. The algorithm is such that simply excludes them [23].

The algorithm chooses a starting point and goes moving the standard distances to construct the entire mesh [23]. At this point a key fact is to avoid choosing a false point as the starting one.

Another important issue is to detect the relative position of the starting point in its character that has 6 possible positions; by simply, suppose all the possible positions and then use the method explained above of maximal counts (counts of points found with this suppositions). First, determine to which column (out of 2) belongs the point and then to which row (out of 3) [23].

The final stage is the construction of the entire mesh beginning from the starting point. This movement is accomplished in an adaptive fashion. The method is to move a mesh of size 6 (a one character mesh). After the movement looking for the next character, if there are points near the mesh ones, then, correct the mesh position based on these points. This continues until the right limit of the image [23].

2.11.3 Translation into ASCII

Having done the previous stage, the Braille text now represented as characters of six points (six bits that may be 1 or 0) which are ready to convert into ASCII by applying a Braille alphabet [23].

2.11.4 Regular Feature Extraction for Recognition of Braille

The aim of this paper is to present an automatic system to recognize the Braille pages and convert the Braille documents into English/Chinese text for editing. The proposed System in this study consists of six operations: Scene constraints, Image acquisition, pre-processing, segmentation, feature extraction, and

Interpretation [14].

2.11.4.1 Scene Constraints

It is a preliminary process to exploits and imposes the environmental constraints such as, illuminate with a yellow polarized light source, placed at an angle about 45 degrees away from the page top [14].

2.11.4.2 Image Acquisition

The capturing device used in this experiment is a digital camera, which is placed directly above the Braille. The image captured with 512 * 512 pixel resolution with each pixel representing an 8-bit gray scale value [14].

2.11.4.3 Pre-processing

Modify and prepare the pixel values of the digitized image for subsequent operations. The pre-processing operation consists of two sub-operations: noise filtering and edge enhancement [14][21].

A low-pass spatial Gaussian filter is applied to the image to attenuate the high spatial frequency noise from the image while at the same time preserving the detailed edge information of the Braille dots. The objective of edge enhancement is to sharpen the fine details of the image that has been blurred. The process performs two independent filtering operations using convolution Sobel kernels denoted X and Y, the resulting values are obtained by using the convolution formula [14][21]:

$$\text{output} = |\text{convolute}(\text{input}, X)| + |\text{convolute}(\text{input}, Y)|$$

2.11.4.4 *Segmentation*

This step separates the acquired image into foreground (dots), and background regions. This operation also separates the front-faced dots and back-faced dots [14].

This study consider a histogram made up of only those pixels that lie at or near the edges of the Braille dots as the digitized Braille page images are noisy. To determine which pixels lay on or near an edge, the Laplacian of Gaussian operator is first applied to the entire image to identify the edge pixels. The average gray-level value of these edge pixels will then be used as the global threshold value [14].

2.11.4.5 *Feature extraction*

The main function of this process is to extract the Braille dots from the binarised image. 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 [14]. The chain code was devised in 1961 by Herbert Freeman. This method recodes the boundary of a region into a sequence of octal digits, each one representing a boundary step in one of the eight basic directions relative to the current pixel position [17]. The Chain Code algorithm detects the boundary coordinates of the Braille dots. As the coordinates of the dots are detected, the diameter of the dot can also be deduced. Based on the boundary coordinates information and the illumination characteristics, two standard templates were constructed to represent the front-face dots and the back-face dots. The templates were then applied to every position of the image and evaluate the correlation at each pixel position. Based on the correlation values, the front-faced and the back-faced dots are extracted [14].

2.11.4.6 *Interpretation*

The work of this phase is to group the Braille dots into cells and converts them into their English

correspondences. To do this, for each dot, the distance between the centroid and the four possible neighbors are determined. From this information, the dots are grouped into cells. Within each cell, 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 checked against an English dictionary [14].

Using both single-sided and double-sided Braille pages as the specimen, the system had been proved to be 100% and 97% accurate respectively. The system takes advantages of the regular spacing between Braille dots within a cell, and the regular spacing between cells [14].

2.11.5 A Software Algorithm Prototype for Optical Recognition of Embossed Braille.

The rationale behind this study was that most of the systems developed before concentrated on the translation; however, the formatting of the document is also critical when reproduction is involved [41].

The paper attempted to devise a method of Braille recognition that aims at preserving the formatting of the Braille page, as well as the efficiency and accuracy of the recognition algorithm [41].

The proposed system involves three steps: Half-character detection, half-character recognition, and text file transcription [41]. The block diagram for the proposed system is presented in *Figure 2.6*

- The half-character detection algorithm determines the whereabouts of the characters by detecting the possible dot positions.
- The half-character recognition determines the half-character that the dots represent by using a probabilistic neural network.
- After the half-characters have been recognized, the grid will be determined with the text file

transcription algorithm to produce a Braille text file where the formatting is preserved as much as possible.

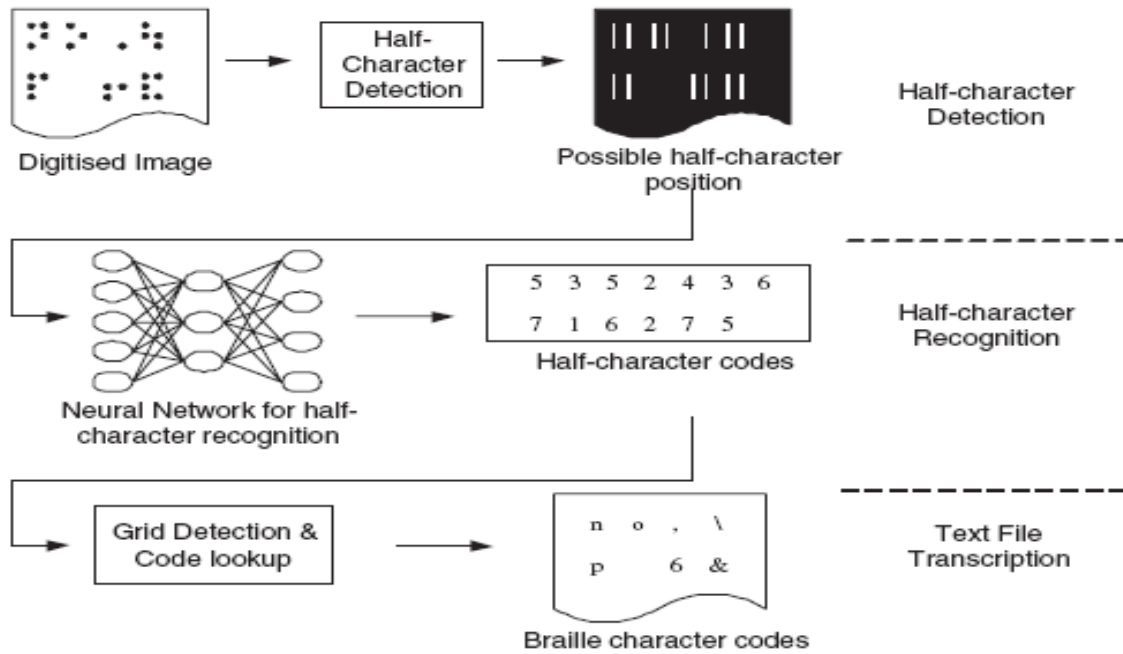


Figure 2.6 A Software Algorithm Prototype for Optical Recognition of Embossed Braille.

(source: Wong, L., W. Abdulla, and S. Hussmann. "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", the 17th conference of the International Conference in Pattern Recognition, Cambridge, UK, pp. 24, August 2004. Fig. 1)

2.11.5.1 Half-Character Detection

In this step, the row of pixels is changed into a two-tone image to distinguish the pixels that are likely to be part of the shadows of the dots from the others. The result from this process is two-tone images and the position of the character [41].

Compared with the conventional method of using a mask to determine the position of the dots, thresholding and buffering, although it does not give the exact location of the dots, can find the area where processing is needed to determine what character is present [41].

2.11.5.2 *Half-Character Recognition*

In this step further processing is made on the buffered two-tone image to recognize the half-characters detected from the previous step. The half-characters are processed and classified as one of seven possible arrangements. The seven arrangements come from having three possible dot positions in each half-character, without considering “empty half character”. This processing is done by using a probabilistic neural network, with [41]:

2.11.5.3 *Text File Transcript*

This process determines the position of the original parent whole characters. To do this, it takes advantages of the fact that, the Braille format has very strict rules on the spacing

between dots within a character and between neighboring characters, thus these rules can be used to help determine the positions of the characters in the document, as well as whether the half-characters are the left half or the right half [41]. By looking at the distances between the dots, the intra and inter-character spacing were estimated. Using this information a grid was formed to all possible positions for the characters in the document. Once the grid has been estimated, the position of the characters with respect to the image can be used to calculate the positions of the characters in the document [41].

This study proposed an algorithm to optically recognize Braille documents and transcribe documents into a computer file. The algorithm processes the image one row at a time and uses thresholding to pre-process the image. The processed sections that contain the Braille characters are then recognized using a probabilistic neural network. On experiments performed on single sided computer embossed documents, more than 99% accuracy was achieved [41].

2.11.6 *Image Processing Techniques for Braille Writing Recognition.*

This study attempts the development of BrailLector, a system able to speak from Braille writing. The system applied dynamic thresholding, adaptive Braille grid, recovery dots techniques and TTS software (Text-To-Speech) presented in *Figure 2.7* [32].

The paper first describes the characteristics of the database created for such purpose, followed by, image processing techniques used for dots detection and recovering. Then, explain the conversion from Braille text to standard text. The proposed system block diagram presented in

2.11.6.1 Database

A big database was created in order to check the global system with as many characters (30862) from 26 Braille sheets; to ensure the correct testing of the developed system and a good analysis of the error variance [32].

The mechanism used for image acquisition has been a flat-bed scanner with different resolution than 100 dpi since it uses interpolation methods to resize the input image [32].

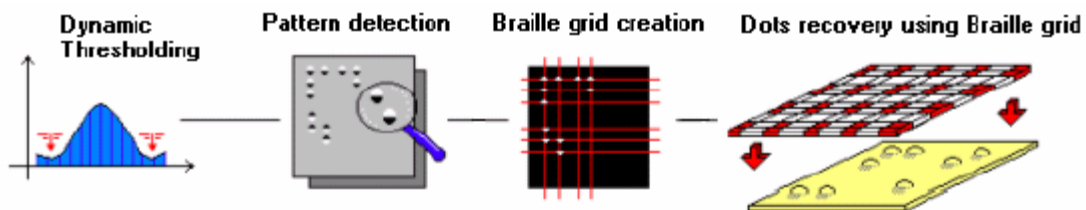


Figure 2.7 Image Processing Techniques for Braille Writing Recognition

(Source: Wong, L., W. Abdulla, and S. Hussmann. "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", the 17th conference of the International Conference in Pattern Recognition, Cambridge, UK, pp. 308, August 2004, Fig.2)

The different image processing steps of the scanned image for its translation include the following.

2.11.6.2 Thresholding

This is a primary process that applies innovative thresholding method to extract the useful information of Braille images. Once the optimum area of Braille spots (to reduce the number of wrong symbols detected) are known, an iterative algorithm looks for the best threshold according to these areas of Braille dots. This is done, to get the optimum levels to separate black, white and grey [32].

2.11.6.3 *Pattern detection block*

After the first step, no useful information is rejected, and in this step it takes advantage of the shadows protrusions which make dot patterns. So this difference facilitates the separation between front and back side dots [32]:

The goal of this secondary process is to separate each side of the document in different images. The algorithm consists of a “shift and overlap” process since it only moves the spots downwards or upwards and carries out a logical and. As it avoids the sequential reading of the image matrix approach was fast, very simple and efficient [32].

Skew detection: this stage applies horizontal histogram and mass centers calculation to detect skew angle and correct by rotating the original image [32].

Problem: all dots are not detected with overlapping process, some of them are missed, either because they are very small or their shadows do not overlap with only 4 pixels. For this reason, a new stage was added to the system: Braille grid [32].

2.11.6.4 *Braille grid.*

In order to produce the mesh from the detected dots, they develop adaptive algorithm. The algorithm

works as follow [32]:

- Builds columns in a first stage: since distances between points are normalized, the process begins searching for groups of dots in the same vertical plane that respect these distances.
- Then, builds the columns according to the pattern of Braille columns adapting itself to the layout of the document. This result, a flexible mesh that tolerates small differences between columns. The process for the rows is quite similar but in this case the pattern to search is a Braille row.
- Detected columns and rows will be arranged together to create the final structure. As the grid was flexible, it makes suitable for copying Braille sheets without losing the format.

2.11.6.5 Dots recovery using Braille grid

After mesh building, all valid Braille positions are known; intersections between rows and columns define a valid position for a Braille dot. So in this step [32]:

- First, dilation techniques are used to expand the search zone.
- Then, fit this image on the dots image after thresholding in order to check in detail those positions where dots were not found.
- Only potential positions of dots are checked; this means time saving and efficient search. Original dots will be recovered (they belong to correct Braille positions) and false dots will be discriminated as they are out of the valid places for a Braille dot.

2.11.6.6 From Braille to Normal Text

In this process the final image that contains the Braille dots represented by spots, is analyzed and text is segmented in rows and characters. Segmentation process takes advantage of the Braille mesh again as it marks all the positions of Braille dots. Every character is converted into a binary number according to the active dots. The process consists of reading character by character and each one of the six positions that make the basic cell. The output of this step is a file with each character coded like a binary number [32].

This paper explained the development of an automatic system for translating Braille text to normal text or voice. To achieve this the system apply, dynamic thresholding and adaptive Braille grid had been used, adding some intelligence to the global process and making it able to detect dots in both sides of the document with only one scan. It is a robust application with 99.9% of correct symbols. The conversion time is 26 seconds for double-sided documents by MATLAB programming language [32].

2.11.7 An Arabic Optical Braille Recognition System

This paper looks at developing algorithms to recognize an image of embossed Arabic Braille of both single-sided and double-sided material obtained by a regular scanner; the research was published in 2007 [6].

The research comprises the following steps [6]:

- Capturing an image of a Braille document page using a flatbed scanner.
- Converting the image to a gray color.
- Followed by cropping any white or black frames.
- The result from the previous step is thresholded so that only three classes of regions exist: dark, light and background.
- In case of having a tilted paper, de-skewing is applied using a binary search algorithm.
- Perform initial identification of Braille dots (Having labeled each of the different types of regions)
- Finally, Braille cells recognition.

2.11.7.1 Image acquisition/Converting the Image to Gray Level

To reduce complexity the first step in this system converts colored scanned images to gray level so that any pixel value in the image falls within the range 0 – 255 [6][17].

2.11.7.2 Image pre-processing

a. Cropping the Image Frame

Images suffer from having either black or white frames that would affect thresholding, so cropped before proceeding forward. To handle the problem, calculate the average gray level for the whole image and then for each of the rows and columns separately [6].

b. Image Thresholding

Once again, at this step the research involved examining each pixel in the 2-D array representing the scanned Braille document and classify in to three categories with threshold:

- (1) Gray level pixel elements having values 32 and above are considered bright and given a threshold value +1.
- (2) Gray level pixel elements having values 23 and below are considered dark and given a threshold value -1.
- (3) Gray level pixel elements having values between 23 and 32 are considered gray and given a threshold value 0.

The algorithm to handle works as follows; after converting the image to gray level and removing highest and lowest values to leave out distinct values, the average gray level for the whole image is calculated.

After that both a and b values are calculated according to the following equations [6][17]:

$$a = \text{mean} (\max(I) + \text{avg}) / 2$$

$$b = \text{mean} (\min(I) + \text{avg}) / 2;$$

Where:

- a highest value and b lowest value
- I is an array representing the whole image.
- $\max(I)$ and $\min(I)$ represent the highest and lowest values respectively in each column of the array I.

The classification parameters L and H are calculated according to the following equations:

$$H = \text{avg} + (\max_avg - \text{avg}) / 3;$$

$$L = \text{avg} - (\text{avg} - \min_avg) / 3$$

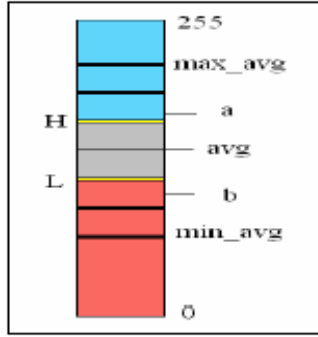


Figure 2.8 Arabic Braille Recognition proceeding

Where:

- min_avg represents the average of all values less than b
- max_avg represents the average of all values higher than a.

The image is then classified according to L and H values as shown in *Figure 2.8*.

- Values less than L are considered dark.
- Values higher than H are considered bright.
- Values between L and H are considered gray (background).

Histogram of the scanned Braille document show a Gaussian distribution ($1.5 \cdot \sigma$, where σ is the standard deviation), which gives good results to find L and H values [6].

c. Image De-Skewing

The research has developed a binary search algorithm to correct any de-skewing in tilted scanned images. According to the report, the maximum degree of recognizing a de-skewed image is 4 degrees from either the left or the right side.

2.11.7.3 Segmentation

a. Dot Parts Detection

The study preferred the detection of Braille dot parts first than detection of the whole dot. It assumes that the average dot height is 8 pixels and each dot is composed of a bright and a dark region with a small

space between them. This is shown clearly in *Figure 2.9*. The implication is that if the bright region comes at top where the dark one comes at bottom then this is a recto dot (protrusion, raised dots part), the contrary situation results in a verso dot [6][21].

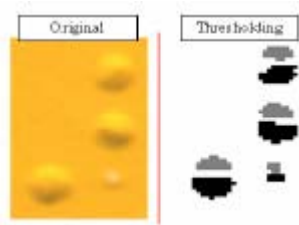


Figure 2.9 Arabic Braille document recognition system

The algorithm in this process ends with recto and verso dots that can separate them in two arrays. Errors from this step may occur and correct the error with either of the following methods [6]:

- Local Measure: select the best column within all columns that are 8 pixels below it.
- Global Measure: all columns are classified into three categories: columns holding recto dots, columns holding verso dots, or undefined columns. The second method was found excellent results.

b. Whole Dot Detection:

This stage involves the following steps [6]:

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.)

2.11.7.4 Braille Cells Recognition

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 [6]:

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 [6]:

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 [6].

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 [6] [21] [36]:

$$\text{Decimal-Code} = b_1 + b_2 * 2 + b_3 * 4 + b_4 * 8 + b_5 * 16 + b_6 * 32$$

The final step is to convert the decimal code to its corresponding Arabic letters to get the translation.

In this research, the proposed system has used some new techniques to recognize Braille cells using a standard scanner. The system has been tested for both single and double sided, written in the Arabic language and scanned with different scanners. And 99% of the dots are correctly recognized for both single and double sided Braille.

Chapter Three

3. Braille Recognition Technique

3.1 Introduction

The present research is concerned with the adoption of different Braille image recognition techniques for Amharic Braille documents. For this purpose approach that are used to Braille recognition such as image thresholding/Binarization, segmentation, feature extraction and recognition are reviewed. The part also discusses the neural network techniques to classify the given Braille-to-print character.

3.2 Digitization/Image Acquisition

In Optical Braille Recognition (OBR), digitization enables to produce a digital image of the scanned document in the form of bitmap. Currently, there are different alternatives used to digitize Braille document: from digital camera to flat bed scanner, hand held scanner or a sheet-fed scanner. A flat-bed scanner is a cheap alternative which can be used for so many other applications of this sort.

To distinguish the dots adequately, the spatial resolution setting should be between 80 and 200 dots per inch (dpi). This is because, below 80 dpi there is an insufficient amount of information and above 200 dpi the extra amount of information is superfluous. Although 16 grey levels (4 bits) yield acceptable results and minimize the memory space required, the use of 256 (8 bits) is recommended for better results [21] [36].

3.3 Binarization/Image Thresholding

In image processing, Binarization is the process of identifying the foreground (black color), which would be the content of the scanned image, from the background (white color). However, the identification of

Braille dots from the differences in the intensity of reflected light is not straightforward. There are various methods that can be used to binarize digitized documents. Among the great deal of techniques thresholding is the simplest and most commonly used approach that applies some value for the purpose. The techniques can be global or local thresholding. Meanwhile there is one outstanding problem, i.e. how to devise an automatic procedure for determining the optimum thresholding level.

In Global thresholding a single threshold value is calculated from the total intensity level of the pixel that forms the image. The resulting threshold value is then applied to the whole image pixel. Accordingly, a given pixel, based on its intensity level compared with the threshold, to set its value [17].

In local thresholding, the threshold value is calculated by considering the intensity of the image pixel in a given area, so we may have different threshold value for different area in the image. The locally calculated threshold value is used to determine the class of that pixel in the local area such as: black or white. However, local thresholding is computationally expensive as it accesses each pixels many times. And it is not recommended in many cases [17].

The global- thresholding algorithm adopted in this study is given in *Figure 3.1*. The algorithm adopted involves the conversion of color pixel to gray level to reduce computation expense.

Figure 3.1 Global-thresholding algorithm for Braille image

```

//Image thresholding algorithm
Let Img is a matrix with same size with Braille bmp image and
Let x and y be horizontal and vertical coordinate independently
Set x=y=0; // set coordinate to first pixel of image
Do
    For each line in Img do
        Read each pixel at Img(x, y) and Convert in to gray-level
        If gray level is less than or greater than the threshold then
            Set pixel at Img(x, y) as black//content or foreground
        Else
            Set pixel at Img(x, y) as white//background
    While end of Img

```

3.4 Image Segmentation

Segmentation refers to the process of separating dots from Braille image that can further be grouped in to 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.

There is different segmentation technique particularly apply in Braille image such as mesh-grids based or local measure. In local measure, the approach takes advantages of the difference in dots pixel intensity level. The dots produce different color level at the top and bottom. So the dots can be extracted by performing analysis of pixels forming the dot with threshold. However, this is usually is not preferred as it lead to extraction of noise as valid dot. The second technique, mesh-grids, is appropriate and widely applicable in Braille. The technique is preferred since Braille cells arranged strictly following vertical and horizontal layout of the document 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 [23][28][32].

In this regard, different techniques have been devised to solve clustering problem domain in data mining. Among, the existing clustering algorithms the interest is on grid-based methods. Grid is a pattern

of straight lines crossing each other which quantizes the space into a finite number of cells. Grid Mesh can be global or adaptive. Grid-based methods have the fastest processing time that typically depends on the size of the grid instead of the data objects. These methods use a single uniform grid mesh to partition the entire problem domain into cells and the data objects located within a cell are represented by the cell using a set of statistical attributes [27].

Adaptive Mesh, on the other hand, is a type of multi-scale algorithm that achieves high resolution in localized regions of dynamic, multidimensional numerical simulations. Its adaptability allows simulating multi-scale resolutions that are out of reach with methods using a global uniform fine grid.

Thus, 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. This process is important 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. Global mesh-technique is the preferred approach, if the size of the dots in the image is uniform. However, it has problem if dots size differ from document to document as it apply the same global threshold value to draw all grids line in the image.

In adaptive mesh-grids technique, to draw the horizontal grids the algorithm finds the maximum height of dots from line segmentation. To perform line segmentation, the algorithm scan each raster line of the image horizontally, if black pixel is found, it marks as the beginning of the dots and increment the dot

count by one. And then jump to the next raster line to check the presence of black pixel. This continues until no more black pixel is found in the next raster line of the image. Thus, the first black pixel and the first white pixel after the black pixel mark the beginning and the end of that particular dot. Once the dot height is determined for a given dot line, the midpoint is calculated by dividing the dot count value by two. Then, the algorithm sets pixels in raster line that pass through the midpoint coordinate, to some color value to draw the horizontal grids line.

To draw the vertical grids the algorithm scan vertically to find the maximum dot width from half-character segmentation. The resulting value which indicates the width of dot is divided by two to draw the vertical grid at the current height position plus half of the dot width value for that specific vertical line. This process continues adaptively for the whole image calculating the new dot width to draw the subsequent grids. The advantage of adaptive mesh is that it considers locally the size of dots in drawing each grids line.

Figure 3.2 and *Figure 3.3* describe algorithm for horizontal and vertical grids.

```

//Adaptive grids-mesh to construct Horizontal grids line
Let img is matrix same size with Braille image,
Let x and y are coordinates points for width and height of the image.
Do
    For each y where y is less than img height
        If pixel at img(x, y) is black
            Do while pixel at img(x, y) is black
                Increment dotcount by one
            Go to next row
        End loop
    Else
        Jump to next line
    End if
End for
If dotcount less than min-threshold
    Continue to next column
If dotcount greater than min-threshold and less than max-threshold
    Let w=0
    For each w where w is less than img width
        Draw horizontal grid-line at y-(dotcount/2)
Else If dotcount greater than max-threshold
    Let w=0
    For each w where w is less than img width
        Draw horizontal grid-line at y-(3/4)*dotcount
        Draw horizontal grid-line at y-(1/4)*dotcount
End loop

```

Figure 3.2 Horizontal Mesh-grids algorithm

```

//segmentation algorithm based on adaptive mesh-grids
Let img is matrix same size with Braille image
Let x and y are coordinates points for width and height of the image.
//construct vertical grids line
Do
  For each x where x is less than img width
    If pixel at img(x, y) is black
      Do while pixel at img(x, y) is black
        Increment dotcount by one
        Go to next column
      End loop
    Else
      Jump to next line
    End if
  End for
  If dotcount less than min-threshold
    Continue to next line
  If dotcount greater than min-threshold and less than max-threshold
    Let h=0
    For each h where h is less than img height
      Draw vertical grid-line at x-(dotcount/2)

  Else If dotcount greater than max-threshold
    Let h=0
    For each h where h is less than img height
      Draw vertical grid-line at x-(3/4)*dotcount
      Draw vertical grid-line at x-(1/4)*dotcount
  End loop

```

Figure 3.3 Vertical Mesh-grids algorithm

The algorithm is implemented to tolerate noise in case the two consecutive dots do not have space to delimit the end of the first dot and the beginning of the next dot. Thus, if the size of dot is above the threshold set to single dot size two grids line is drawn, at current height value minus $\frac{3}{4}$ and $\frac{1}{4}$ of dot count for the first and the second grids line respectively.

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. The algorithm perform search for the foreground pixel with threshold. As the segmentation is based on mesh, search to extract dots are performed around the intersection of the horizontal and vertical gridline.

The segmentation algorithm starts by determining the first intersection point of the mesh on the Braille image. Then the algorithm search pixel color of the foreground (content) with threshold in all direction around the intersection point to declare the presence of valid dot which would be the possible location of the dot.

The threshold can be relaxed to consider the minimum dot size. Accordingly, if the pixel meet the threshold, the (x, y) coordinate values of that point is registered in an array prepared for such purpose. This continues for the whole mesh intersection points and the corresponding (x, y) coordinates is registered if there is a dot on that coordinate.

Finally, the algorithm convert the whole image in to black color and set the dot coordinates value with one pixel of white color.

The segmentation technique which depends on the constructed grids mesh is robust and tolerates noise. This is because the algorithm does not require checking every pixel in the image. Moreover it tolerates noise as the search is made only following the mesh coordinates. The algorithm is described in *Figure 4.20*

3.5 Feature Extraction and Recognition

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, it will suffice to identify the (relative) positions of its dots. Then, the character can be described in terms of the numbers of dot positions.

Since the positions of Braille characters in the image have already been identified, it is straightforward to check for the existence of a dot at each of the six positions of points in a Braille cell.

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 [21].

So, modification of region based approach is adopted in the current study. 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. This is because, after segmentation dots has a single pixel representation. Then, context analysis is performed to recognize the cell as part of the Braille character or stand alone as a Braille character.

At this stage feature extraction, takes advantage of the mesh. The algorithm 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. Accordingly, a cell will have three possible statuses in a Braille character. Depending on the number of cell (one or two) defined for a Braille character, the cell recognized as part of a character or as a character alone. That is once, the feature extraction is performed, which group dots in a cell, up on recognition the cell is recognized as a Braille or part of Braille character. Algorithm adopted for feature extraction and recognition is presented in appendix V.

3.6 Artificial neural network

3.6.1 Introduction

An artificial neural network (ANN) is a mathematical model or computational model based

on biological neural networks. It can also be referred to as an adaptive statistical model on an analogy with the structure of the brain. Neural network find application in modeling, patter recognitions, signal processing and control by virtue of an important property; the ability to learn from input data [24]. The following part of the section dedicated to discuss on the concept and issues on neural network.

Artificial neural networks have been studied for more than four decades since Rosenblatt first applied the single-layer perceptrons to pattern-classification learning in the late 1950s. Recently, many researchers apply neural network in the field of computer engineering and artificial intelligence. It has been used in large number of application and has proven to be effective in performing complex function in a variety of fields. These include pattern recognition, classification, vision, control system and predictions [19].

Different definitions are forwarded to neural networks. However for the purpose of this study the one used by Heikin, S, 1999, will be adopted, which views the ANN as a formalized adaptive machine:

“An artificial neural network is a massive parallel distributed processor made up of simple processing units. It has the ability to learn from experiential knowledge expressed through inter unit connection strengths, and can make such knowledge available for use.”

3.6.2 Basic structure of a neuron

3.6.2.1 The Human brain

Artificial Neural Networks were inspired by biological findings relating to the behavior of the brain as a network of units called neurons. The human brain is estimated to have around 10 billion neurons each connected on average to 10,000 other neurons. The neural model comprises the basic structure of the human brain shown in **Figure 3.4** .The tree like structure called the **Dendrite**. The neurons collect signals from other neurons through these dendrites. The neuron send out spikes of electricity activity through a long, thin stand known as **axon**, which split into thousands of branches .At the end of each branch, a structure called a **synapse** converts the activities from the axon into electrical effects that inhibits or

excites activity in the connected neuron .When a Neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it ends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapse so that the influence of one neuron on another changes.

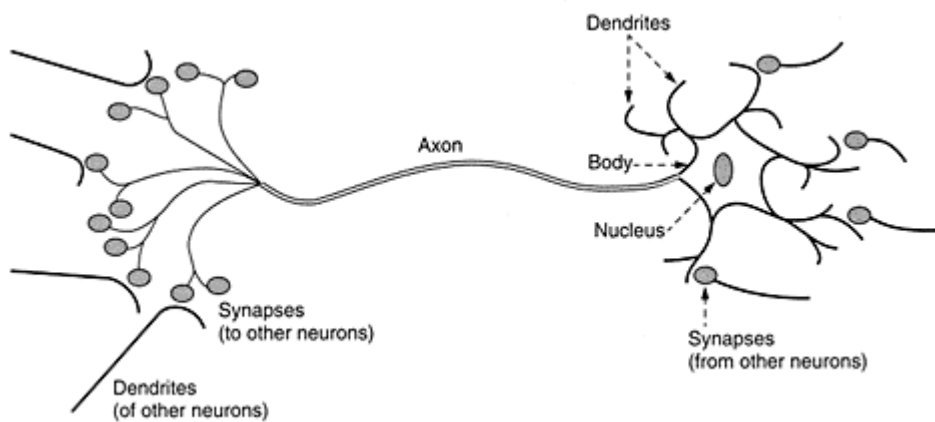


Figure 3.4 The Biological Neuron Model

The biological neuron model is the foundation of an artificial neuron. It simulates the four basic components of the natural neuron: Dendrite, **soma**, **axon**, and **synapses**.

In the neural network literature, the unit analogous to the biological neuron is referred to as a processing elements [**PE**] or units. A processing element has many input path and combines, usually by a simple summation, the values of these input paths.

When creating a functional model of the biological neuron, there are three basic components of importance. First, the synapses of the neuron are modeled as weights. The strength of the connection between an input and a neuron is noted by the value of the weight. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections [24]. The next two components model the actual activity within the neuron cell. An adder sums up all the inputs modified by their respective weights. This activity is referred to as linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and

1, or -1 and 1.

Mathematically, this process is described in **Figure 3.5**.

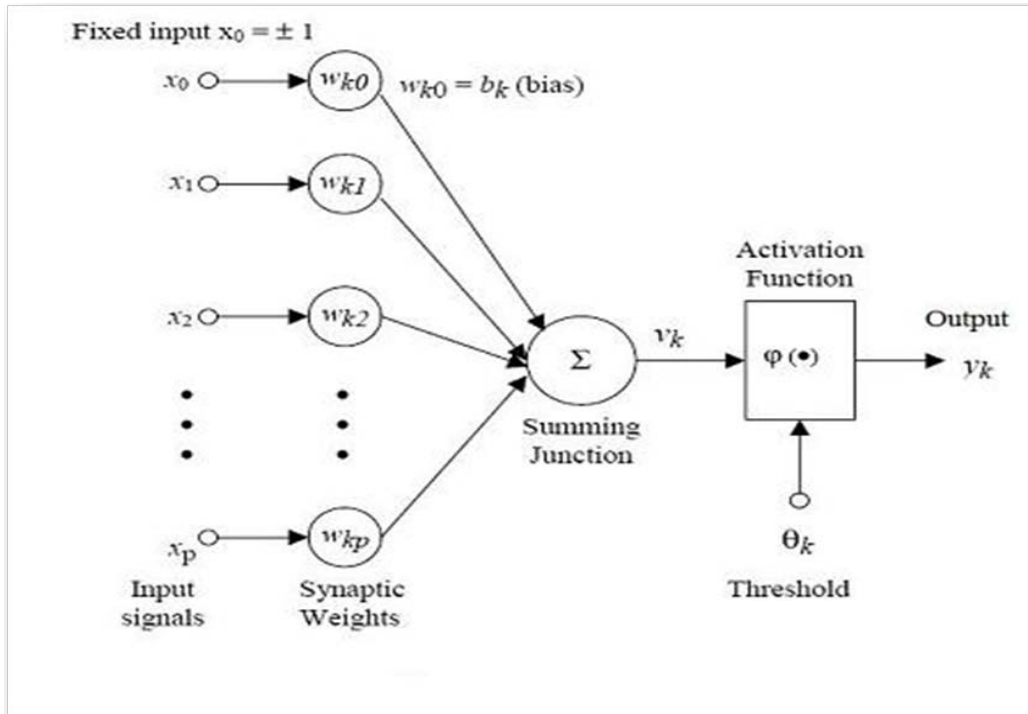


Figure 3.5 Artificial Neural Network Model

From this model the interval activity of the neuron can be shown in **Equation 3.1**:

$$v_k = \sum_{j=1}^p w_{kj} x_j$$

Equation 3.1 Weight Summation function for Neural Network

The output of the neuron, V_k , would therefore be the outcome of some activation function on the value of V_k .

3.6.3 Component of Neural Network

There are seven major components that make up an artificial neural network such as: Weighting Factors, summation function, transfer function, scaling and limiting, output function Error function and Back-

propagated value and learning function [8]. These components are valid where the neuron is used in any layer: input, output or is in one of the hidden layers.

3.6.3.1.1 Weighting Factors

A neuron usually receives many simultaneous inputs. Each input has its own relative weight, which gives the input the impact that it needs on the processing element's summation function. Weights are adaptive coefficient within the network that determines the intensity of the input signal in the artificial neuron. They are the measures of an input's connection strength. This strength can be modified in response to various training sets and through its learning rules.

3.6.3.1.2 Summation function

The first step in processing element's operation is to compute the weighted sum of all of the inputs. Mathematically, the inputs and the corresponding weight are vectors which can be represented as $(X_1, X_2, X_3, \dots, X_m)$ and $(W_1, W_2, W_3, \dots, W_m)$. The total input signal is the dot product, or inner product of these two vectors. The result is a single number, not a multi-element vector.

The summation function can be more complex than just the simple input and weight sum of product. The input and weight coefficients can be combined in many different ways before passing onto the transfer function. The specific algorithm for combining neural input is determined by the chosen network architecture and paradigm.

3.6.3.1.3 Transfer Function

The result of the summation function, almost always the weighted sum, is transferred to a working output through an algorithmic process known as the transfer function. In transfer function, the sum total can be compared with some threshold to determine the neural input. If the sum is greater than the threshold value, the processing elements generate a signal. If the sum is less than the threshold, no signal is generated.

Both types of response are significant.

3.6.3.1.4 Scaling and Limiting

Once the transfer function has completed, the result can pass through additional processing such as scaling and limiting. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset. Limiting is the mechanism, which insures that the scaled result does not exceed an upper or lower bound. This limiting is in addition to the hard limits that the original transfer function may have performed. And such type of scaling and limiting is mainly used in topologies to test biological neuron models.

3.6.3.1.5 Output Function

Each processing element is allowed to output one signal, which it may send to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output. Normally, the output is directly equivalent to the transfer function's result. Some network topologies, however, modify the transfer result to incorporate competition among neighboring processing elements. Neurons are allowed to compete with each other, inhibiting processing elements unless they have great strength. Competition can occur at one or both of two levels. Which determines neuron that will be active, or provide an output; and help to determine which processing elements will participate in the learning or adaptation process.

3.6.3.1.6 Error function and Back-propagated value

Artificial neuron's error is calculated as the difference between the current output and the desired output, in most learning networks. This raw error is then transferred by the error function to match the particular network architecture. The artificial neuron's error is then typically propagated into learning functions of another processing element. This error term is sometimes called the current error.

The current error is typically propagated backward to a previous layer. Yet, this back propagated value

can be either the current error, the current error scaled in some manner or some desired output depending on the network type. Normally, this back-propagated value, after being scaled by the learning function, is multiplied against each of the incoming connection weight to modify them before the next learning cycle.

3.6.3.2 Learning functions

The purpose of the learning function is to modify the variable connection weights on the inputs of each processing elements according to some neural based algorithms. This process of changing the weights of the input connections to achieve some desired result could also be called adaptation functions, as well as the learning mode.

3.6.3.3 Type of Activation Function

3.6.3.3.1 *The activation function*

The activation function $\Phi (\cdot)$ performs a mathematical operation on the signal output. As mentioned above, the activation function acts as a squashing function, such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). In general, the most commonly used activation functions are three, which are described below.

First, there is the Threshold Function, also known as Heaviside function, which takes on a value of 0 if the summed input is less than a certain threshold value (v) and the value 1 if the summed input is greater than or equal to the threshold value shown in *Equation 3.2*.

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

Equation 3.2 Threshold Function

Secondly, there is the Piecewise-Linear function. This function again can take on the values of 0 or 1, but

can also take on values between that depending on the amplification factor in a certain region of linear operation given in Equation 3.3.

$$\varphi(v) = \begin{cases} 1 & v \geq \frac{1}{2} \\ v & -\frac{1}{2} > v > \frac{1}{2} \\ 0 & v \leq -\frac{1}{2} \end{cases}$$

Equation 3.3 Piecewise Linear Function

Thirdly, there is the sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function. The sigmoid function is strictly continuous function, which can be differentiable. This property makes it appropriate for use in neural network trained by back-propagation shown in Equation 3.4

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

Equation 3.4 The Sigmoid Function

Some more types of activation function are also used in different areas of application of ANN .The choice of the activation function mainly depends on the distribution of the target value, which is generated by the output units. For continuous valued targeted with a bound range, the sigmoid function are preferable, provided that either the output or the target to be scaled to the range of the output activation function [37].

3.6.3.4 The network Architecture

Artificial neural networks have simple structures and are designed to mimic the function of the biological neurons. The principal importance of a neural network lies on their interconnections. The natural connection of a human brain is too complex to be implemented directly in neural network. The structures, therefore studied so far are simple and easy to be implemented in digital computers.

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. It is defined by a number of layers, the number of units per layers, and the interconnection pattern between layers. In general, ANN has a similar structure of topology, basically, these are three types of layers each consisting of a group of neurons.

- The input layers consist of neurons that receive input from the external environment
- The output layers consist of neurons that communicate the output of the system to the user or external environment
- One or more hidden layers between the output and input layers

When the input layer receives the input, its neurons produce output; this becomes input to the next layer of the system. The manner in which the neural network is structured is internally linked with the learning algorithm used to train the network [24].

3.6.3.4.1 Single layer feed forward networks

A feed forward network is a network where the neurons on the first layer send their output to the neurons on the second layer, but they do not receive an input back from the neurons on the second layer.

In other works the network is strictly feed forward. A network with the input and output layer only is called single-layered network.

3.6.3.4.2 Multi layer feed forward networks

The multilayer feed forward networks or the multilayer perceptron, are generalization of the single layer perceptron. It includes one or more hidden layers each with their respective number of neurons as shown in Figure 3.5. The function of the hidden neurons is to intervene between the external input and the network output in some useful manner.

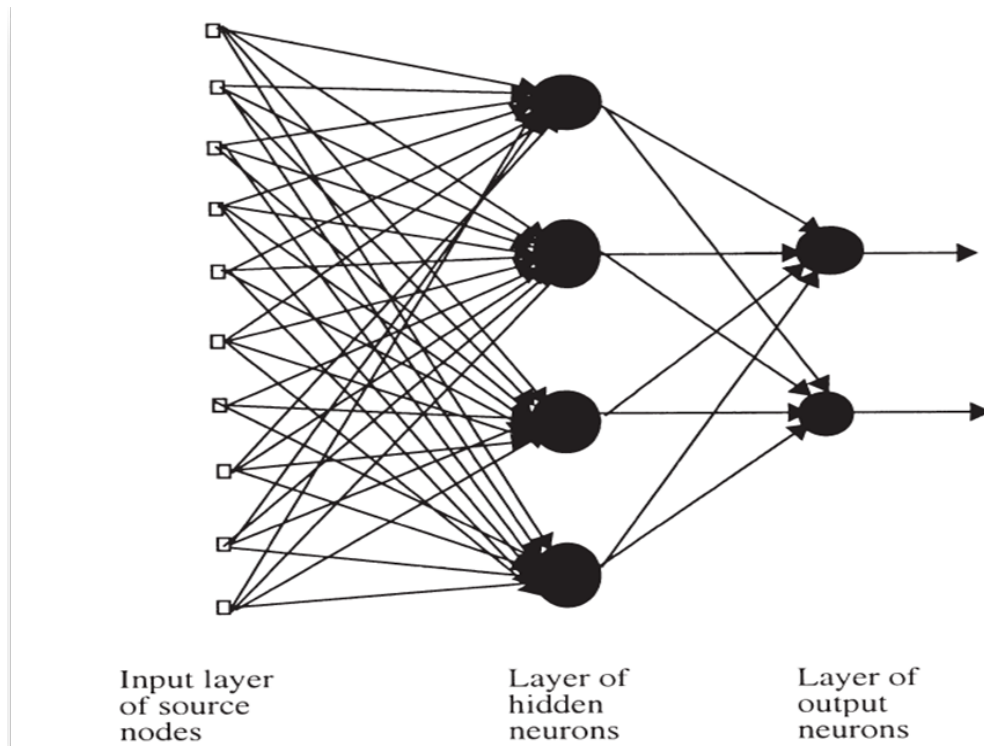


Figure 3.6 Feed Forward Neural Network

3.6.3.4.3 Recurrent networks

A Recurrent networks, which is sometimes referred to as feedback networks can have a signal traveling in both direction by introducing loops in the network. There may, exist one or more such loops as shown in **Figure 3.7**. In these networks a neuron of layer after receiving input from another layer, send its output back to the previous layer neurons. Recurrent network are said to be dynamic in that their state is changing continuously until they reach an equilibrium point [34].

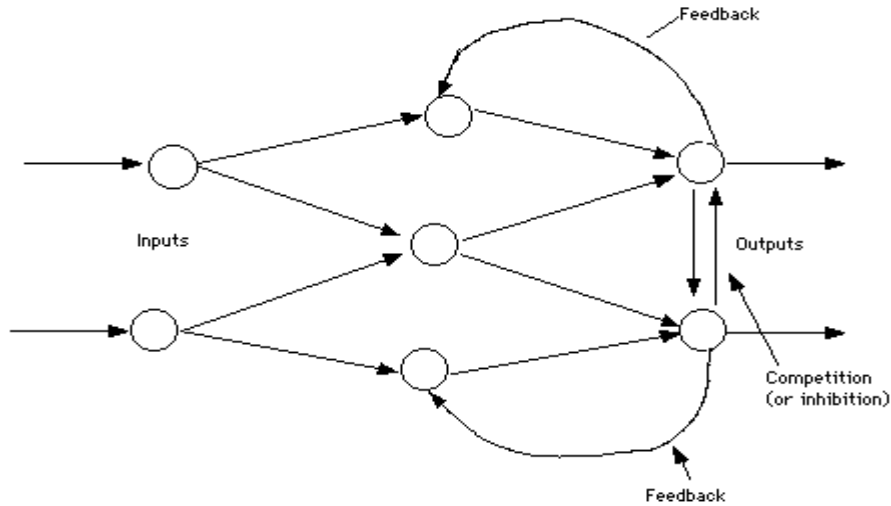


Figure 3.7 Simple Networks with Feedback and Competition.

3.6.3.5 The learning process

The process of adjusting the weight to make the network learn the relationship between the input and target is called learning. The connection between neurons in a neural network is given adjustable weights or measure of importance. To be more precise a definition of learning that is adapted from Mendal and McClaren [1970] is given by [24] as follows:

“Learning is a process by which the free parameters of a neural network are adopted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameters changes take place.”

The process of making the network to learn the solution to a problem follows a set of well defined rules called the learning algorithms. Many learning algorithms have been invented to help find an optimum set of weight that result in the solution of the problem. In general, all the learning methods of an adaptive network can be classified into two major categories.

3.6.3.5.1 Supervised learning

In Supervised learning the neural networks are trained to produce desired outputs in response to sample inputs, making them particularly well suited to modeling and controlling dynamic systems, classifying noisy data, and predicting future events. Most commonly, neural network use supervised training. These input-output pairs are provided by an external teacher or by the system containing the network. For this reason it is also called learning with a teacher. The actual output of the net may or may not match the desired output, depending on the weight at the particular moment. In all cases the training algorithms modifies the weights with the idea that the next time the net sees the same input pattern, it will more closely reproduce the target output pattern as its actual output.

3.6.3.5.2 Unsupervised learning

In Unsupervised learning the neural networks are trained by letting the network continually adjust itself to new inputs to find pattern or regularity in the input data. So the network is not trained towards specific outputs. It is self-organizing learning process that does not require external teacher. Once, the network has become tuned to the statistical regularities of the input data. It develops the ability to form internal representation for encoding feature of the input and thereby to create new classes automatically [9].

Learning rate

It is rate at which the artificial neural network learns and it depends on the several controllable factors. In selecting the approach, there are many tradeoffs to consider. Obviously, a slower rate means of a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the transfer learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly.

Generally, several factors besides time have to be considered when discussing the offline training test.

Network complexity, size, architecture, type of learning rule or rules employed and desired accuracy must be all considered. These factors play a significant role in determining how long it will take to train a network.

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

3.6.3.6 Learning laws

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law [8].

3.6.3.6.1 Hebb's Rule

The first learning rule was introduced by Donald Hebb. His basic rule was: if a neuron receives an input from another neuron and if both are highly active, the weight between the neuron should be strengthened.

3.6.3.6.2 Hopfield Law

It is similar to the Hebb's rule with the exception that it satisfy the magnitude of strengthening or weakening. It states,"if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate".

3.6.3.6.3 The Delta Rule

This rule is the further variation of the Hebb's rule. It is one of the most commonly used rules. The delta rule works in the way that the delta error in the output layer is transferred by the derivative of the transfer function and is then used in the previous neural network to adjust input connection weights. In the other words, this error is back-propagated in to the previous layers one layer at a time. The process of back-

propagating the network errors continues until the first layer is reached. This rule changes the synaptic weight in the way that minimizes the mean squared error of the network.

3.6.3.6.4 The Gradient Descent Rule

This rule is similar to the delta rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly.

3.6.3.7 Multilayer Feed forward neural networks

Multilayer feed forward neural network or multilayer perceptron network [MLP] is the most popular neural network type. It has been applied successfully to solve some difficult and diverse problems by training them in supervised or unsupervised manner with a highly popular algorithm known as *error back-propagation algorithm*.

The typical network has an input layer, at least hidden layer and an output layer. There is no limit on the number of hidden layer but most application use just one or two.

3.6.3.8 Back-propagation

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. 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. This scheme was developed independently by Webros, Parker, Rumehart, Hinton and Williams [8].

The role of a training algorithm is to set the network's weight and threshold so as to minimize predictive error by the network. The historical cases gathered are used to automatically adjust the weights and thresholds in order to minimize this error. This process is equivalent to fitting the model represented by the network to the training data available. The error of a particular configuration of the network can be determined by running all the training cases through network, comparing the actual output of generated with the desired or targeted output. The difference is combined together by an error function to give the network error. The most common error function is the sum squared error, where individual error of output on each case squared and summed together [30].

The error back-propagation method can be applied to any feed forward network with differentiable activation function. The requirement of differentiability of the activation function is based on the following reason; as mentioned before, the difference between the response of an output unit and expected response is the error made by the network. The unit of the output layer uses this error directly to correct their connection weights, but this is not the same for the units of the hidden layer since they are not in direct contact with the error. As a result the unit of hidden layers needs to estimate their error using error back-propagation method. The amount of error made by the network is first converted into an error signal that is proportional to the rate of change of the nonlinear activation function. In fact, this implies that we want the correction to be proportional to the rate of change of the activation function.

The ultimate purpose of error correction learning is to minimize the cost function based on the error signals, such that the actual response of each output neuron in the network approaches the target response

for that neuron in some statistical sense [24]. The algorithm for feed-forward network is given in **Figure 3.8**.

3.6.3.9 Application of neural networks

When the result of the model is more important than understanding how the model works, neural network are good choices for most classification and prediction tasks. Neural network actually represent complex mathematical equation, with lots of summation, exponential function and many parameters. Neural network do not work well if there are many hundreds or thousands of input feature. Large numbers of feature make it more difficult for the network to find patterns and can result in long training phases that never converge to a good solution [12].

Neural network have a broad applicability to real world business problems .They have already been successfully applied in many industries and in pattern recognition in image processing.

Input: The training samples, samples; the learning rate, l ; a multilayer feed-forward network, network.

Output: A neural network trained to classify the samples.

Method:

- 1) Initialize all weights and biases in network;
- 2) **While** terminating condition is not satisfied{
- 3) **for** each training sample X in samples{
- 4) // Propagate the inputs forward:
- 5) **for** each hidden or output layer unit j
- 6) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j
- 7) **for** each hidden or output layer unit j
- 8) $O_j = 1 / (1 + e^{-I_j})$; // compute the output of each unit j
- 9) //Back propagate the errors:
- 10) **for** each unit j in the output layer
- 11) $Err_j = O_j(1 - O_j)(T_j - O_j)$; //compute the error
- 12) **for** each unit j in the hidden layer
- 13) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$;
- 14) **for** each weight w_{ij} in network:
- 15) $\Delta w_{ij} = (l) Err_j O_i$; //weight increment
- 16) $W_{ij} = w_{ij} + \Delta w_{ij}$; //wight update
- 17) **for** each biase θ_j in network{
- 18) $\Delta \theta_j = (l) Err_j$; //bias increment
- 19) $\theta_j = \theta_j + \Delta \theta_j$; //bias update
- 20) }}

Figure 3.8 Feed-forward algorithms for neural network.

Chapter Four

4. Experimentation

4.1 Introduction

Braille is an invaluable means to visually-impaired to interact with the real world. Different attempt has been made to recognize Braille writing system in different language. In this study effort has been made to recognize Braille document for Amharic language. This chapter provides detailed performance analysis of the Amharic Braille recognizer developed in the present study.

The proposed system starts with acquisition of the Braille documents which is usually called digitization. The Digitization of Braille documents is performed with flat-bed scanner (at 200 dots per pixel resolution), which is the chipset and commonly used device for such study. After the digitization process, the image is processed to identify the foreground (content) from the background; this process is called image Binarization or thresholding. In this study, global-Thresholding technique is applied for Binarization process, since the technique is well known and commonly used approach particularly for image that has uniform color intensity such as Braille document. Threshold value has been set based on the color frequency through experimentation.

During segmentation the binarized image is further processed to locate the position of potential Braille dots. In the study segmentation algorithm is developed for identification of the dots following the gridline constructed vertically and horizontally with the dots pattern. The first step construct mesh grids considering the Braille dots, then segmentation performed by searching the black pixel with threshold around the mesh coordinate. Eventually, the procedure identifies dots in one-pixel white color with black background. The advantage of gridlines is that dots search only made on coordinates. This reduces the computation time. The resulting image is then fed to the feature extraction procedure to extract the

patterns of each Braille cell. At this stage, the mesh comes into the picture to group the dots in to cell. And context analysis is made to recognize the Braille character. In this regard, it reduces the computation time as it does not consult all the pixels of image. Besides, Braille character mainly uses one and/or two cells to represent the corresponding print character. Further analysis is made on the cell content, since in some case two Braille cells represent one print character. So to determine a cell is part of a particular character a context analysis is preferred based on rules. Eventually, the extracted patterns are created and stored with one and two cells in a file to prepare for further processing using neural network. In recognition (classification) of Braille-to-print, neural network is applied. This is because, neural network is widely applied in different problem domain as it is robust to tolerate error and learn from example. Accordingly, the feed-forward neural network is created with multilayer fully connected perceptron. The block diagram in Figure 4.1 shows the proposed system as seven steps process along with the corresponding tools applied in each steps.

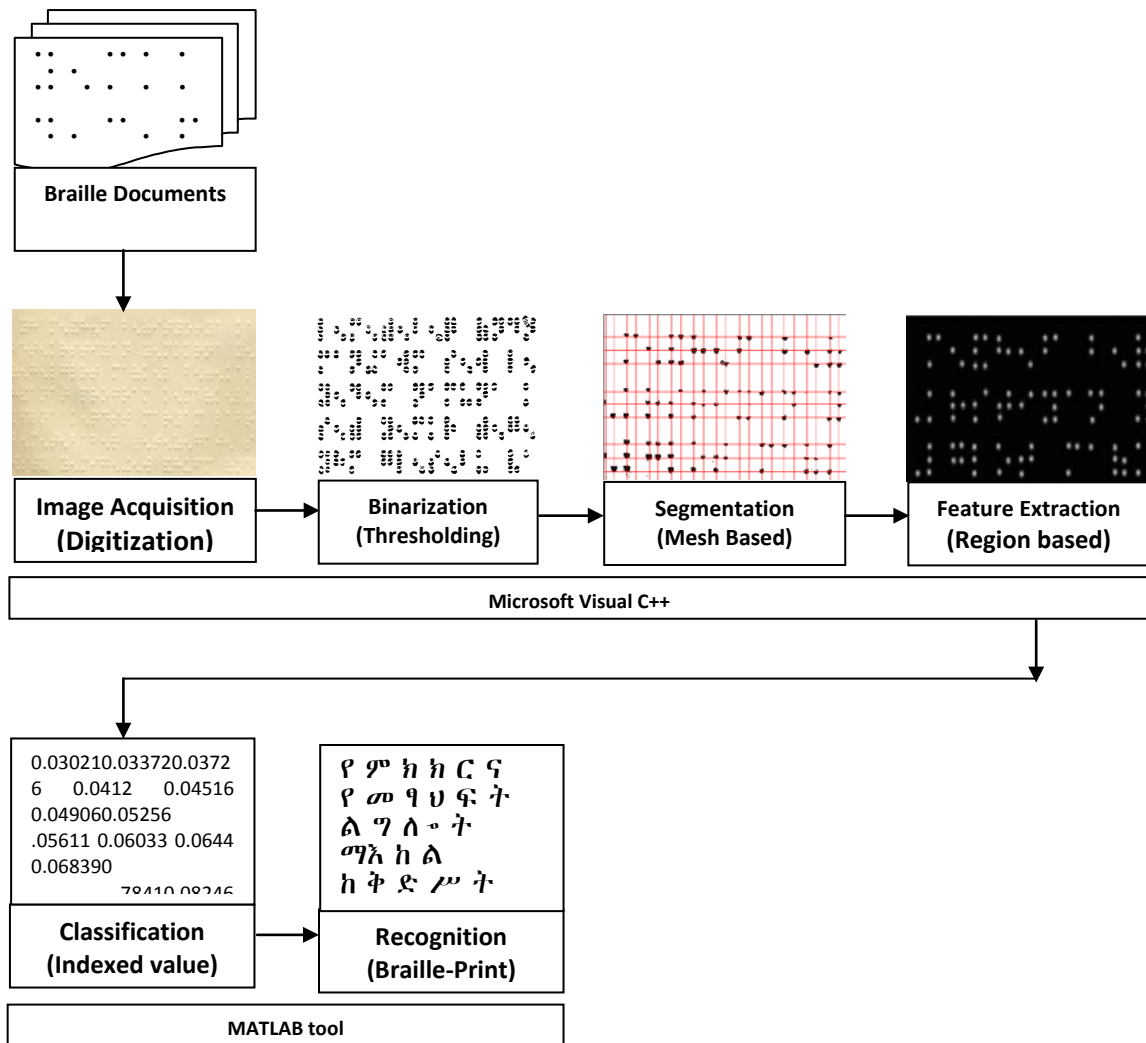


Figure 4.1 Block diagram of the Amharic Braille Recognizer.

4.2 Data Collection

Braille document has been collected mainly from German school and AAU-Kennedy library. The documents include both manually produced and typewritten. As the Braille can be plastic and/or paper sheet, both were collected, most of which is typewritten. In addition, considering color variation in the present study documents with white and light yellow has been selected. This is done because the intensity of pixel in Braille image varies from one color to another, it tremendously affect the noise level. Added to digitized documents, 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 4.1* .

Table 4.1 Summary of Amharic Braille documents collected for the study

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

4.3 Digitization/Image Acquisition

In Optical Braille Recognition (OBR), digitization enables to produce a digital image of the scanned document in the form of bitmap. A flat-bed scanner, with hp photo and imaging program, has been used for digitization process. This is because it is a cheap alternative which can be used for so many other applications of this sort.

The Braille document is scanned with 200 dots per inch (dpi) resolution which is recommended to get good quality image. With this configuration, Braille documents that are collected for the present study are scanned and prepared for subsequent processing.

The document scanned with color and gray level, which is examined visually to get better image.

Sample digitized image is shown in *Figure 4.2* (left for gray and right for color).

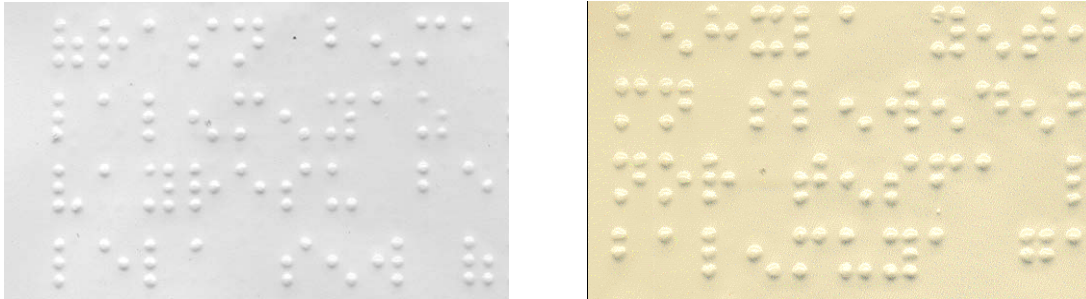


Figure 4.2 Braille document image (left with gray level scale and right with color)

4.4 Binarization/Image Thresholding

In image processing Binarization is the process of identifying the foreground (black color), which would be the content of the scanned image, from the background (white color). Binarization has been performed by thresholding approach, which can be global or local. For the problem at hand, a global thresholding technique has been selected for the reason that it is simple to use and computationally less expensive than local thresholding technique. Moreover, the Braille image has uniform color which is appropriate for global thresholding.

In this research, initial attempt has been made to develop a program coded in visual C++ studio. And the algorithm produces satisfactory result with typical Braille image. To set the threshold value, image pixel-level frequency has been examined for both gray-level and color image. *Table 4.2* shows the frequency of a typical Braille image pixel intensity, for colored and gray-level image along with the corresponding histogram graph given in *Figure 4.3*.

Table 4.2 Frequency distribution of colors (left for gray-level image and right for color image)

Gray-level	Frequency	Color-level	Frequency
92	1	101	1

124	1	118	27
132	13	135	1023
140	68	152	5470
148	628	169	2892
156	2101	173	3304
164	5954	180	14767
172	12200	191	1727
180	19702	194	6223
188	29625	197	29385
196	42194	202	32767
204	61493	203	21542
212	154388	207	15762
220	1143817	213	139651
228	1729945	220	19513
236	379544	224	64455
244	101286	228	442743
252	92507	232	651437
254	1839	236	505634
Total color	19	246	121213
		253	84585
		254	76546
		Total color	22

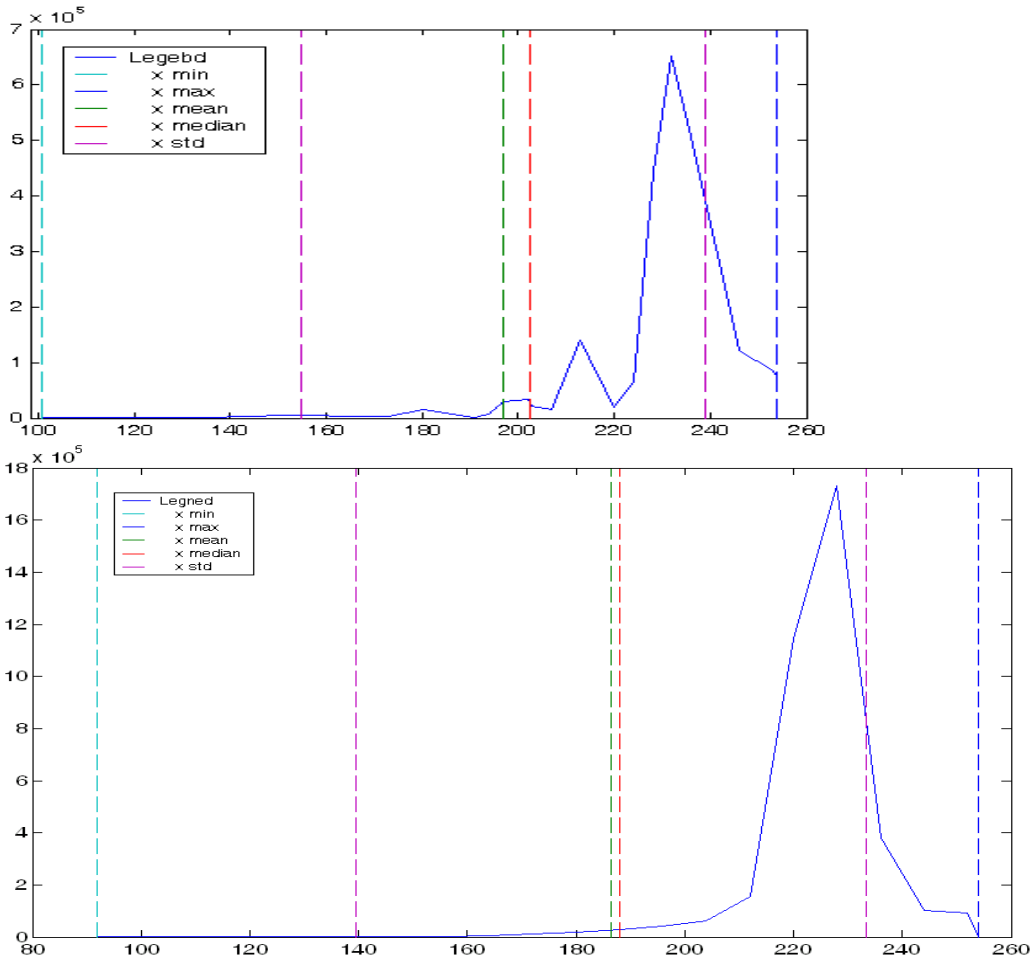


Figure 4.3 Frequency distributions of colors for Braille image (Top- color Braille image, and Bottom-Gray level image)

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 [17].

Accordingly, based on the frequency distribution different experiments have been conducted with three different thresholds. The thresholds that has been considered include value less than, less than or greater than, and greater than the maximum frequency for both type of image. And from experience a better image result is found from the gray level image. It is also observed from experiment that the noise level increase with color image than for same image in gray level. The resulting images based on the three thresholds are given in the following part.

4.4.1 Binarizing Gray Image

To binarize gray image, one-threshold value and two-threshold value are tested.

Image with two-threshold value: From experience for gray image the best image (see *Figure 4.4*) with less noise is resulted with threshold value: less than or equal to 202 or greater than or equal to 247 color values.

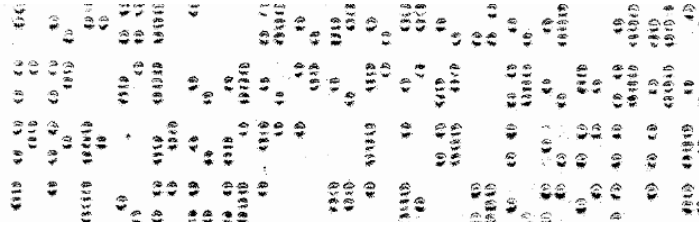


Figure 4.4 Gray-level image after threshold (color value ≤ 202 or color value ≥ 247)

Image with one-threshold value: sample gray-image result based on less than (see *Figure 4.5* left-image) and greater than (see *Figure 4.5* right-image) from the maximum frequency:

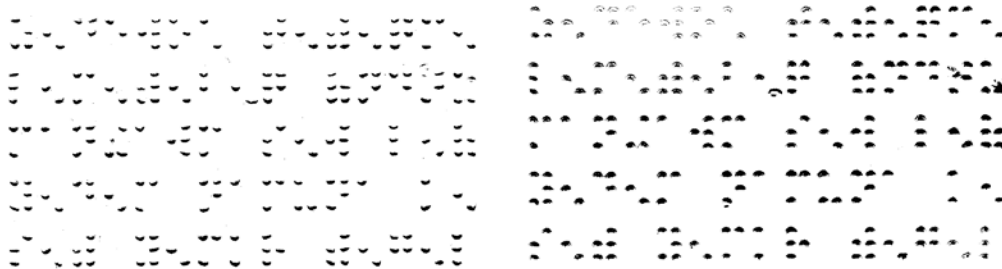


Figure 4.5 Binarized gray-level image (left- Color value less than 202, and right- with Color value greater than 247)

The resulting images from the above process work well for further processing. However, with two threshold values the resulting image produce a single- dot (two black areas) separated by white color in between. Sample image is shown in *Figure 4.6*.



Figure 4.6 Result of Binarization with two threshold value.

Image shown in *Figure 4.6* is resulted because the raised part of the Braille dot produce two different intensity at the top and bottom part of a dot (protrusions) for light reflected while digitizing the document. This image is important to separate double-sided Braille document in to two: front and back side. This study does not consider this image further, since the scope of the study is focused on single-sided Braille.

4.4.2 Binarizing Color image

An attempt is also made to binarize color image as described below.

Image with two-threshold value: *Figure 4.7* given below show sample result from color image. Color Image Binarization has been performed with threshold value less than or equal to 203 or greater than or equal to 254 color value.

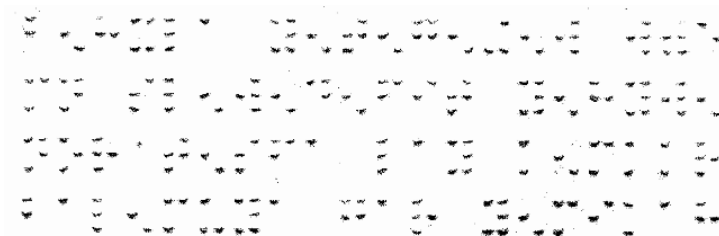


Figure 4.7 Color-level images after threshold

Visual examination of the resulting image show that with color image there is a possibility of losing valid dot point as the dots size is relatively small and the noise level is relatively high. This requires advanced image preprocessing techniques. Hence, in this study the one with gray-level image is considered for further processing.

The program which is coded in visual C++ environment performs the Binarization with threshold value. The algorithm mainly performs two things: first converts each pixel in to gray-level, and then compare the resulting color value with the threshold to set the pixel in one of the two colors (black and white).

```

RGBQUAD color; //store RGB value
COLORREF ColVal; //store graycolor value
        color.rgbRed=pDC->GetPixel(b,a);
        color.rgbGreen=pDC->GetPixel(b,a);
        color.rgbBlue=pDC->GetPixel(b,a);
//convert color value to graylevel

ColVal=(color.rgbRed * 77 + color.rgbGreen * 150 + color.rgbBlue * 29 + 128)/256;

```

Figure 4.8 Ms-visual C++ code to extract the RGB of image pixel and convert to gray-level

To accomplish this, color variable is declared as RGBQUAD, which allow extracting the RGB (red, green and blue) value of a pixel in visual C++. Then the Red, Green, and Blue value of the pixel are stored independently in a color variable declared as COLORREF as given in *Figure 4.8* along with the formula for computing colval (color value).

The global- threshold algorithm perform well on a clean Braille image. However, most of the Braille is degraded in quality, and the noise level affect to the extent of missing some valid dot and/or separating overlapping dots.

The following are some of the challenge worth mentioning while performing the Binarization experimentation which increases noise level.

- First as the Braille document has uniform color it makes difficult to find a value that particularly locate the area of the dot (see *Figure 4.9* (b)).
- Because of repeated use and bend, image is distorted. As a result the binarized image is suffered in scattered type of noisy. This condition is clearly visualized in *Figure 4.9* (a).
- Preparation of image, layout, arrangement and reflection of light affect the digitization

process too (see *Figure 4.9 (c)*). To get clear image decreasing/increasing the threshold will diminish the size of the dots.

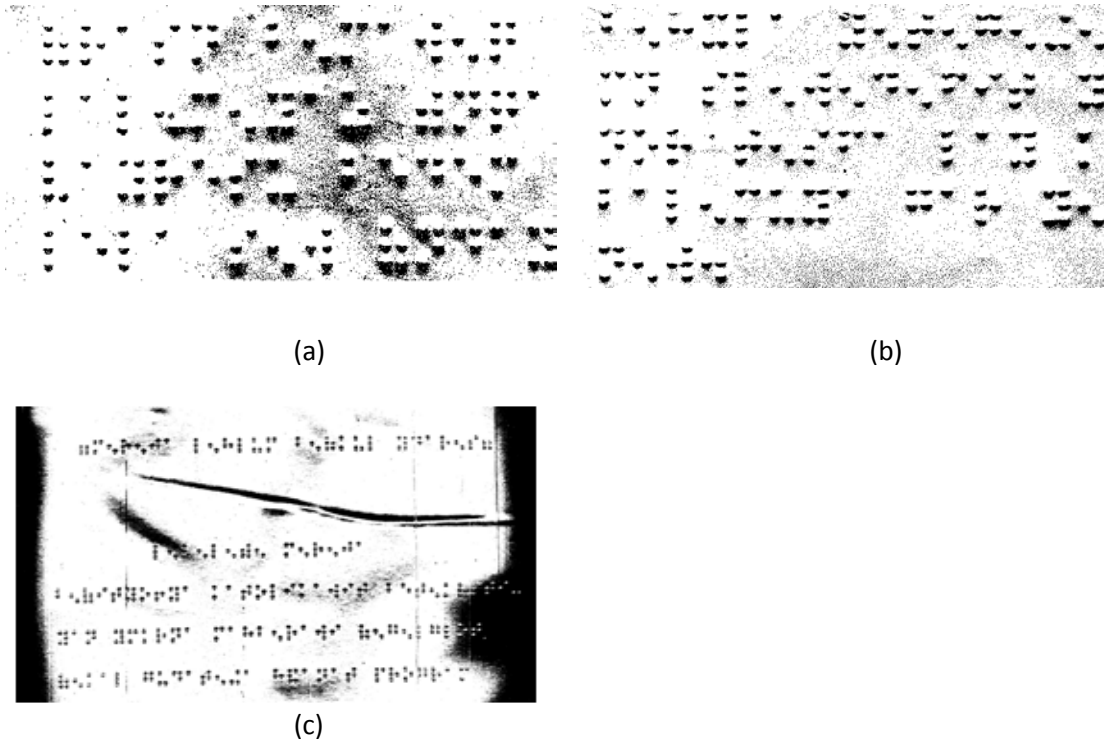


Figure 4.9 Noisy image after thresholding ((a) repeatedly used Braille image, (b) uniform color and (c) light effect).

In such a case attempt was made to make some adjustment on the threshold (by decreasing or increasing). The action would bring some improvement in decreasing the scattered noise, but still this will happen at the cost of losing some valid dots as shown in *Figure 4.10*.

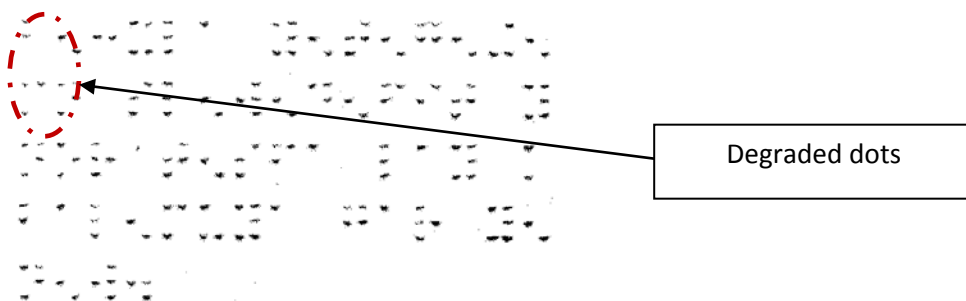


Figure 4.10 Result of Binarized image after reducing the threshold (Image with reduced threshold)

At this point, a second attempt is made to binarize the image using image manipulation program. This is

done with some image in which high level of noise was resulted. Though there are different image processing programs the researcher prefers GIMP2.6 tool because it is found to work well for the intended purpose and available free. The resulting image is saved with two tone color value. *Figure 4.11* shows the final binarized Braille image.

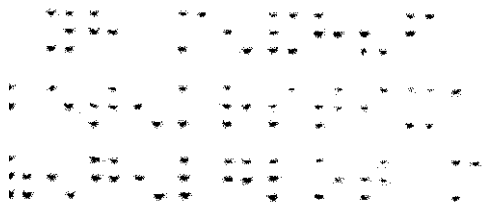


Figure 4.11 Sample binarized which is used for further recognition.

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. This output of the Binarization module will be fed to the next level of processing, which is image segmentation.

4.5 Image Segmentation

As indicated previously, segmentation refers to the process of separating dots from Braille image that can further be grouped in to a cell. These cells further grouped to form character, words or any strokes in Braille. 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 as depicted in *Figure 4.12*.

This could be:

- 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

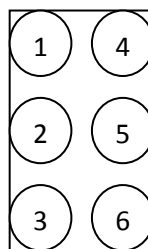


Figure 4.12 Braille cell with dots position

In this research segmentation is performed in two steps: In the first step gridlines mesh is constructed; and then dots are searched following the grids mesh with threshold. 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 [23][28][32].

In this study, both global and adaptive, mesh techniques has been tested to select a method with best result. In using global mesh, 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 illustrate, after the Binarization process is over, horizontal and vertical projection profile of the image has been performed. This process is used to determine the height of a single dot, which would be used to set the threshold. Based on analysis of horizontal projection profile, a threshold value of ten (10) has worked well to isolate Braille dot lines (dot height). After line segmentation, vertical projection is performed that result in half-character segmentation (one column of a cell). With vertical projection, a threshold of 12 works well to isolate dot (dot width). *Figure 4.13* describes the result of vertical and horizontal projection.

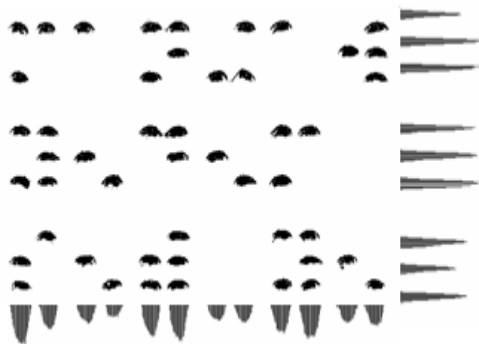


Figure 4.13 Horizontal and vertical projection for Braille image.

Based on the above information (spacing, dot height and width), a mesh is constructed. At this moment initial attempt has been made to test the global mesh that performs in the following ways.

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. So, each vertical grids line is constructed leaving 10 pixels for inter dots (within a cell) spaces and 14 pixels for intra dots (between cells) spaces. After vertical grids, the horizontal grids line is constructed with 10 pixel inter dots spacing (with in a cell) and 10 pixel intra dots spacing as shown in *Figure 4.17*. Inter and intra dot spacing is described in *Figure 4.14*.

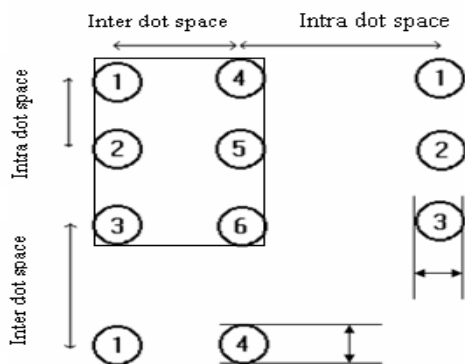


Figure 4.14 Inter and intra dot spacing in Braille

Once the optimal starting dot for vertical grids is determined, the subsequent possible vertical grids continue for the width of the Braille image with threshold value. *Figure 4.15* shows the vertical grids line drawn for a given Braille image.

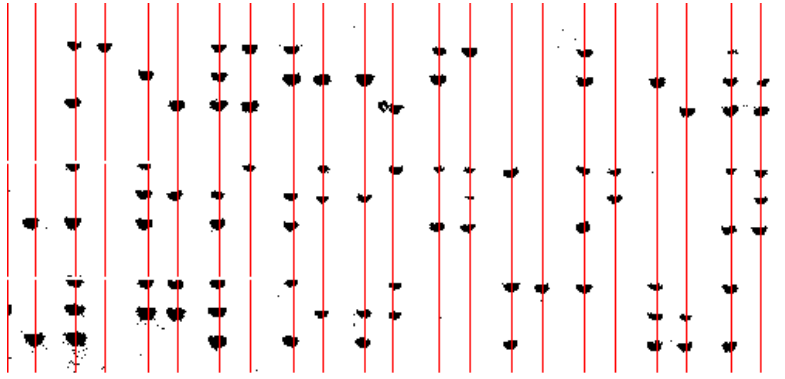


Figure 4.15 Vertical grids line based on threshold value (10 pixels for dot within a cell and 14 pixels for dots between cells).

On the other hand, the horizontal histogram is drawn to determine the height of dots, distance between dots of same character and next dot line. Through experimentation the value is set to 15, 10, and 18 for dot height, space between dot with in a cell and space between dots of different cells, respectively. A sample Braille image with horizontal line is given in *Figure 4.16* below.

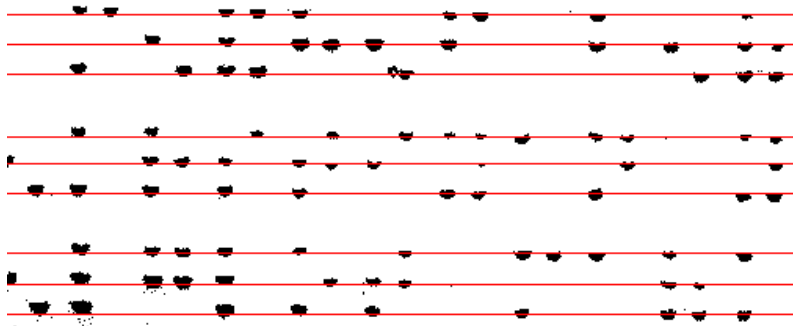


Figure 4.16 Horizontal grids line based on threshold value (10 pixels for dot within a cell and 18 pixels for dots between cells).

The algorithm constructs the mesh grid well. A typical Braille image with global grids is shown in *Figure 4.17*.

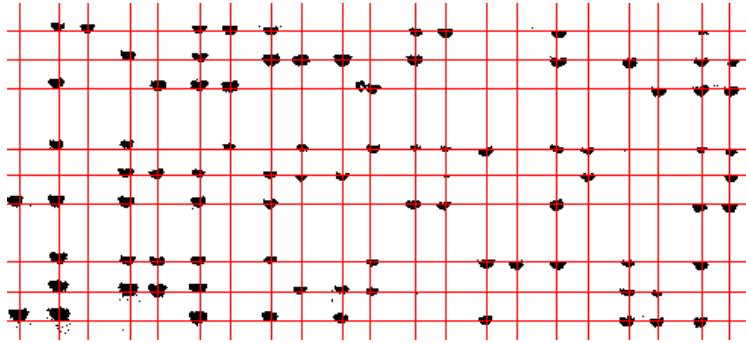


Figure 4.17 Braille image with Global grids

However, experience with different images show that dots on the Braille image do not always have the same size (height and width). Some documents show well formed dots while others consist of very small dots. Moreover the dots collection in a given Braille image does not have uniform size. While constructing mesh for such image, experimentation show that the grids line deviation increase as it moves to the right horizontally and to the bottom vertically. *Figure 4.18* is a typical example in such condition.

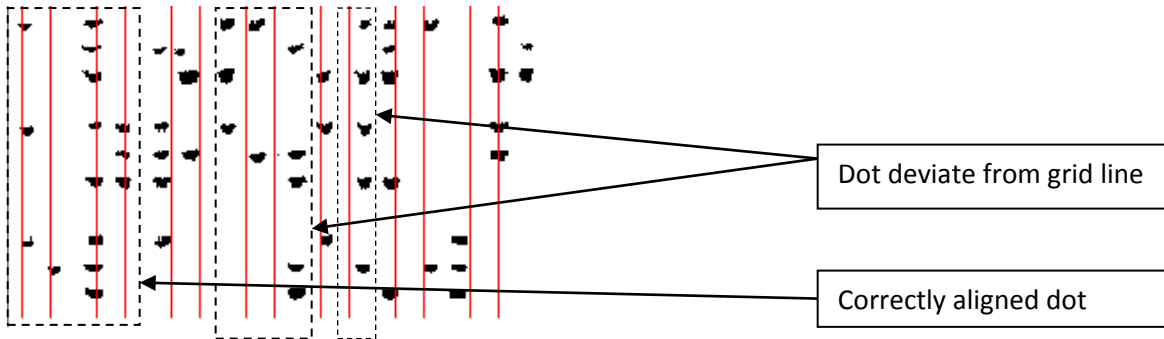


Figure 4.18 Deviation of grids line towards the right ends

At this point adaptive algorithm is developed to construct the grids line with the intension to come up with better result. This time, the algorithm calculate dots size and spacing locally (adaptively) to draw the subsequent vertical and horizontal grids.

The algorithm for grids formation is described in section 3.4. To draw the horizontal grids the algorithm finds the maximum height of dots from line segmentation. To perform line segmentation, the algorithm scan each raster line of the image horizontally, if black pixel is found, it marks as the beginning of the dots and continue until the next white pixel is found. Once the dot height is determined for a given dot

line, the midpoint is calculated. Then, red line is drawn that pass through the midpoint coordinate.

To consider overlapping dots, a threshold value of 35 pixel dot size has been set experimentally. Thus, in this situation the dot count is checked if it is more than the size of the threshold. If the result is true, accordingly two grids line is drawn, at current height value minus $\frac{3}{4}$ and $\frac{1}{4}$ of dot count for the first and the second grids line respectively. Overlapping dots are shown in *Figure 4.19*.

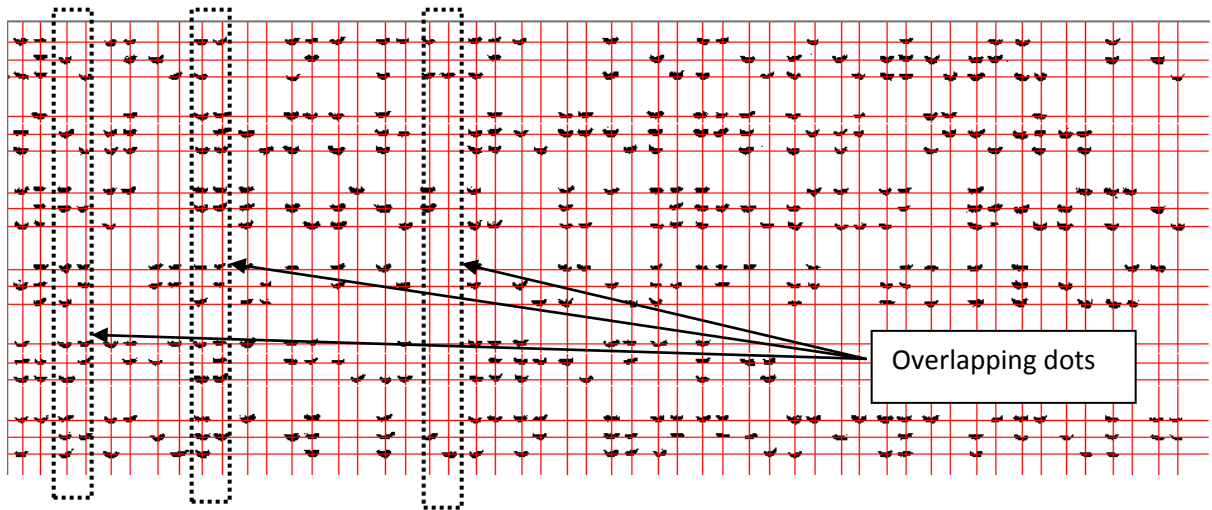


Figure 4.19 A scenario that shows grids for overlapping dots considering the threshold dot width

To draw the vertical grids the algorithm scan vertically to find the maximum dot width from half-character segmentation. The resulting value which indicates the width of dot is divided by two to draw the vertical grid at the current height position plus half of the dot width value for that specific vertical line. This process continues adaptively for the whole image calculating the new dot width to draw the subsequent grids. Part of mesh implementation for horizontal grids is given in *Figure 4.20*

```

bool hit=FALSE;int dotWidth=0,h=0,DotSpace=0,b=0
for(int a=0;a<bm.bmWidth;a++)
{
    //count max dot height for dot line
    for( b=0;b<bm.bmHeight;b++)
    {
        if(pDC->GetPixel(a,b)==0)//check black pixel
        {
            dotWidth++;hit=TRUE;break;
        }
    }
    if(hit)
    {
        hit=FALSE;continue;
    }
    else if(!hit && dotWidth<=5 && dotWidth>0)//eliminate noise
    {
        for(int x=a-dotWidth;x<a;x++)
            for(int c=0;c<bm.bmHeight;c++)
                dc.SetPixel(c,x,RGB(255,255,255));
        dotWidth=0;
    }
    else if(!hit && (dotWidth>5)&& dotWidth<=20)
    {int d=a+1, thr=a-(dotWidth/2);
    for(int c=0;c<bm.bmHeight;c++)
    {
        dc.SetPixel(thr,c,255);//set pixel to red color
    }dotWidth=0;
    bool b_dot=TRUE;DotSpace=0;
    //check space between two dots in a line,
    while((dc.GetPixel(d,0)==RGB(255,255,255))|| dc.GetPixel(d,0)==255)&& b_dot)
    {DotSpace++;
        for(int z=1;z<bm.bmHeight;z++)
            if(dc.GetPixel(d,z)!=RGB(255,255,255)&& dc.GetPixel(d,z)!=255)
            {
                b_dot=FALSE;break;
            }d++;}
    if(DotSpace>=40)//if space b/n two dots >=40 draw additional line
    {
        int thr=a+12;
        for(int c=0;c<bm.bmHeight;c++)
            {dc.SetPixel(thr,c,255);} DotSpace=0;
    }
    else if(!hit && dotWidth>30)
    {int thr=a-dotWidth, thr_1=(a - dotWidth*(3/4)), thr_2=(a - dotWidth/4);
    for(int c=0;c<bm.bmHeight;c++)
    {
        dc.SetPixel(thr_1,c,255);
        dc.SetPixel(thr_2,c,255);
    }dotWidth=0;}
}
}

```

Figure 4.20 Horizontal mesh implementation

As the adaptive mesh take in to consideration the maximum dot height and/or dot width for horizontal and vertical grids line respectively, none of the dots will bring significant deviation from the line. Even the small dots fall adjacent either above or below the grids line. Sample image taken during the experimentation is given in *Figure 4.21*.

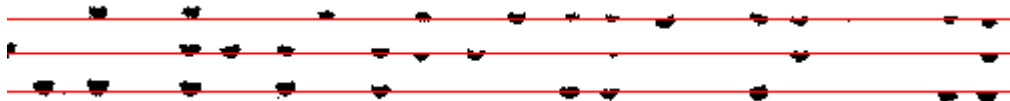


Figure 4.21 Horizontal gridline for Braille image

At this stage it is worth to mention some problem investigated during experimentation of adaptive mesh in different Braille image. The algorithm is developed on the assumption that there is a minimum of one dot down the image vertically. And the algorithm draws vertical line only for the presence of valid dots to that specific half-character. However, experience shows that some Braille image consists of one or two line of character. Accordingly, there is a probability of missing dot in any of half character down the column. This condition is undesirable as a Braille character should have 2 columns by 3 rows structure. The problem also affects the feature extraction, which would be the forth coming phase in the study. So, further modification is made to mesh program to draw ideal grid lines where valid dots are missed vertically in half-character. To correct the problem one module (see *Figure 4.22*) is added to check if the distance between the two dots in a line is within the range of the threshold for dot space within a character. If the spaces fail to meet the criteria, the algorithm draw a hypothetical vertical line considering the dot distance which is set experimentally to be 12 pixels from the first dot. To draw the line the algorithm check the white space between two dots is greater than 40 pixels (which is set experimentally).

```

if(DotSpace>=40){int thr=a+12;
    for(int c=0;c<bm.bmHeight;c++)
        dc.SetPixel(thr,c,255);
        DotSpace=0;
    }

```

This is importa

three rows. Braille image

Figure 4.22 Visual C++ code to draw hypothetical vertical grids line

with ideal vertical grid is given in **Figure 4.23**.

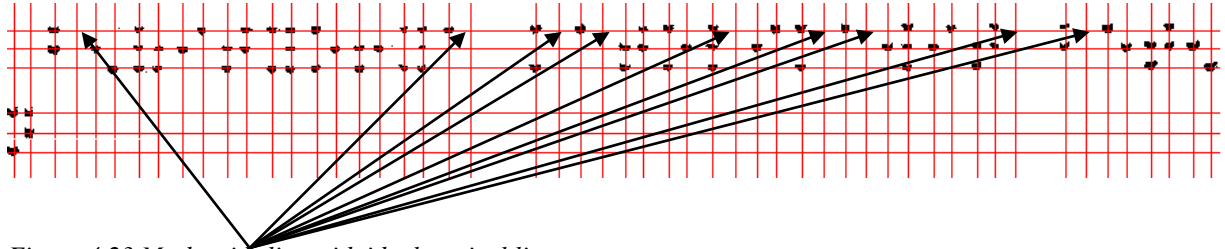


Figure 4.23 Mesh grids line with ideal vertical line.

```

int dot[2000][2],dotcount,h,k,w,da;//da index of array
bool hit=FALSE;
do{
    if(dc.GetPixel(0,h)==255)//check vertical mesh grids
    {
        do{
            if(dc.GetPixel(w,0)==255){//chkec horizontal mesh 255 is red color value
                for(int z=h-5;z<=h+5;z++){
                    for(int y=w-5;y<=w+5;y++){
                        if(pDC->GetPixel(y,z)==0)
                            dotcount++;}
                    if(dotcount>=10)
                        {dot[da][0]=w; dot[da][1]=h; dotcount=0; da++;
                        }w++;
                }
            }while(dc.GetPixel(w++,0)!=255&&w<bm.bmWidth);
            w=0;h++;
        }
    }while(dc.GetPixel(0,h++)!=255&&h<bm.bmHeight);
    for(int i=0;i<bm.bmHeight;i++)//convert background in to black!!
        for(int j=0;j<bm.bmWidth;j++){
            if(dc.GetPixel(j,i)!=0)
                dc.SetPixel(j,i,0);
        }
    k=0;
    while(dot[k][0])//set dot with white color pixel on black background!!!
    {
        dc.SetPixel(dot[k][0],dot[k][1],RGB(255,255,255)); k++;
    }
}

```

Figure 4.24 Implementation of mesh based segmentation

Once the adaptive 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. The program search black pixel around the intersection of the horizontal and vertical line to segment dot.

The segmentation given in *Figure 4.24* algorithm starts by determining the first intersection point of the mesh on the Braille image. Then the algorithm search black pixel in all direction around the intersection point to declare the presence of valid dot which would be the possible location of the dot. The search made within the area of 5 pixel to the left, right, top, and bottom from the intersection point. And a threshold value is set experimentally to 10 and more black pixels to announce valid dot. The value is relaxed to consider the minimum black pixel a Braille dot may contain and the red pixel of the grids line that cross the dots. If the black pixels meet the threshold, the (x, y) coordinate values of that point is registered in an array prepared for such purpose. This continues for the whole mesh intersection points and the corresponding (x, y) coordinates is registered if there is a dot on that coordinate. This step is important to clearly display the dot with single white pixel on black background as shown in the picture below. Finally, the algorithm convert the whole image in to black color and set the dot coordinates value with one pixel of white color. To set the white pixel that represents the presence of dot, (x, y) coordinate value is taken from previously registered array. Sample result of the segmentation algorithm is given in *Figure 4.25*.

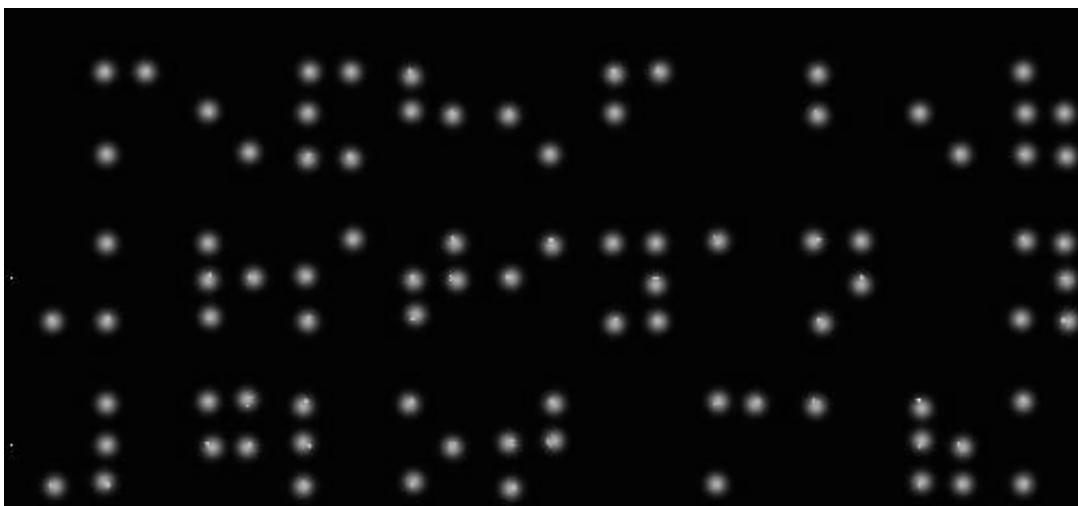


Figure 4.25 Segmented Braille image

The segmentation module which depends on the constructed grids mesh is robust and tolerates noise. This is because the algorithm does not require checking every pixel in the image; instead only potential dots areas are checked for the threshold value; there by also reducing the computation time. Moreover black pixels which are found outside the mesh coordinates are not checked. This provides the program to tolerate noise. The algorithm is found effective for different Braille image with varieties of dot size. And performs well with clean image and tolerates some scattered noise.

Once the segmentation has completed, the resulting image is used to further process dots by grouping in to Braille cell to draw patterns for Amharic Braille code.

4.6 Amharic Braille Feature Extraction and Recognition

It is discussed in the literature review section 2.7 that a Braille character is made up of six dots arranged in two column and three rows. Dot on the six possible place can be turned on (raised) or flat to create Braille character. Depending on the type of print character used in Amharic language, one cell (six dots), two cells (12 dots) or/and rarely three cells (18 dots) may be used in Braille writing system for the corresponding character in print. That is why the Braille documents blamed to be bulky and take much volume in hard copy version. The following part describe which type of print character requires one, two and/or three cells; and decisions made to determine the number of cells used to represent print equivalent to come up with standard input for the network.

Amharic Braille, like the print Amharic writing system applies all the basic characters with their six forms (34 x 6). While all the basic forms (34) are represented with a single Braille cell, the rest requires two cells (the first cell indicating the basic form, and the second cell the vowel (sound) for one of the six forms). To illustrate the situation, sample character is given in that shows Braille code and print. *Table 4.3* shows one-to-one mapping between Braille-to-print characters. In the following illustration to represent dots in a cell, the dot position is labeled with number 1 through 6. And to represent the dot is active for a particular

label binary digit 1 will be used.

Table 4.3 Basic Amharic Braille character with one cell

1	2	3	4	5	6	
1	1			1		ሀ
1	1	1				ለ
1	1				1	ሐ
1		1	1			ዎ
	1	1	1			ሠ
1	1	1		1		ረ

To write the first forms for character in **Table 4.3**, we use two cells as shown in **Table 4.4**:

Table 4.4 First variant of basic Amharic Braille character with tow cells.

1	2	3	4	5	6	1	2	3	4	5	6	
1	1			1		1				1		ሀ
1	1	1				1				1		ለ
1	1				1	1				1		ሐ
1		1	1			1				1		ዎ
	1	1	1			1				1		ሠ
1	1	1		1		1				1		ረ

This indicates that the majority of the Amharic print characters require two cells with the exception of the 34 basic form characters.

With regard to numbers, the Amharic Braille applies both Ethiopic and Arabic numerals system. Both numerals system require two Braille cells for their representation. The difference however, is on numeral symbol dot combination that precedes the numbers. The numerals mode defined for Ethiopic number is “111111”, while “001111” used for Arabic numbers (see section 2.4.1 for more). Accordingly, the first cell in numerals defines the numerals mode (Ethiopic or Arabic) and the second cell defines the numbers (0-9). Table 4.5 and Table 4.6 illustrate the numerals writing system in Braille for Ethiopic and Arabic numbers respectively:

Table 4.5 Arabic Braille numerals with two cells.

1	2	3	4	5	6	1	2	3	4	5	6	
0	0	1	1	1	1	1	0	0	0	0	0	1

0	0	1	1	1	1	1	1	0	0	0	0	2
0	0	1	1	1	1	1	0	0	1	0	0	3
0	0	1	1	1	1	1	0	0	1	1	0	4
0	0	1	1	1	1	1	0	0	0	1	0	5
0	0	1	1	1	1	1	1	0	1	0	0	6
0	0	1	1	1	1	1	1	0	1	1	0	7
0	0	1	1	1	1	1	1	0	0	1	0	8
0	0	1	1	1	1	0	1	0	1	0	0	9
0	0	1	1	1	1	0	1	0	1	1	0	0

Table 4.6 Amharic Braille numerals with two cells.

1	2	3	4	5	6	1	2	3	4	5	6	
1	1	1	1	1	1	1	0	0	0	0	0	፩
1	1	1	1	1	1	1	1	0	0	0	0	፪
1	1	1	1	1	1	1	0	0	1	0	0	፫
1	1	1	1	1	1	1	0	0	1	1	0	፬
1	1	1	1	1	1	1	0	0	0	1	0	፭
1	1	1	1	1	1	1	1	0	1	0	0	፮
1	1	1	1	1	1	1	1	0	1	1	0	፯
1	1	1	1	1	1	1	1	0	0	1	0	፰
1	1	1	1	1	1	0	1	0	1	0	0	፱
1	1	1	1	1	1	0	1	0	1	1	0	፲

The Amharic Braille also applies different punctuation marks used in print Amharic language. These days there are 41 punctuation marks that can be used in Braille writing. Except three rarely used punctuation marks, (indicated in Table 4.8), all punctuation marks require one or two Braille cell (see Table 4.7).

Table 4.7 Lists of punctuation marks with one and two cell.

1	2	3	4	5	6	1	2	3	4	5	6	Symbol description
0	0	0	0	0	0							Space
0	0	1	0	0	0							. Point
0	0	0	0	0	1	0	0	1	0	0	0	: colon
0	0	0	0	1	0	0	1	0	0	0	0	/ forward slash
0	0	1	0	0	1							-hyphen
0	0	1	0	0	1	0	0	1	0	0	1	---
0	1	0	0	0	0							' single quote
0	1	1	0	0	1							«»
0	1	0	0	1	1							:: full stop
0	1	1	0	1	1							() Brackets
0	0	0	0	0	1	0	1	1	0	1	1	[opening square bracket
0	1	1	0	1	1	0	0	1	0	0	0] closing square bracket
0	0	0	1	0	1	0	0	0	1	0	1	_ underline

Braille character alone or part of a Braille characters. This is because depending on the context, a Braille character may have one or two cells.

To prepare two different patterns input for neural network, a program is developed that represent Braille character as six and twelve dots. The program converts dots in a cell as binary digits. One (1) is assigned to indicate the presence (active) of dots and zero (0) is assigned for the absence (flat) of dots in a cell.

This is done following the algorithm described in appendix V. The extracted dots are grouped as one and two cells to prepare two different patterns for the neural network in the forth coming step. To do this the program scan one cell at a time, following the grids line coordinates constructed previously. The first cell may define the basic Amharic Braille character (prefix), number mode or punctuation mark; and the next six bit (suffix) defines the corresponding vowel, number or punctuation mark for particular character.

Once the dots in a cell accessed, to decide the status and the position (from two cells), the content has been checked for three criteria: Braille vowel, numerals and punctuation. Then the algorithm continues to check the next possible cell, if this is found to be among the six vowels, it will be registered as suffix of the previous cell. This is preferred because the sound cell in Amharic Braille cannot be found before the basic character. To make the concept clear the Amharic Fidel “ሀ” which is the sixth form of “ሀ”represent without vowel as “110010”. To write the remaining variant, however we need to add additional cell for the vowel as given in *Table 4.9*:

Table 4.9 Two cell Braille-print character representation

ሀ	1	1	0	0	1	0	0	1	0	0	0	1
ሀ•	1	1	0	0	1	0	1	0	1	0	0	1
ሁ	1	1	0	0	1	0	0	1	0	1	0	0
ሁ•	1	1	0	0	1	0	1	0	0	0	0	0
ሂ	1	1	0	0	1	0	1	0	0	0	1	0
ሂ•	1	1	0	0	1	0	0	0	0	0	0	0
ሃ	1	1	0	0	1	0	1	0	1	0	1	0

On the other hand, if the cell status is vowel, or is preceded by number mode or punctuation mark (suffix); the content fills the second cell (second 6 dots of cells). Moreover to give room for some Braille code that does not considered in this study: such as the remaining punctuation mark and extra Amharic character, all other cell that fail to meet the above criteria, are labeled as others cell content with value different from previously used (binary digit). The algorithm adopted for feature extraction and recognition is described in appendix V.

4.7 Amharic Braille-print character recognition using ANN

So far, all the potential Braille character features are extracted and stored to file with 6 and 12 bit that can be turned on(1) or off(0) in any combination. This section, that depend on the result of the previous steps, describes how the artificial neural network is created, trained and tested to classify an inputs pattern as one of the target output results. The overall task of the recognizer is to be able to classify pattern extracted (input, Braille character) into one of the possible target values (print character). For the purpose, the study apply MATLAB neural network tool box, which is the most commonly used tool in Neural Network researches such as creating recognizer. Thus, the next part of the study discusses in detail how the network is designed, created, trained and tested within the facilities and the functionalities of the MATLAB tool.

4.7.1 Artificial Neural Network Architecture

The first step to application of neural network to solve a specific problem is designing the architecture of the network. In this study, the architecture of the neural network created for the recognition problem is a multilayer perceptron that accepts the binary representation of Braille character as 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. Another input is prepared with 6 bits and 12 bits each from the Braille document. The two inputs with two different standards are prepared representing the Braille character for training and testing the network.

The diagram in *Figure 4.26* is designed to describe the architecture of the recognition system starting from the Braille character input to the print character recognition phase. The diagram depicts the two parts in the recognition process. The first part ends with the extraction of Braille character as binary digit which is accomplished with Ms-visual C++. The next part recognizes the print character which is accomplished with MATLAB tool.

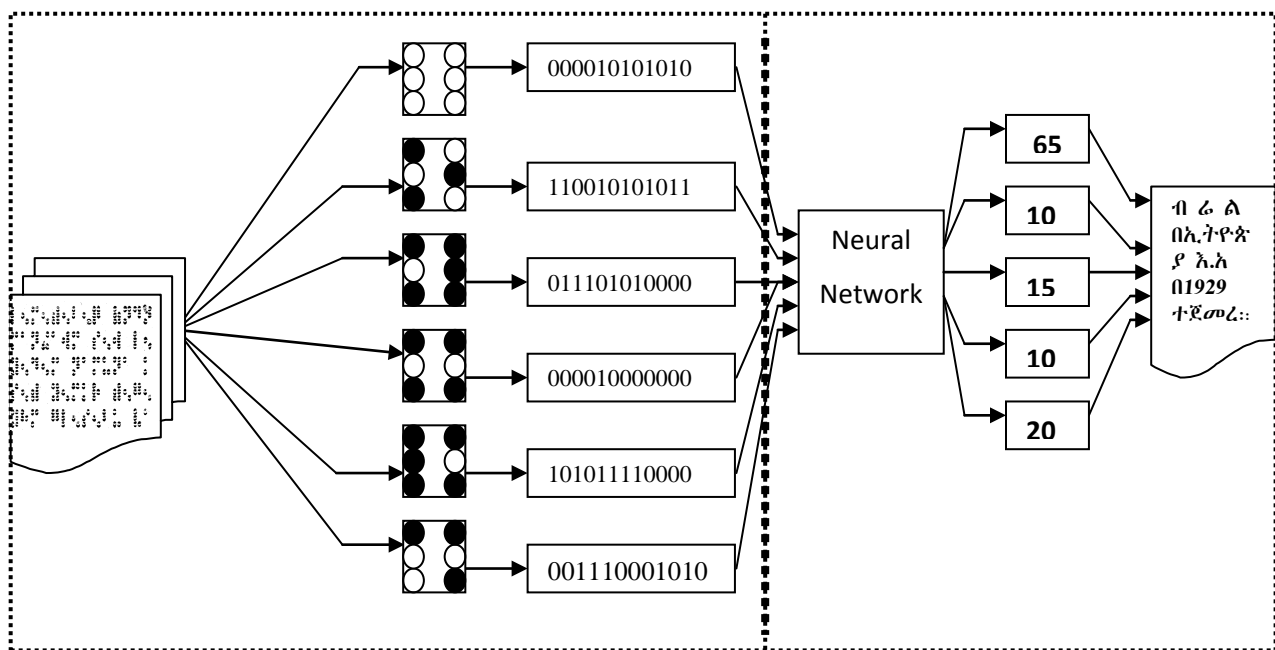


Figure 4.26 Amharic Braille-print recognition network architecture

4.7.2 *The Neural Network*

A feed-forward neural network with back propagation algorithm organized into three layers is created. The topology, or structure, of this network is typical of networks used for prediction and classification. 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 on each steps of training. Supervised learning is used, where the desired output of the network for the known patterns of input is fed to the network during training. Accordingly, the network takes inputs, perform some operation and provide the output using the feed forward approach. Then the mean squared error between the desired and the actual output is calculated. If this value is found to be greater than the threshold the error propagates back to adjust the weights of each connection in the course of finding the optimal weight for each connection.

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.

4.7.3 *Braille Character Input Method for the Neural Network*

In MATLAB tool, to train the network the input pattern is first rearrange into column vector which contain elements equal to the number of neurons at the input layer. Accordingly, binary representation of the Braille characters is created in a file as 6 and 12 bits pattern of Braille characters. The input is converted in to one column vector where each preceding column of the character matrix will be appended one after the other.

Table 4.10 Sample Amharic characters with same ASCII value

	Latin	ASCII		
ሁ	X@	88	64	152
ኧ	~	152		152
ቤ	u?	117	63	180

ዝ	'	180		180
ገ	%<	37	60	97
ሮ	a	97		97

Before making decision on output nodes it is worth to mention that all Amharic print characters (target) do not be represent with one ASCII- code value. Examples of such character are given in *Table 4.10*. This situation results, two different Amharic characters to have same value (when convert to decimal and add), which would bring confusion on later conversion to print. 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 mentioned in section 4.7.5 and this value is converted between the value 0 to 1 to fit the network output with Min-max normalization formula.

$$V' = \frac{V - \min_A}{\max_A - \min_A} (new_max_A - new_min_A) + new_min_A$$

Equation 4.1 Min-max normalization formula

Thus, the number of nodes at the output layer is decided to be one. After deciding on the possible number of nodes at each layer, the next phase is creating the network and proceeding with the training part, which is discussed in the subsequent part.

4.7.4 Creating the Neural Network

At this stage, everything is ready for creating and training the network. This part devoted to describe how the network is created. As indicated earlier, a feed-forward back-propagation network is created with gradient-descent momentum training and adaptive learning rate. Training the network with two parameters momentum and learning rate is important for two reasons. First, learning rate is used to specify how the weights are adjusted after calculating errors on the output layer. Likewise, the purpose served by the momentum parameter is that, once the learning network knows to which direction to move on the learning

curve, we can 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 will 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 will 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 minimum only to the points in its vicinity and not to the total learning curve.

With too low momentum constant the network will be slow and may get stuck in local minima. On the other hand, if it is set to large value, the network will have less probability of getting trapped in local minima and will converge fast, but it may also oscillate from one point to the other in the curve without touching the global minima.

Based on the above assumption, in MATLAB tool such facility to train the network is accessed with training function TRAINLM. Weight and bias value are updated in TRAINLM network training function according to gradient descent momentum and with adaptive learning rate. 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. Function like TRAINRP has fastest algorithm, but it does not perform well on function approximation problems. Its performance also degrades as the error goal is reduced.

To create network based on TRAINLM using MATLAB, the command is:

```
AmharicBrailleNetwork = newff(input_range,[], {'tansig', 'purelin'}, 'trainlm');
```

Where: input_range is the minimum and maximum values at the input nodes, {'logsig', 'logsig'}; is the activation function at the hidden and output layers independently. [12, 1]; is size of neurons at hidden and output layer respectively

4.7.5 Training the Neural Network

Once the neural network is created, the dataset prepared for the purpose is introduced to the network to achieve the intended target output. With the objective to consider different pattern (with one/two cells) Braille character and character type, three training dataset are prepared with the following proportion to train the two networks:

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

To train the network, different network parameters such as: goal, epochs, learning rate are adjusted as per the performance of the network (see *Table 4.11*), and then input patterns along with matrix of target values fed to the network for training.

Table 4.11 Training parameters for the proposed network

functions: adaptFcn: 'trains' performFcn: 'mse' trainFcn: 'trainlm'	parameters: goal: 0.001 lr: 0.2 show:50 epoch: 1000
--	---

The line of instruction to perform the training is:

{AmharicBrailleNet, tr}=train (AmharicBrailleNet, BrailleInput, PrintTarget);

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

Accordingly, to train the network the three 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 and with different noise pattern which are randomly generated by the training module. And there is 50 iteration, which makes a total of 700 for each character pattern, which are provided in the training set. The random noise is set to 0.03 and 0.05.

4.7.6 Testing the Neural Network

Once training the artificial neural network is performed, it is required to test the performance of the network model with different test cases. In this study, testing is conducted in two steps:

The first one is testing the network with the pattern that the network trained (training set), and the other testing is done with patterns extracted from the Braille document which was 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 used to test the network trained so far for the three training set.

R=sim (AmharicBrailleNet, Test);

where: R is the output of the network, AmharicBrailleNet is the network created and trained for different patterns, Test is the testing input pattern, and

Thus, the networks created for three training set with different numbers of input nodes (six and twelve) are tested for accuracy of classifying new character pattern to one of the 34 and 267 characters of the particular dataset. The output of the network could range between 0 and 1 for the reason described in section 4.7.4. Then, this value can be converted back in to the indexed value in the range 1-34, and 1-267.

At this point, to see whether the network can recognize the character set used for training, all the three training dataset that contain 34, 238, 257 and 267 characters set is fed to the network independently.

Accordingly, analysis of the experiments shows that the neural network could not learn the following character:

From 238 Amharic characters 2 are error.
 From 10 Numerals 9 are error.
 From 19 Punctuation Marks 5 are error.

Table 4.12 Error type and list in validating the network

Input pattern given												Expected Target	Result(error)
1	1	0	0	1	0	0	1	0	0	0	1	ሀ	ሀ
1	1	0	0	1	0	1	0	1	0	0	1	ሁ	ሂ
0	0	1	1	1	1	1	1	0	0	0	0	2	4
0	0	1	1	1	1	1	0	0	1	0	0	3	2
0	0	1	1	1	1	1	0	0	1	1	0	4	5
0	0	1	1	1	1	1	0	0	0	1	0	5	4
0	0	1	1	1	1	1	1	0	1	0	0	6	5
0	0	1	1	1	1	1	1	0	1	1	0	7	8
0	0	1	1	1	1	1	1	0	0	1	0	8	7
0	0	1	1	1	1	0	1	0	1	0	0	9	8
0	0	1	1	1	1	0	1	0	1	1	0	0	space
0	0	1	0	0	1	0	0	0	0	0	0	-	[
0	0	1	0	0	1	0	0	1	0	0	1	---]
0	1	1	0	0	1	0	0	0	0	0	0	«»	፪፫
0	0	1	0	0	1	0	0	0	0	0	1	---	[
0	0	1	0	0	1	0	0	1	0	0	1	Line between words]

Evaluation of the performance of the model with the training set indicates that the majority of error happens in numerals and punctuation marks. The possible reason for this is that the number of bits size used is small and some pattern have similar pattern that are differentiated with one bit. It has been

shown that the model is not sensitive to randomly generated noise level at 0.05. However, the network has not been result satisfactory result when the random noise is increased. To improve the performance other training function were tried however, the result was the same.

4.7.6.1 Performance Evaluation with one and two cell test dataset

At this stage, the network is then tested for test set extracted from Braille document. Three test set has been prepared with a total of 550 characters. This time the performance is tested with 12 node network. The reason is that most of the documents consist of a two cell Braille characters.

- Test 1: contains a total of 100 characters with two cells.
- Test 2: contains a total of 350 characters with two cells.
- Test 3: contains test 2 and additional 100 characters set.

Accordingly, test set 1, test set 2 and test set 3 have been given to Model-1, Model-2 and Model-3 respectively. The performance of the neural network for the three test dataset is given below in *Table 4.13*.

Table 4.13 Performance rates for test dataset

Network	Performance /Rate of Recognition	
	Test Set size	Test Set
Model -1 with 238	100	96 %
Model -2 with 257	350	94 %
Model -3 with 267	450	92.5 %

The performance of the network is decreased with the test set. This is because there are some characters found in the test document for which the network did not trained. This includes the excluded punctuation

marks and numerals.

Finally, from the three models the researcher selects the one with 92.5% performance. The rationale is that this model trained with training set contained in the other two models plus additional numerals set. And as discussed in above the majority of the error attribute to numerals and punctuation marks. However, for the remaining dataset the three network models are produced similar performance. So it would be better to consider the one that consider wide variety of character than the one with less character set. The artificial neural network algorithm, adapted for the study is presented in *Figure 3.6*.

Chapter Five

5. Conclusions and Recommendation

5.1 Introduction

In this study an attempt has been made to recognize the Amharic Braille from image scanned with standard scanning device. Braille system is an invaluable means for visually impaired society to communicate with the vision society which use dots arranged in two columns and three rows, called Braille cell. To accomplish Braille recognition different algorithms are reviewed, adopted and used. To this end, the part followed discuss the major tasks performed, followed by forwarding set of recommendations for future research direction.

5.2 Conclusion

Since its invention by Luis Braille, there is no other means that found to replace the Braille. So far a great deal of research work has been conducted to transcribe print document to reach to the visually impaired society. However, little efforts have been made to work on the reverse side, transcription of Braille to print document. To this end, research works revealed that with the advent of optical character recognition (OCR) technology contents of a document can be extracted and made machine readable.

Optical character recognition devised a wide variety of approaches to recognize most print character image. However, in Braille there are many challenges as the document are not intended for vision society. And the Braille does not intend to convey any visual information; the content is not visible in black and white as print document.

In this study, different preprocessing algorithms such as: global thresholding, adaptive mesh segmentation, and context based feature extraction and feed forward neural network has been adopted for

recognition of Braille image.

Digitization process in Braille requires careful preparation. This is because reflection of light intensity affects the resulting image. Flat-bed scanner has been the best alternative for this purpose; and the document is scanned with 200dpi (dot per inch).

In image preprocessing, first image converted to gray-level to reduce the computational complexity. For this, the study adopts global thresholding techniques. The threshold has been determined through experimentation based on analysis of image pixel level frequency. The technique performs well on sample documents. However, two problems are worth to mention on the resulting image. The first is that there are some overlapping dots because of the dots size. Moreover, because of impurity of the document some noise exists as a valid dot point.

Next, based on the result of threshold image, mesh-grid technique has been used to segment the dot point. An adaptive mesh has been developed to construct the grids line based on locally determined dots size. In adaptive mesh the grids line is drawn only for the presence of at least one dot across three rows or down two columns of a cell. Once the entire mesh has been constructed, segmentation identifies dot point following the mesh coordinates. In this case based on threshold the algorithm search black pixel around each coordinate points and if dot count meet threshold, a pixel with white color set to that particular coordinates representing the presence of dot with a black background.

Unlike print character, Braille does not have any structure to define to extract the feature of the character. So context analysis is the preferred technique with defined rules. For this reason feature extraction also takes advantage of the mesh-grids. The approach access one cell at a time and determine the status of the cell within a Braille character. Braille character, depending on the context may be defined in one or more cell. In Amharic Braille system basic character defined with one cell. Whereas, the remaining six variant

defined with two cell. All Braille numbers requires two cells: for its numeral mode (symbol) and for the specific number (digit). Most punctuation marks are defined with one or two cells. Thus, extraction of the Braille character is made based on the context analysis of dots in a cell.

In Braille-to-print character recognition, attempt has been made to apply the neural network approach. Accordingly, a feed-forward multilayer perceptron network has been designed with two possibilities for the number of nodes at the input layer. The network is created in MATLAB tools. The network has been trained with TRAINLM training function, and two parameters including learning rate and momentum constant. The rationale to apply learning rate is to let the network adjust learning value after each iteration, whereas the momentum constant enable the network not to trap in local minima and converge quickly.

Amharic Braille system has been used all printed Amharic character, punctuation marks and numbers. In the current research an attempt has been made to include all the 238 (7x34) character, 19 punctuation marks and Arabic numerals. The classifier which is performed with neural net work produced satisfactory result with random noise generated at 0.05. With noise above 0.05 the network does not respond satisfactory result.

To sum up, successful result has been achieved in the study in all the steps of Braille-to –print character recognition. However, during experimentation different challenges has been investigated. And attempt has been made to address some of them which are within the scope of the study. To this end, in order to devise a full-fledged solution advanced preprocessing and post processing techniques needs to be adopted for Braille image. The following part ends by forwarding recommendation for future work.

5.3 Recommendation

Braille document is not clean for the very reason that it is in a tactile format. Repeated use, bend and scratch are also cumulates the image impurity challenge. Because of this the resulting image after Binarization process are noisy. During experimentation it is experienced that most of the noise are scattered type which will increase around the age of the document. On the other hand some documents produce noise that exists as if a valid dot. This is probably because of bend and any noise marks on the document. However, these conditions are undesirable for the reason that it will significantly affect the subsequent recognition process. So to address the problem more advanced noise removal techniques should be tested on the resulting image.

The current system would set the threshold for Binarization with trial and error. With this regard it would be more important to devised automatic thresholding system that may work with different image color.

In regard to color, the current study performs the experiment on gray-level image. So, further work can be tested with color image document.

On the other hand, it is known that Braille documents are voluminous and bulky because of the nature of the writing system. However, the study has shown that the possibility of keeping Braille documents out of print in computable format. So, future effort is important towards creating Braille digital library, retrieval system and Braille dictionary which would be basic in information era.

So far, there are little efforts made to develop computer application to produce Amharic Braille. Thus, the result of the current study will be a starting point to think such application.

In problem domain such as Braille recognition, grids based segmentation is appropriate and produce better result. The reason is that in Braille, cells adhere strict structure vertically and horizontally. The problem; however is that all dots in a document or in different documents may not be of the same size. Some are large and visible, while the others are small. Thus the grids line that would be expected to pass bisecting all dots in a rows or columns is not consistent over the entire image. As a result some dots left adjacent

below or above the grid lines. In such situation, unless the threshold for segmentation takes in to consideration the smallest dot size, valid dots with small size may be missed. So with this regard the current study has room for improvement. One way is to adopt interpolation technique to work with variations of dots size.

Image resulted from two threshold value would create a single dot separated by white space. This image would be more helpful to separate the two sides of double-sided Braille document. The current study does not examine such image result further. This is because double-sided Braille is not commonly applied in Amharic Braille. However, this would lay foundation in the future to separate two sides of double-sided Braille whenever expanded.

The neural network on the recognition of Braille-to-print character trained for all basic Amharic character including the Arabic character and some selected punctuation marks. With this regard, in order to realize the system fully, future work should be conducted on the entire character set.

Eventually, in the current study Braille-to-print recognition is performed with the application of neural network. One way to improve this is to apply neural network in the preceding steps of the current system such as feature extraction.

Reference

- [1]. “Better Education for the Visually Impaired”, Available at:
<http://www.expresscomputeronline.com/20071224/technology01.shtml>
- [2]. “Blind and Partially Sighted & Education & Training & Medical Research & Conditions.”
Available at : <http://rnib.smartchange.org/>
- [3]. “Twain standard From Wikipedia, the free encyclopedia.” Available at:
<http://en.wikipedia.org/wiki/TWAIN> .
- [4]. “United Nations expert group meeting on disability-sensitive policy and programme monitoring and evaluation.” UNHQ, New York, 3-5 December 2001. Available at URL:
<http://www.un.org/esa/socdev/enable/disid2001h.htm>
- [5]. “World summit on the information society.”, Geneva 2003- Tunis 2005. Ethiopia: Adaptive Technology Centre for Blind.
- [6]. AbdulMalik Al-Salman, et. Al. “An Arabic Optical Braille Recognition System. “, 2007.
Available at: http://www.esst.rnu.tn/utic/tica2007/sys_files/medias/docs/p15.pdf
- [7]. American Council of the Blind. Available at: <http://www.howstuffworks.com/framed.htm?parent=braille.htm&url=http://www.readingcenter.buffalo.edu/center/research/tmgr.html>
- [8]. Anderson D. and McNeil, G. “Artificial Neural Network Technologies: Data Analysis for Software.” Rome, NY: Rome Laboratory. 1992
- [9]. Becker, S. “Unsupervised learning procedures for neural networks. International Journal of Neural System, 1991 17-33.
- [10]. Bender, M.L. et al. (ed), “Language in Ethiopia”. London, Oxford University. 1976.
- [11]. Berhanu Aderaw, “Amharic Character Recognition using Artificial Neural Networks”, (Masters Thesis). Addis Ababa: Department of Electrical Engineering, Addis Ababa University, 1999.
- [12]. Berry, M. a. “Mastering Data Mining: The art and science of

Customer Relationship Management.” New York: John Wiley and sons inc. 2000

- [13]. Blenkhorn, P., “A System for Converting Braille into Print”, IEEE Transactions on Rehabilitation Engineering, Vol. 3, No. 2, pp. 215-221, June 1995
- [14]. C. Ng and V. 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.
- [15]. Cambridge University. (1998). Optical character recognition.
- [16]. Dereje Teferi, "Optical Character Recognition of Typewritten Amharic Text. (Masters Thesis). Addis Ababa, School of information Studies for Africa, Addis Ababa University, 1999.
- [17]. E. R. DAVIES. “Machine Vision, Theory Algorithms Practicalities.” Elsevier Third Edition. 2003
- [18]. Ermias Ababe, "Recognition of Formatted Amharic Text Using Optical Character Recogniton". (Masters Thesis). Addis Ababa, School of Information Studies for Africa, Addis Ababa University, 1998.
- [19]. Fauset, L. “Foundation of Neural Network.” New Jersey: Prentice-Hall. 1994
- [20]. Getaneh Hamid. “Provision of Special Needs Education for the Blind in Sebbeta and Walaita Schools.” (Masters Thesis). Addis Ababa University. 1999.
- [21]. H. Bunke and P.S.P. Wang. “Handbook of character recognition and document image analysis. “, 1997
- [22]. Hentzschel, T. W., and P. Blenkhorn. “An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach”, Journal of Microcomputer Applications, pp. 341-345. 1995.
- [23]. Hermida, X. F., et al. "A Braille OCR for Blind People. “, Proceedings of ICSPAT-96. Boston (U.S.A.).October, 1996.
- [24]. Hykin, S. “Neural Network-A comprehensive foundation. “, Upper Saddle River,NJ:

Prentice Hall. 1999

- [25]. Id21, Institute of Development Studies University of Sussex Brighton BN1 9RE, UK. "Breaking barriers Building access for disabled people." Available at: <http://www.id21.org/insights/insights55/insights55.pdf>
- [26]. J. Mennens, L. V. Tichelen, G. Francois, and J. Engelen. "Optical recognition of Braille writing using standard equipment." *IEEE Transactions on Rehabilitation Engineering*, 2(4): pp. 207–212, December 1994.
- [27]. Jiawei Han et al. "Data Mining: concepts and Techniques." 2001
- [28]. Mennens, J., et al, "Optical Recognition of Braille Writing Using Standard Equipment", *IEEE transactions of rehabilitation engineering*, Vol. 2, No. 4, pp. 207-212, December 1994.
- [29]. Mennens, J., et al, "Optical Recognition of Braille Writing", *IEEE*, pp. 428-431, 1993
- [30]. Michael J.A. Berry, Gordon S. Linoff. "Data Mining Techniques for Marketing, Sales, and Customer Relationship Management Second Edition." Wiley Publishing, Inc. 2004.
- [31]. Million Meshesha, "A Generalized Approach to character Recogniton of Amharic Texts", (Masters Thesis). Addis Ababa, School of information Studies for Africa, Addis Ababa University, 2000.
- [32]. Néstor Falcón et. al. "Image Processing Techniques for Braille Writing Recognition." *EUROCAST, LNCS 3643*, pp. 379 – 385, 2005.
- [33]. New York institute for special Education. "History of Braille." *Blindness Resource Center*, 2008. Available at : <http://www.nyise.org/blind/barbier2.htm>
- [34]. Perlovsky, L. I. "Neural Network and Intellect: Model Based Concept." New York: Oxford University Press. 2001
- [35]. Radial basis networks (neural network toolbox): Probabilistic neural networks. [Online] Available: http://www.mathworks.com/access/helpdesk/help/toolbox/n_net/radial10.shtml#2412, c_1994-2003. The MathWorks, Inc.

- [36]. Ritchings R.T et. al. "Analysis of scanned Braille document." World scientific Publishing co. pp 413-421. 1995
- [37]. Sarles, W. Retrieved 10 11, 2008, from Neural Network-FAQ ,periodic posting to the usenet news group comp.ai.neural-net: 1997 Available at: ftp://ftp.sas.com/pub.neural/FAQ.html
- [38]. Statsoft. Retrieved 11 02, 2008, from Neural Network: 2003
http://www.statsoftinc.com/textbook/steneunet.html
- [39]. Wikipedia free. Braille Available at http:// www.wikipedia.com.
- [40]. Wondwossen Mulugeta. "OCR for Special type of Handwritten Amharic text ("YEKUM TSIFET"). " (Masters Thesis). Addis Ababa, Department of information science, Addis Ababa University, 2004.
- [41]. Wong, L., W. Abdulla, and S. Hussmann. "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", the 17th conference of the International Conference in Pattern Recognition, Cambridge, UK, pp. 23-26, August 2004.
- [42]. Worku Alemu. "The application of OCR techniques to the Amharic script." (Mastrs Thesis). Addis Ababa, school of Information Studies for Africa, Addis Ababa University. 1997
- [43]. በልዩ ትምህርት ቤቱን ሥ/ት/ዝ/ጥናትና ምርምር ኢንስቲትዩት። የብሬል ማስተማሪያ የመምህሩ መመሪያ። አዲስ አበባ 1991።
- [44]. በኢትዮጵያ አይነስውራን ብሔራዊ ማህበር 12ኛ ዙር ስራ አስፈፃሚ ኮሚቴ የተቋቋመው የአማርኛ ብሬል አሻሻይ ኮሚቴ። የአማርኛ ብሬል አሻሻይ ኮሚቴ ዘገባ። ታህሳስ 19 እና 20 ቀን 1995 ዓ.ም. ለተጠራው አገር አቀፍ ጉባኤ የቀረበ። 1995
- [45]. ጌታነህ አበበ። የአማርኛ ብሬል ከነሀሴ 24-28/1991 በሰበታ እውራን ት/ቤት አውደ ጥናት የቀረበ። ። በአፀደ ሕፃ/ልዩና መ/ተ/ሥ/ት/ዝ/ዋና ክፍል። ሥ/ትም/ዝ/ክ/ም/ኢንስቲትዩት። 1984
- [46]. ጌታነህ አበበ። የአማርኛ ብሬል ንባብና ጽሕፈት መማሪያ የመምህሩ መመሪያ። በአፀደ ሕፃ/ልዩና መ/ተ/ሥ/ት/ዝ/ዋና ክፍል። ሥ/ትም/ዝ/ክ/ም/ኢንስቲትዩት። 1984
- [47]. ነብየልዑል ዮሃንስ ዘመነ ብርሃን አዲስ አበባ ብርሃንና ሰላም ማተሚያ ቤት 1954።

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

Table I.2 List of vowels for first version Braille

ሀ	1:2:4	ኸ	2:3:4
ሐ	1:4:5	ወ	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
ቦ	1:3:4:5	ጠ	1:2:5:6
ተ	1:3:5	ጬ	1:2:3:5:6
ቸ	1:3:5:6	ጸ	2:4;5:6
ኀ	1:2:3:4	ፀ	2:4:5:6
ነ	1:2:3:4:5	ጸ	1:3:6
ፕ	1:2:3:4:5:6	ፈ	1:3:4:5:6
አ	1:2:3:5	ፕ	2:3:4:5:6
ከ	1:2:6		

Table I.3 List of first Version Extended Amharic Braille Character

ቈ	1:2:3:4:6 2:3:4
ከፐ	1:2:3:4:6 1:3:4
ኀፐ	1:2:3:4:6 1:2:3:4
ነፐ	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

1 st		7 th		1 st		7 th	
ሀ	1:2:5	ሀ	1:3:4:6	ሸ	2:3:6	ሸ	5 2:3:6
ለ	4:5:6	ለ	1:2:3	ወ	2:4:5:6	ወ	2:4:6
ሐ	Not apply			ዐ	Not apply		
መ	1:3:4	ፆ	2:3	ዘ	1:3:5:6	ዘ	2:3:4:6
ሠ	2:3:4	ሥ	5	ዠ	3:5:6	ዠ	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	ደ	1:4:5:6
ሸ	1:4:6	ሸ	1:5:6	ጀ	2:4:5	ጀ	1:2:6
ቀ	1:2:3:4:5	ቀ	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	ጬ	1:4	ጭ	3:6
ቸ	1:6	ች	2:5	ጸ	2:3:5	ጹ	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	ጺ	Not apply		
ጥ	3:4:6	ጥ	2:6	ፈ	1:2:4	ፍ	5 1:2:4
አ	3	አ	3:4	ጥ			
ከ	1:3	ከ	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	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th

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

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

6 th		6 th	
ሀ	1:2:5	ኸ	2:3:6
ለ	1:2:3	ወ	2:4:5:6
ሐ	Not apply	ዐ	Not apply
ም	1:3:4	ዠ	1:3:5:6
ሥ	2:3:4	ኧ	3:5:6
ር	1:2:3:5	ይ	1:3:4:5:6
ሰ	Not apply	ደ	1:4:5
ሸ	1:4:6	ጀ	2:4:5
ቀ	1:2:3:4:5	ግ	1:2:4:5
ብ	1:2	ጥ	2:3:4:5:6
ተ	2:3:4:5	ፆ	1:4
ች	1:6	ጸ	2:3:5
ገ	Not apply	ዕ	1:2:3:4:6
ን	1:3:4:5	ጻ	Not apply
ኝ	3:4:6	ፍ	1:2:4
ኧ	*1:2:3:5:6	ጥ	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
ከፊ	1:2:3:4:5 2:4:5
ገፊ	1:2:5 2:4:5:6
ገፊ	1:2:4:5 2:4:5:6

IV. The fourth version Amharic Braille (1949E.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, 3 and 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
እና ወይም	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. Feature extraction Algorithm

```

Let cell_1 and cell_2 are array to store prefix and suffix of a character respectively
Let BrailChar is an array defined to store a character containing cell_1 and cell_2
Let hordotArr is an array to store vertical cordinate point of grids intersection of image
Let VerdotArr is an array to store horizontal cordinate point of grids intersection of image
Let temp is an array to store current six dot as binary digit
Let img is matrix same size as image
Let dotColor is one pixel representation of dot at mesh cordinate
While h in hordotArr not null
    For each w in VerdotArr
        If pixel at img(w,h)equals to dotColor //First column dot position 1 2 3 starts here
            set temp[0]=1
        else
            set temp[0]=0
        end if
        if pixel at img(w,h++)equals to dotColor
            set temp[1]=1
        else
            set temp[1]=0
        end if
        if pixel at img(w,h+2)equals to dotColor
            set temp[2]=1
        else
            set temp[2]=0
        end if
        if pixel at img(w+1,h)equals to dotColor//Second column dot – 4 5 6 starts here
            set temp[3]=1
        else
            set temp[3]=0
        end if
        if pixel at img(w+1,h+1)equals to dotColor
            set temp[4]=1
        else
            set temp[4]=0
        end if
        if pixel at img(w+1,h+2)equals to dotColor
            set temp[5]=1
        else
            set temp[5]=0
        end if
        If isVowel(temp)equals to true//check content of a cell
            If BrailleChar cell_1 not empty
                buffer temp to cell_2
            Else
                buffer zero to cell_1
                buffer temp to cell_2
        //vowel comes after consonat cell otherwise that is wrong starting
        End if
    
```

Continued in the next page...

```

        Else if isnumberMode(BTemp)equals true or isPunctuationMarkPfix(BTemp) equals true
            buffer temp to cell_1
            buffer zero to cell_2
            go to next cell //untile checkes next cell cell_2 set to default zero
        Else if BrailleChar cell_1 not null AND isnumberMode(cell_1)is true
            buffer temp to cell_2
            go to next BrailleChar
        Else if isconsonant(temp)is true
            buffer temp to cell_1
            buffer zero to cell_2
            Go to next cell
        Else if isvowel(temp)is true and isPunctuationMarkPfix(BTemp)not true
            buffer temp to cell_2
        Else
            buffer unknown to cell_1
            buffer unknown to cell_2
            increment BrailleChar and Go to next cell
    End if
End for loop
end while
while BrailleChar not null //write BrailleChar to file
    write cell_1,cell_2 to file
end while
//function implementation
Begin function isnumberMode(integer temp) //check the current cell content is number mode
let numMode array for number mode//{0,0,1,1,1,1}
    for each x,y in numMode, temp
        if x not equals y
            return false
    return true
function end
Begin function isVowel(temp) //check the curent cell content is vowel and return
let vowel is an array for braille
character//{0,1,0,0,0,1};{1,0,1,0,0,1};{0,1,0,1,0,0};{1,0,0,0,0,0};{1,0,0,0,1,0};{1,0,1,0,1,0}
let hit is boolean
    for each cell in vowel
        set hit false
        for each x,y in cell,temp
            if x equal y
                set hit true
            else
                go to next cell//break inner loop
        if hit true
            return true
    return false
function end
Begin function isPunctuation(temp) //check cell content is one of punctuation mark prifix cell
let PuncMarkS is an array for punctuation prifix//{0,0,1,0,0,0;0,1,0,0,0,0;0,0,1,0,0,1;0,1,1,0,1,1;0,0,0,1,0,1;0,0,1,1,1,0}
    for cell in PuncMarks
        set hit false
        for x,y in cell,temp
            if x equals y
                set hit true
            else
                go to next cell//break inner loop
        if hit is true
            return true
    return false
function end

```

Figure V.1 Feature extraction algorithm based on grids-mesh and contextual approach

VI. Matlab code for Recognition

```
%neural network function implementation to create model for 6 and 12 bit Braille characters
function[ABOCRNetwork,tr]=ABOCRTrain(Train238,Train267,input_range,Target238,Target267)
AmharicBrailleNetwork=newff(input_range,[12 1],{'tansig','tansig'},'trainlm');
Time_Start=clock;
    AmharicBrailleNetwork.performFcn='mse';
    AmharicBrailleNetwork.trainparam.lr=0.2;
    AmharicBrailleNetwork.trainparam.goal=0.001;
    AmharicBrailleNetwork.trainparam.show=50;
    AmharicBrailleNetwork.trainparam.epochs=700;
    AmharicBrailleNetwork.trainparam.mc=0.8;
    AmharicBrailleNetwork.trainparam.min_grad=1.0000e-010;
    TrainSet=[Train238,Train267];
    TargetSet=[Train238,Target267];
%Training the network with the 238,257 and 267 pattern
    [AmharicBrailleNetwork,tr]=train(AmharicBrailleNetwork,TrainSet,TargetSet);
    AmharicBrailleNetwork_1=AmharicBrailleNetwork;
    AmharicBrailleNetwork_1.trainParam.goal=0.001;
    AmharicBrailleNetwork_1.trainParam.lr=0.02;
    AmharicBrailleNetwork_1.trainParam.epochs=1000;
    [R,Q]=size(Train267); T=[Target267];
%Training the network with noise
    P=[Train267+randn(R,Q)*0.05];
    [AmharicBrailleNetwork,tr]=train(AmharicBrailleNetwork_1,P,T);
%setting parameters to train with noise dataset
    AmharicBrailleNetwork.performFcn='mse';
    AmharicBrailleNetwork.trainParam.lr=0.2;
    AmharicBrailleNetwork.trainParam.goal=0.0001;
    AmharicBrailleNetwork.trainParam.show=50;
    AmharicBrailleNetwork.trainParam.epochs=1000;
    AmharicBrailleNetwork.trainParam.mc=0.8;
    AmharicBrailleNetwork.trainParam.min_grad=1e-15;
%Training the network after noise with the original pattern
    [AmharicBrailleNetwork,tr]=train(AmharicBrailleNetwork,TrainSet,TargetSet);
%simulate the trained data to see performance
    Rst=sim(AmharicBrailleNetwork,TrainSet);
%convert the output normalized value to index value
    Res=[((round(Rst*266+1))- (round(TargetSet*266+1))))];
    disp(Res); disp(size(Res));
    time=etime(clock,Time_Start);
    disp(time);
return;
```

VII. Sample Braille Image

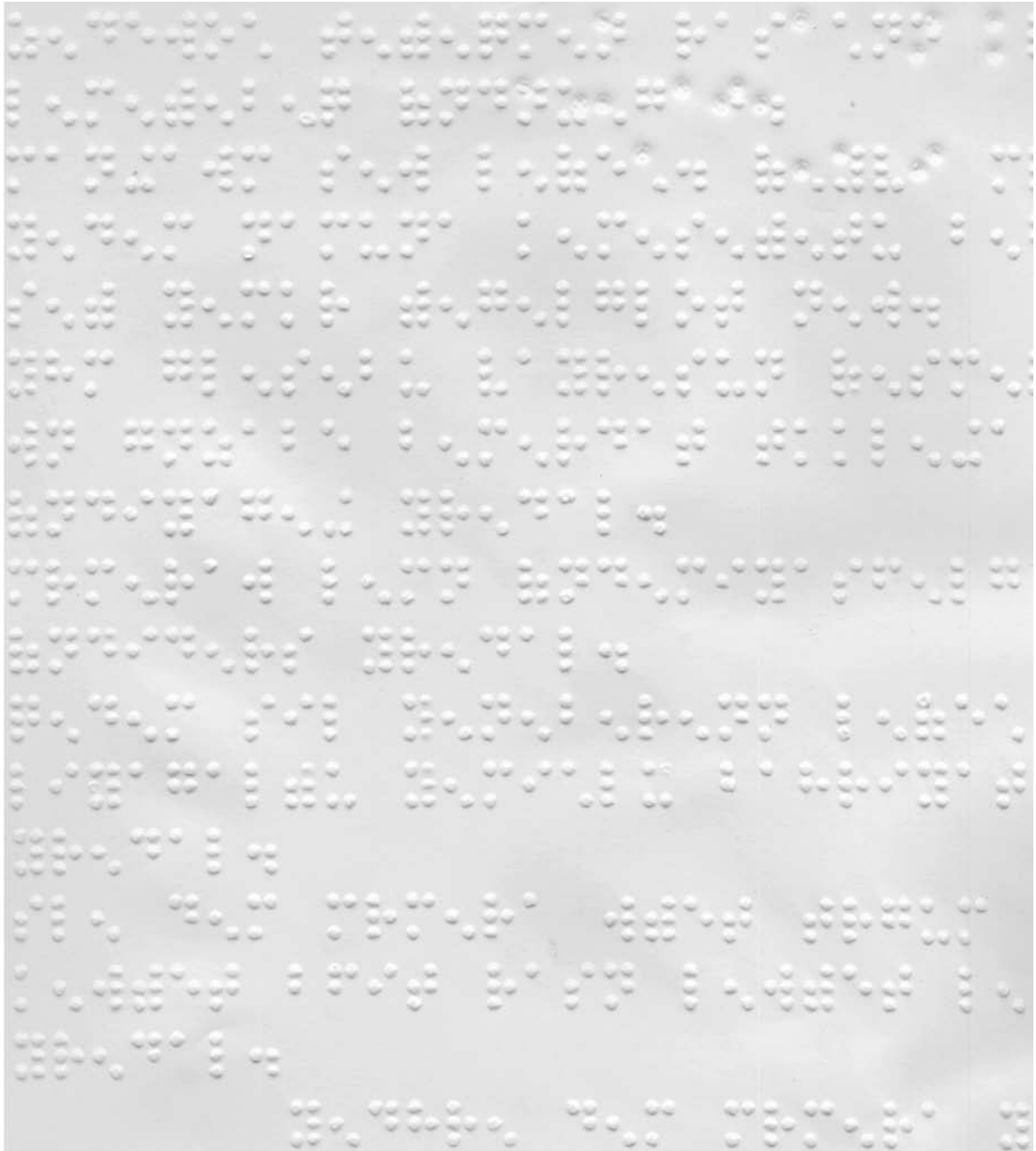


Figure VII.1 Sample Braille image with Gray-color

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.

Teshome Alemu

March 2009

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

Dr. Million Meshesha

March 2009