Addis Ababa University

School of graduate studies
Faculty of Informatics
Department of Information Science

HAND-WRITTEN AMHARIC CHARACTER RECOGNITION: THE CASE OF POSTAL ADDRESSES

A thesis submitted to the school of Graduate Studies of Addis Ababa University in partial fulfillment of the requirement for the Degree of Masters of Science in Information Science

By
Mesay Hailemariam

JUNE 2003

ADDIS ABABA UNIVERS
LIBRARIES
P.O.BOX 1176
ADDIS ABABA ETHIOPIA

DEDICATED TO:

MY FATHER ABA HAILEMARIAM MOREDA

AND

MY MOTHER EMAHOY ALEMAYEHU AFRO

ADDIS ABABA UNIVERSITY
FACULTY OF INFORMATICS
PHILOGRAPHIC LAB

Acknowledgement

My special thanks go to the management of Unity University College for the moral and material support it provided and for partially funding this research. I am deeply indebted to my brother and friend Ato Seyoum Tolla for his patience and support for many continuous nights. My sincere thanks is also forwarded to Ato Assefa Mammo and Ato Yigezu Tsegaye for their contribution in collecting data that I used for the research work.

I would also like to thank w/t Azmera Tesfaye and w/t Asnakech Mengistu for their help in typing the script and for their unforgettable coffee.

My greatest gratitude is extended to my Sister w/o Kebedech Hailemariam and my nephew Fikremariam Alemayehu for their economic support without which this research work would not be a reality.

Finally, I would like to extend my thanks to the community of department of Information Science in the faculty of Informatics whose contribution is involved in this work in one way or the other.

TABLE OF CONTENTS

ABSTRACT

Currently researchers are attracted to the area of Optical Character recognition primarily due to challenging nature of the research and secondly due to the industrial importance that it provides in the area of Reading machine for the Blind, postal Address interpretation, Bank Curtsey amount processing, hand filled form processing, and the like.

Research in the area of Amharic OCR systems is ongoing since 1997. Attempts were made in adopting algorithm to Amharic language, incorporating preprocessing techniques to the adopted algorithm, and in generalizing the system so as it recognizes Type written characters as well as hand written characters.

Sufficient amount of work is done in the areas of preprocessing such as segmentation and Noise Removal. However, the consideration given to the simplification of the feature extraction and the efforts made to alleviate the problems of high dimensional input still requires the contribution of many additional researches in order to come up with a system that the society can use to solve real world problems.

To this end, Line fitting is used to Amharic Optical character recognition by applying simple geometric calculations to determine features which could represent and describe the character as uniquely and precisely as possible. The image of a segmented character which is normalized into 32x32 pixels is divided into 16 smaller squares of 8x8 pixels. Then the least square technique was applied to fit a linear model to the distribution of foreground pixels and three features were extracted from each smaller square.

Finally, a feed forward Neural Network trained using a back propagation algorithm is used on handwriting of three individuals using a cross validation technique as well as a separate test set and results are depicted on tables and confusion matrices.

Relevant Conclusions were drawn and some valid recommendations were forwarded to indicate future direction of further works on the area.

combining the shape of the letters so as to form written words) [Plamondon and Srihari, 2000].

Handwriting, since it entails an individualistic skill and contains artificial graphical marks on the surface, is still a challenge in pattern recognition. The success of handwritten optical character recognition system is attributed to the availability of machine learning techniques [Lecun et.al 1998]. However, the availability of machine learning techniques alone is not able to solve the problems of offline OCR systems. To this end, some of the problems remain rather far away from being solved successfully.

Since 1951, a time remarked by the invention of GISMO – a robot reader writer, many OCR systems were developed due to the advantages that they provide in overcoming the problem of repetitive and labor intensive tasks [Srihari & Lam, 1996]. At present hundreds of OCR systems are commercially available, and they are less expensive, faster, and more reliable due to less expensive electronic components, and extensive researches in the area [Yaregal, 2002].

- Technically, Handwriting Recognition Systems comprise procedures like Scanning documents, Binarization, segmentation, feature extraction, recognition, and/or possible post processing [Million, 2000; De Lesa, 2001].

As Dereje mentioned in 1999, the OCR systems are highly influenced by factors like mode of writing, condition of the input, quality of the paper, and the presence of extraneous marks. In order to increase the performance of OCR systems, various preprocessing tasks like noise removal, skew detection and correction, and slant correction were applied to printed and type written scripts. Effort was also made in using structural features partly to increase the versatility of OCR systems [Yaregal, 2002].

In addition to the problems of machine printed and type written scripts, handwriting recognition has additional inconveniences introduced because of the great inconsistency of writing styles, and handwriting instruments.

1.2 STATEMENT OF THE PROBLEM

Since the early days of pattern recognition, it has been known that the variability and richness of natural patterns make it almost impossible to build an accurate recognition. One such pattern is a written text [Plamondon and Srihari, 2000].

Isolated handwritten character recognition has been studied in the literature and was one of the early successes in applications of neural networks [Lecun et. al, 1998; Ermias, 1998; Dereje, 1999; Yaregal, 2002]. These days, Europeans, Americans and others have been conducting researches and applying OCR technologies to their languages. As a result, these systems can read different documents written in English, Latin, Japanese, Chinese, Hindu, Arabic, Russian, and the like but do not read documents written in Amharic [Million, 2000]

Since 1997, after Worku conducted a research in adopting segmentation algorithm to the Amharic characters, researches on the Amharic language are ongoing. Ermias in 1998 attempted to incorporate preprocessing techniques (thinning and underline removal) to the adopted algorithm on formatted text. As a further work, in 1999 Dereje conducted a research in the area of improving Amharic OCR system by enabling it to recognize typewritten Amharic text in addition to printed ones. In 2000, Million attempted to work on the aim of generalizing the previously adopted algorithm. In the same year, Nigussie had investigated the recognition of Handwritten Amharic legal amounts of checks, the purpose of which is investigating the application of neural networks as a tool to recognize hand written Amharic Characters.

However, Amharic Handwriting Recognition is still an area that requires the contribution of many research works. One such area is simplifying the extraction of features which would represent and describe the characters as precisely and uniquely as possible. The other area of research is in improving the speed and reliability of recognizers so as users would develop confidence and motivated to use it.

Segmentation which is one of the important tasks in Character recognition, hiders the success of character level solution to the problem of handwriting recognition. This and other problems related to handwriting recognition attracted and challenged researchers to identify, test, and implement technological solutions [Plamondon and Srihari, 2000.]

The task of interpreting handwritten addresses is one of assigning a mail – piece image to a delivery address [Plamondon and Srihari, 2000]. An address for the purpose of physical mail delivery involves determining the country, state, city name, post office, street, primary numbers (which could be street numbers or a post office box) and secondary numbers(such as an apartment number) , and finally, the firm name or personal name[Plamondon and Srihari, 2000]

information in ones particular language highly depends on the fundamental characteristics of the language and state– of – the art technology [Plamondon and Srihari, 2000].

For written languages of developing countries which use their own written languages, digitizing the language is much harder due to the lack of extensive studies that reveal the fundamental characteristics of their languages. In addition to this, handwriting recognition by itself has many problems of its own. It is highly individualistic skill, which is influenced by numerous behavioral and environmental factors.

Since Amharic served as a working language of Ethiopia for many years, large amount of information is mounted up in churches, in caves, libraries, and private collections handwritten in this language. Accessing the contents of this information and providing it online for other users highly depends on the digitizing of the fundamental components of this language: its characters called 'Fidelat'.

Few researches were done on Amharic character recognition since 1997. Some of these researches investigated printed Amharic characters [Ermias, 1998; Dereje, 1999; Million, 2000] and one research is attempted to explore Handwritten Amharic text [Nigussie, 2000]. However, Amharic is yet far

behind from using the results of these offline Amharic text recognition researches. This is partly attributed to the complexity of the processes involved in the character recognition and/or the low success rate of the recognition results.

In addition to the above mentioned justifications, the persistence and convenience of handwriting in human communications by itself would call for researches to be conducted on handwriting recognition.

Automation of Handwriting recognition, nevertheless, highly depends on the ability of the computer to recognize the handwritten document. The success of handwritten document recognition could be influenced by the ability of the computer to recognize individual characters constituting the document.

However, recognition of handwritten characters is a formidably challenging task that contains chains and chains of activities – image capturing, Binarization, noise removal, segmentation, feature extraction, and recognition.

Since the success of character recognition highly depends on the significance, availability, and quantity of its features, feature extraction constitutes fundamental part in character recognition. In, researches conducted previously on Amharic Character recognition, efforts were made and are being made to use 'good' features. Some of the researches used a 16 x16 matrix of raw pixels (256 inputs) as an input for training and testing their Neural Networks[Worku, 1997; Dereje, 1999; Million 2000; Nigussie, 2000]. And some used 64 input nodes extracted as structural features [Yaregal, 2002].

Features that are relevant and important for the purpose of character recognition are highly needed for the success of Amharic Character Recognition.

This study attempted to explore into the possibilities of extracting relevant and important features by applying less complex geometric calculations and regression analysis techniques. It also attempted to reduce the number of input nodes of Neural Network recognizer considerably so as to increase its speed.

1.4 OBJECTIVE OF THE STUDY

1.4.1 GENERAL OBJECTIVE

The main objective of this study is to explore and test the application of simple geometric calculations and line fitting for handwritten Amharic Character Recognition by using characters used in writing of destination Addresses in Amharic language. It attempted to reduce number of inputs for a single character to the Neural Network recognizer in order increase the speed of the recognizer.

To meet the general objective of applying simple geometric calculations and statistical regression analysis to the area of character recognition the following specific objectives were met individually.

1.4.2 SPECIFIC OBJECTIVE

In order to meet the general objective,

- literature on general characteristics of Amharic writing system, line fitting and its application in pattern recognition area, algorithms and

techniques that are used for training and testing an OCR system were reviewed

- literature on address recognition , interpretation, and its application were reviewed
- handwritten Addresses were Collected and prepared for the experiment
- a Handwritten Amharic address recognizing system that fits this purpose was designed
- geometric features which are capable of representing and describing the Addresses were extracted
- prototype program for the character recognizer was developed
- an appropriate machine learning approach, algorithm and Package was selected
- a training and test data set for the neural network recognizer was prepared
- the recognizer was trained using the training data set
- the performance of the recognizer was tested using the test data set
- some recommendation for further studies were forwarded

1.5 METHODS APPLIED IN THE RESEARCH

The following techniques have been applied to undertake this research.

1.5.1 LITERATURE REVIEW

In order to fulfill its objective, extensive review of previous studies - both local and international have been conducted. Literatures on Amharic writing system, Amharic character recognition, Line fitting, machine learning, application of line fitting to the character recognition area, and MFC window programming were reviewed.

1.5.2 DATA COLLECTION TECHNIQUES

In offline handwriting recognition, only the completed writing is available. Hence, they are space ordered and the trace at the time of writing is not available like that of online handwriting recognition, where data consists of time ordered coordinate points. This makes the trace that the writer followed unknown and unavailable for processing in the offline handwriting recognition.[Plamondon, 2000]. The data of offline handwriting recognition thus, is available on paper which is inherently analog medium and should be

converted into digital form through scanning in order to get an image that a computer can process [Yaregal, 20002; Dereje, 1999].

The input of this Amharic character recognition system is image of 196 handwritten addresses that were collected from address books of three different individuals. These addresses were written by the individuals themselves on normal A4 papers eight addresses per page. Then they were converted to computer processable form by scanning on HP ScanJet series 3500c scanner.

With this regard, the data were collected, scanned, and saved in a monochrome bitmap format for further processes. Before, the end of the data collection phase, a small survey was conducted to get the data of most frequently used Amharic characters (see table 4.1) in the addresses so as the system would be trained on these characters.

1.5.3 DEVELOPMENT AND/ OR ADOPTION OF PATTERN EXTRACTION ALGORITHMS

In order to design and develop the prototype of this Handwritten Amharic character recognition system, line fitting approaches using the least square

methods on some geometric characteristics of the distribution of foreground pixels in a square region of scanned characters is used.

1.5.4 NEURAL NETWORK CLASSIFIERS

Character recognition is a process of assigning a predefined character tag to a set of input attributes [Perez et. al, nd]. Thus, it could be a classification problem and neural network classifier based on back Propagation algorithm was used to train and test the system. The feature extraction module should output the result as per the expectation of the classifying software: a comma or tab delimited data.

The out put of features extraction module of this research is a text file containing matrix of decimal numbers calculated from scanned character images separated with a comma A machine learning toolkit that is robust in handling such an output is selected and WEKA was found to serve the purpose. Thus, one of the reasons to choose WEKA machine learning over a Brain Maker software is its file handling ability and second reason is its convenient Graphical User Interface that enables easy way of adjusting learning parameters, and moreover its ability in helping the analysis of the result.

Thus, the Neural Network classifier integrated in WEKA was used to train and test the system.

1.5.5 TRAINING AND TESTING

The Neural Network classifier was trained by using features extracted from 415 character images from the handwriting of three writers: 175 character images from writer D, 136 character images from writer A, and 104 character images from Writer Y). Then matrices of features were prepared for all the three writers (49x175 for writer D, 49x136 for writer A, and 49x104 for writer Y) and the features were comma separated and saved separately in three files and their combination(415x49 matrix) was saved in another file with the format the recognizer requires.

In a matrix of 49xF (F is the number of character images) feature space, 48 of the 49 are features extracted from 16 square regions of the character image. And the features extracted from all the three writers were combined to supply a training data of 415x49 matrixes.

In machine learning, a very large size of training dataset is required to train the system which is some times exponentially related to the number of the input. This situation is called 'curse of dimensionality' [Trier et. al, 1996;

Mori et. al, 1999]. If adequate amount of training data could not be produced, a cross validation technique is used [Witt and Frank, 2000]. In this research, one of the training approaches is a ten fold cross validation is used which means that the classifier will first divide the training data into ten equal parts, train the itself on 9 of the divisions and then test on one portion left out for testing. Hence, each division gets a chance to be a test dataset.

The training and the testing in this research were organized in such a way that the system is trained in one of the handwritings using a ten fold cross validation technique to evaluate its performance on the handwriting of single writer. Moreover, the system was also tested on the other three datasets individually in order to see how robust the system is and on the combination of the three writers to see the scalability of the system (see table 4.2).

1.6 SCOPE AND LIMITATION OF THE STUDY

This study, like any research in handwriting recognition, was challenging due to the richness of natural patterns in the handwriting of individuals. It is a common approach in challenging areas as this one to constrain the research in a limited domain of application, and limit the type and amount of data used in the experiments. This research is also constrained and limited to the following conditions. The greatest problem of this research was the lack of reference materials on application of Line Fitting for the problem of

Character Recognition. Frankly, there was only one reference found on the WWW that gave an inspiration to this research. Thus, any Reference to this research, any conclusion about this research, and any comparison and interpretation of results should take these limitations into consideration in order to be valid. The scope and limitation of this research are:

- The research is on Handwritten Amharic characters considering only the characters used in writing destination addresses in postal addresses

- This research is limited to characters written on A4 paper not on real postal envelops.

- It uses only a linear model of regression to fit the distribution of foreground pixels in a cell. Using non linear methods are recommended for further investigation.

- Other classifiers were not tested using the features extracted by this technique. Hence, further works are encouraged on this line.

- Machine learning approaches other than classification were not considered in this research. Thus, extra work could be done on this area using the same technique of feature extraction methods.

- The application of line fitting for extraction of global feature of words is not in the scope of this research. Thus, studies investigating the application of line fitting at global level of extracting features of words are appreciated.

1.7 ORGANIZATION OF THE THESIS

This thesis work is organized in five chapters: the first chapter is the introduction that provides the background information, statement of the problem, justification of the study, and limitation of the study.

The second chapter discusses the Amharic writing system and some important features of the Amharic characters (Fidelat). The third chapter discusses line fitting, local line fitting and its application in pattern recognition, and the fundamentals of neural networks; the fourth chapter discusses the new Handwritten Amharic character recognition system applied to characters written on a normal A4 paper the fifth chapter forwards

some recommendations for further studies in the area of applying line fitting for the problem of handwriting recognition.

CHAPTER TWO

AMHARIC WRITING SYSTEM

2.1 HANDWRITING EVOLUTION

The history of hand writing as a means of communication and conveying ideas and information traces back to the days before the birth of Christ. It has started in the form of Egyptian pictorial writing – hieroglyphics that finally gave birth to most of the Middle Eastern scripts [Dereje, 1999]. The Geez script, which is derived from South Arabian alphabet called the Sabaean, is one of such scripts. The genetic structure of the Ethiopic, one of the ancient alphabets in the world, used to write in some Ethiopian languages such as Geez, Amharic, Tigrigna, etc) shows that Geez is at the top of the genetic tree of Ethiopic writing [Yonas et al., 1966 E.C.; Bender et al., 1976]

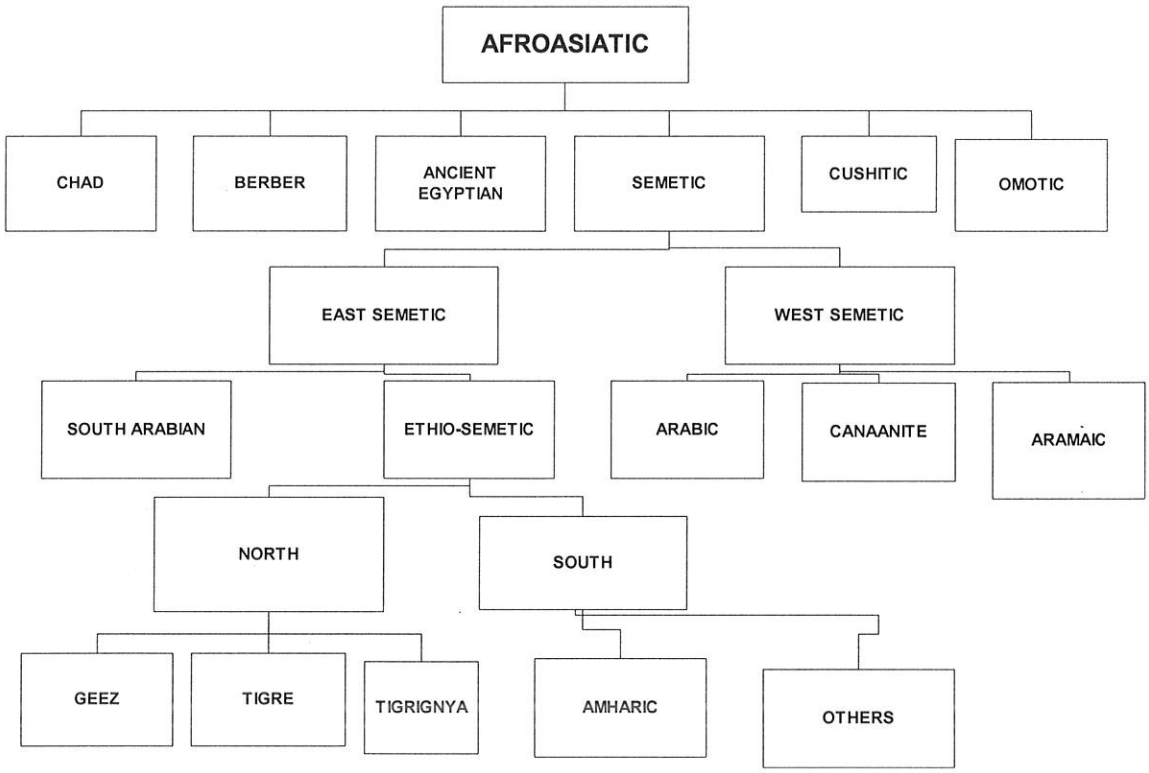

Fig 2.1 The Genetic Structure of Amharic Script [million, 2000; Yaregal, 2002]

The current writing system of Amharic is taken from Geez that in turn evolved out of Sabaean Language – the descendent of South Semitic Script. It was brought to highlands of Ethiopia by immigrants from South Arabia in the first century A.D [Bender et.al, 1976]. Geez, which remained the ecclesiastical and literary expression in Ethiopia until the 16th century, gradually gave way to Amharic that was used both in spoken and writing in the royal courts. It began to be used for literary purposes at the beginning of the 19th century as the administrative state changed its way of communication from oral to written one[Million, 2000].

Using the Sabaean Script for Ethiopic writing dates back to the period 50 – 350 E.C [Aklilu,1984].The 29 symbols in the Sabaean alphabet, that were in use in Northern highlands of Ethiopia about 2500 years ago, finally gave birth to Geez characters after undergoing some major changes in shape and direction[Bender et.al,1976]. Becoming the official language of Ethiopia both in writing and speaking, Geez took over 24 of the Sabaean symbols by undertaking some changes regarding their shapes (i.e. change in direction for example (to ሠ,(to ረ, (to ገ) and reduction and improvement of the appendages (e.g. Y to ሀ, X to ተ , and (to ዘ)[Bender et.al,1976]. In addition, the invention of two new symbols (ጸ and ፐ) to represent sounds of words borrowed from Latin and Greek has made the total number of symbols used in Geez 26.

One of the major breakthroughs in Geez over the Sabaean script is concerning the direction of writing – Geez writes from left to right while the Sabaean writes from right to left. Amharic writing system has also inherited the direction of writing from its ancestor – the Geez language (Bender et al. 1976; Aklilu, 1984). The other breakthrough is concerning order of alphabets; Geez alphabets are ordered as ሀ-ለ-ሐ-መ, while the Sabaean alphabets are arranged as አ-በ-ገ-ደ [Yonas et al, 1966 E.C; Million 2000].

Geez scripts have no vowel indications until around 350 A.D. Latter, however, vocalized consonant signs had come into being by undergoing a variety of changes in the structure of the consonantal symbols. The structural changes added six additional forms to each basic consonant increasing the total number of symbols to 182(26x7). Since then, vowels became an integral part of Ethiopic writing [Ullendorff, 1973; Million, 2000].

By the time Geez was replaced by Amharic, in addition to the 26 symbols that were used in the Geez language, it added symbols by deriving them from the already existing Geez alphabets.

ሸ From ሰ

ቸ From ተ

ኘ From ነ

ዠ From ዘ

ጀ From ደ

ፏ From ጠ

ኸ From ሀ

This increased the total number of fundamental characters used in Amharic handwriting system to 34; out of which 33 are core characters and 1 is a special character [Million, 2000].

and slows down typing, as well as making the Amharic word processor difficult to operate. The solution that got a wide acceptance among scholars was making the order all uniform in some way using the seven forms of the basic '□' to make vowels(Bender et al., 1976).

2.3 CHARACTERSTICS OF THE AMHARIC CHARACTER

Amharic writing system is often called syllabary rather than an alphabet because the seven orders of Amharic characters indicated above represent syllable combination consisting of consonant and following vowel. The non – basic forms (vocalization) are derived from the basic forms (consonants) by attaching small appendages (diacritic marks) to the right, left, top, or bottom in more or less regular modification. Some are formed by adding strokes, others by adding loops or other forms of differentiation to each core character. In particular the second order is constructed by adding a horizontal stroke at the middle of the right side of the base characters (for example ሀ ለ መ ቡ ሱ ...). Similarly, the third order is formed by adding horizontal stroke at the bottom of the right leg of the base character (e.g. ሂ ሊ ሚ ሲ ቲ ኪ ጂ). The fourth order is formed by elongating the right leg of the base character (for example (ላ ማ ጣ ሃ ሳ ሣ) and the fifth order is constructed from the base characters by adding a loop at the bottom of its right leg (ጤ ጰ ፏ ፔ) (Worku, 1997).

While the second, third, fourth, and fifth orders indicated above are formed according to patterns of great regularity, others, the sixth and the seventh are highly irregular (Bender et al., 1976) . The sixth order is constructed by adding a stroke, loop or other forms in either side of he base characters. Consider as an example the characters ህ ል ም ስ ር ብ.

In same way the seventh order is formed from the base characters by elongating the left leg or adding a loop at the top or right side. For instance, characters ሆ ሎ ቆ ሶ ሞ ዎ.

As compared to English scripts, the concepts of upper case and lower case characters are absent in Amharic writing system. In addition, a line of Amharic script lies at the same level, having no ascent and descent features.

Characters	Method of construction	examples
2nd order	Add a horizontal stroke at the middle of the right side of the base character	ሁ ሱ ሑ
3rd order	Add a horizontal stroke at the bottom of the right leg of the base character	ሚ ኒ ጂ ዚ
4th order	Elongate the right leg of a two or three leg base character	ሃ ላ ማ ሐ

	Add a diagonal stroke at the bottom of the leg of a one – leg base character	ጋ ታ ቻ ፓ
5th order	Add a ring at the bottom of the right leg of base character	ኤ ሌ ሔ ሜ
6th order	Highly irregular	ፅ ዝ ስ ብ
	Some characters bend their leg	ህ ሕ ጥ ት
	Some looped characters add horizontal stroke at their loop	ው ድ ጅ ጽ
7th order	Highly irregular	ሞ ዎ ጎ ዖ
	Shortening last leg (or the last two legs for characters that have three legs	ሶ ቦ ጦ ሐ
	Adding loop at the top right of the character	ሆ ሮ ቆ ሎ

Table 2.2 Methods of Order Formation in Amharic Writing System and Sample Characters.

CHAPTER THREE

OFFLINE HANDWRITING RECOGNITION SYSTEM

3.1 INTRODUCTION

Handwriting recognition is one of the several computerized handwriting analysis methods. It is the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation [Timar et. al, 2002]. Generally, Handwriting recognition and interpretation are processes whose objectives are to filter out the variation so as to determine the message [Plamondon, 2000]. It involves different tasks that vary from image capturing to post processing through segmentation and recognition.

There are two main approaches in handwriting recognition. They are: global approach and segmentation based approaches [Blumenstein and Verma, 2000]. The first approach entails the recognition of a word as a whole by the use of features identifying the global characteristics of the word and the second approach requires that the word be first segmented into letters. In the second approach, the letters are recognized individually and used to match up against particular words [Breuel, 2002; Blumenstein, 2000].

Segmentation, nevertheless, is the source of degradation in the performance of an OCR system. In an attempt to solve the problem of segmentation, some researchers have used the conventional, heuristic techniques for both character segmentation and recognition [Pandaya and Macy, 1996; Plamondon, 2000] where as others use heuristic techniques followed by ANN based method for the character recognition purpose [Blumenstein 2000].

To assuage the problem of segmentation, attempts were also made to use techniques which do not involve complex segmentation algorithms or techniques which do not use segmentation algorithms at all[Wen – Tsong Chen and Gader, 2000]. Lexicon – directed techniques are example of the techniques which do not involve complex segmentation algorithms. They were applied and successful results have been obtained for printed and cursive handwriting recognition [Plamondon and Srihari, 2000].

Recently, handwriting recognition (both on – line as well as offline) is making an efficient use of the representing power of HMMs(Hidden Markov Models) and the discrimination power of ANN(Artificial Neural Net Works)[Lallican et. al,2000]. The basic idea behind using HMMs is the property that handwriting can be interpreted as a left – right sequence of ink signals, analogous to the temporal sequence of wave pattern in speech recognition. Although HMMs are good in modeling temporal sequences, the usual

maximum likelihood training procedure gives them less discriminative power than Neural Networks trained with mean square error criterion. The latter are good in discriminating shapes from different classes but they do not model temporal data sequences very well. [Lallican et. al, 2000].

3.2 HANDWRITING AND ITS SURVIVAL

Handwriting has continued to persist as a means of communication and recording information in daily life even with the introduction of highly sophisticated communication technologies. Thus, recognition of handwriting has practical significance in the areas of postal address recognition, bank check recognition and hand filled form analysis etc [Plamondon et. al, 2000]. One of the possible reasons for its persistence is the convenience of paper and pens over the key boards and other data input hardware. This convenience enables it to persist even at the time of the advent of the state of the art digital technologies; rather being benefited than threatened.

The widespread acceptance of digital computers seemingly challenges the future of handwriting. Handwriting, in the current information era, nevertheless, has tremendously been changed by the advent of these technologies and others like type writers, printing press, and computers [Plamondon et al, 2000; Dereje 1999].

3.3 RECOGNITION, INTERPRETATION, AND IDENTIFICATION

There are several kinds of handwriting analyses techniques: handwriting recognition, interpretation, and identification. Handwriting recognition is the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation [Plamondon et at., 2000].For English orthography, as with many languages based on the Latin alphabet, this symbolic representation is typically the 8 bit ASCII representation of characters. Today, characters of most of the language of the world are representable in the form of 16 – bit UNICODE representation [Plamondon et al, 2000, Lallican, 2000 et. al, Million 2000].

Plamondon et. al, 2000 puts the difference between Handwriting interpretation and Handwriting identification as follows. Handwriting interpretation is the task of determining the meaning of body of handwritten text such as handwritten addresses whereas Handwriting identification is the task of determining the author of a sample of handwriting from a set of writers, assuming that the handwriting of each individual is unique. Another area in the study of handwriting recognition is signature verification. It is the task of determining whether or not the signature is that of a given person [Plamondon, 2000].

3.7 OFFLINE HANDWRITING RECOGNITION

The central tasks in offline handwriting recognition are character recognition and word recognition [Madhvanath and Govinderaju, 1996]. A necessary preliminary step to recognizing written language is the spatial issue of locating and registering the appropriate text when complex, two dimensional spatial layouts are employed – a task referred to as document analysis[Madhvanath and Govinderaju, 1996; Plamondon et al, 2000]. A typical OCR system consists of processing steps such as scanning and Thresholding, preprocessing such as noise removal, segmentation, feature extraction, recognition, and possibly post processing.

3.7.1 PREPROCESSING

It is necessary to perform several document analysis operations prior to recognizing text in scanned documents. Some of the common operations performed prior to recognition are: Thresholding(the task of converting a gray – scale image into a binary black – white image), noise removal(the extraction of the foreground textual matters by removing, say, textured background), line segmentation(the separation of individual lines of text), word segmentation(the isolation of textual word), and character

The nature and outputs of the preprocessing steps (Thresholding, Binarization, and segmentation) depends on the choice of feature extraction methods.

In order to recognize many variations of the same character, features that are invariant (features which have approximately the same values for samples of the same characters to certain transformations on the character) need to be used [Trier et. al, 1996]

In this regard, a representation method for recognition of handwritten character called Local line fitting (LLF) is reviewed in detail. It is suggested by Juan – Carlos Perez, Enrique Vidal, and Lourdes Sanchez from Universidad Ploitecnica de Valencia, Spain; they argue that this method, based on simple geometric operations, is very efficient and yields a relatively low dimensional and distortion invariant representation. The most interesting part of this method is that no preprocessing of the input is required. A black & white or gray – Pixel representation is directly used without thinning, contour following, Binarization etc. they believe, therefore, a high recognition speed can be achieved.

3.8.1 LOCAL LINE FITTING(LLF) IN FEATURE EXTRACTION

Line fitting or regression analysis is a statistical methodology to estimate the relationship (using a theoretical or an empirical model) of a response variable (dependent variable) to a set of predictor variable (independent variable) where the response variable denoted by Y and the predictor is represented by X. The simplest relation that could exist between two variables is linear ($y = a + bx$). One of the methods of determining the best line that fit the data, is to use eigen values and the other is the Least square method

The goal of the feature extraction is finding a set of parameters (features) that define the shape of the underlying character as precisely and uniquely as possible. The other important feature of a parameterization (feature extraction) method for providing the highest degree of noise immunity and a good generalization capability of the resulting system, according to Perez et. al, is the continuity of the representation. This means that similar characters must be mapped into similar representation.

The summarized three features that identify a good parameterization method are: precision, Uniqueness and continuity. On the other hand, the economy of the system, dependent on the number of output parameters and, to a lesser extent, on their range of resolution, is also a key factor for several reasons.

points(pixels) to the straight – line. They believe that this can be done in with the cost of O (n_i), where n_i is the number of pixels in the cell i.

The density of black pixels in the cell, relative to the total number of black pixels, is also computed for each cell. One of the features extracted by these researchers represents this density and the other features represent the fitted straight line.

$fi,1 = \frac{ni}{N}$ (feature one of the cell i) .....(3.8.2.4)

Where n_i is the number of black pixels in the cell (or the sum of grey values, if applicable) and N is the number of black Pixels for the whole character.

A line $y = a + bx$). is uniquely defined by two parameters: the slope b and the intercept a. In order to lure the advantage that it may give them in tolerating certain variations of patterns, Perez and his colleagues did not consider the intercept as a feature to be extracted. In this case invariants invariant to position or size is not necessary. Invariance to rotation is also unnecessary, as the orientation of the characters in document is usually fixed. Therefore the only invariance required is to deformation introduced by different writing styles, acquisition conditions, etc. this is the reason of overlapping the receptive fields or the cells.

an instance correctly and a failure if it does not predict the class of an instance correctly. Thus, the proportion of the errors made over the whole set of instances (called the error rate) could measure the performance of a classifier. Error rate on the training data (re – substitution error) might not determine the true error rate of the classifier [Witt and Frank, 2000].

Neural networks, especially the multi layer perceptrons trained by back propagation, are among the most popular and versatile forms of neural network classifiers [Pandaya and Macy, 1996]. Feed forward Multi Layer perceptron are used for both basic steps of recognition phases (Training and operational) by adapting the weights to reflect the problem domain and by keeping them constant (fixed) in the operational phase [Pandaya and Macy, 1996].

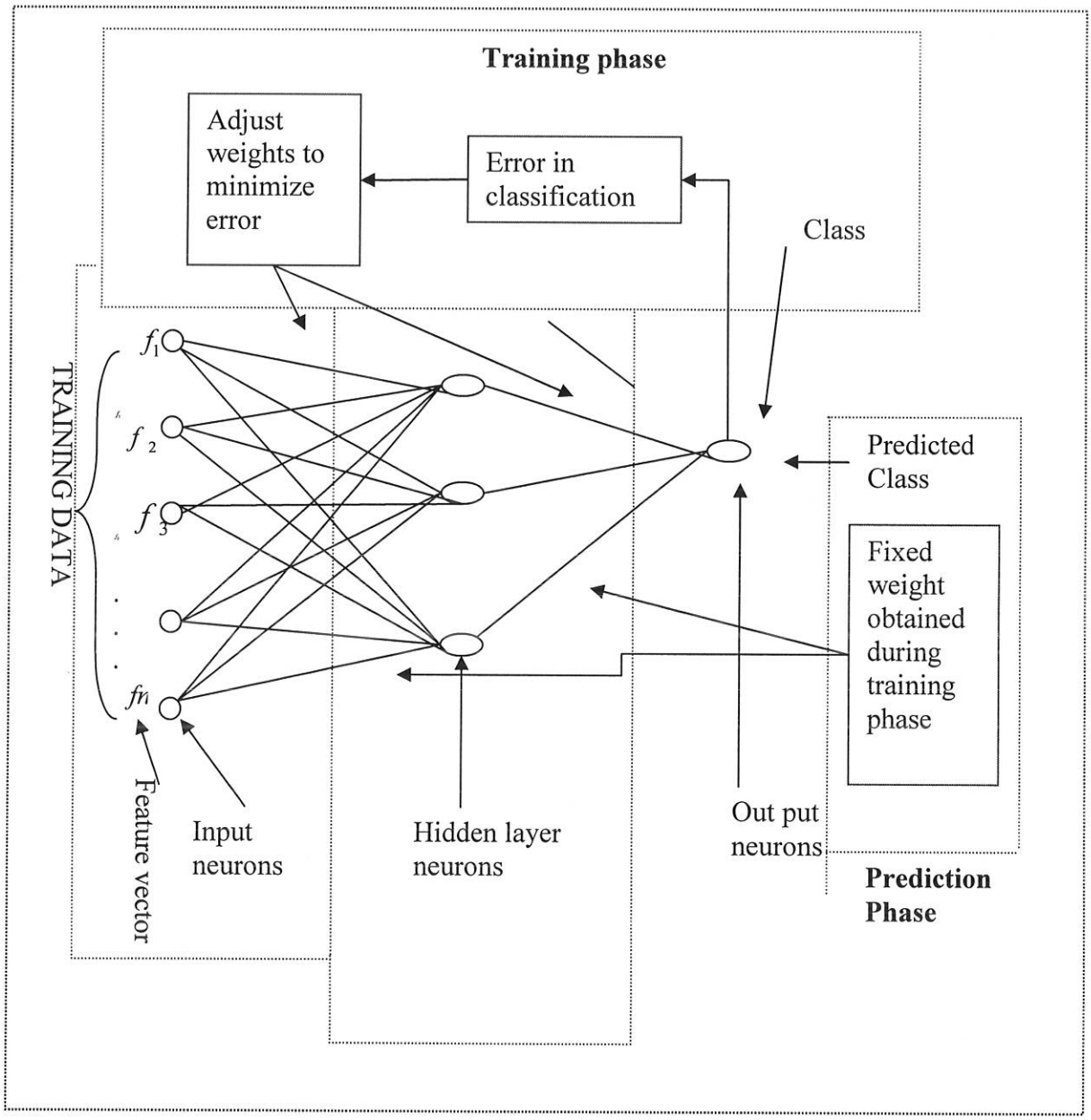

Fig 3.7.1. A Neural Network Architecture divided in to Layers and Phases

Back propagation learning algorithm, which is one and the simpler member of Gradient Descent algorithms, minimizes the difference (distance) between the desired and the actual output. For the back propagation training

algorithm, an error measure called Mean squared error is used[Weh Tsong and Gader, 2000]

$$E_p = \frac{1}{2}\sum_{j=1}^{n}(t_{pj} - o_{pj})^2 \quad \text{.....(3.7.1)}$$

(Note that E_p is the error for the pth presentation vector; t_{pj} is the desired value for the jth output neuron (i. e the training set value); and O_{pj} j jth output neuron and each sum is the error contribution of a single output neuron.

The minimum number of inputs required to successfully train the Neural Network increases exponentially with the dimensionality of the input space (a phenomenon called curse of dimensionality). The use of feature extraction techniques, is thus well justified [Pandaya and Macy, 1996; Trier, 1996].

In training where the amount of data for training and testing limited, training the classifier by holding certain amount of data for testing (holdout method) and using the rest for training is common (for instance quarter of the data for testing and three fourth for training). Cross validation is to fix the proportion or number of divisions to split the data in to and use one part for testing and the remaining for training in such a way that all the portions are

used for testing. The standard one is to use ten fold cross validation (use 90% of the data for training and 10% for testing to determine the error rate and do this ten times on each of the divisions and the final error rate would be the average of the ten error rates obtained) [Witt and Frank, 2000].

3.9.1 CHARACTER RECOGNITION USING ANN

The basic problem in recognition is to assign the digitized character to its symbolic class. In the case of printed image, this is referred to as optical character recognition. In the case of handprint, it is loosely referred to as intelligent character recognition (ICR) [Plamondon et. al, 2000; Weh Tsong and Gader, 2000].

Even though, methods in OCR have differed in the specific utilization of the constraints provided by the application domain, their underling core structure is the same. A typical methodology involves preprocessing, a possible segmentation phase (which could be avoided if global word features are used), recognition and post processing.

The methods of feature extraction are central to achieve high performing recognition in OCR systems. One approach utilizes the idea of "regular" and "singular" features. Handwriting is regarded as having a regular flow

modified by occasional singular embellishments (decorations) [Plamondon et al. 2000]. One of the approaches is to use HMM to structure the entire recognition process and the other method is to use a limited size of dynamic lexicon[Plamondon et al., 2000].One approach could also be the segmentation based recognition of words or names.

In segmentation based OCR systems, characters constitute the fundamental part as the smallest units to convey meaningful pattern. There are two major approaches in character recognition: the structure analysis and the statistical classification [Perez et. al; Mori et. al, 1999; Pandaya and Macy, 1996]. Some researchers in character recognition emphasized the structural analysis that was extracting of strokes of every character and deciding the attributes of characters and the relationship among them [Yaregal, 2002]. Structural approach to pattern recognition is based on primitives and their relation ships to recognize characters [Yaregal, 2002].

Researches showed that some of the problems with structural analysis include the difficulty level attached to the extraction of structures correctly, and difficulty in handling the effects of various noises that cause rather complicated variations of structures elements and their relation.

Other researchers, nevertheless, use straightforward template matching by directly comparing the input character image array with reference character image matrix. Additional discussion of template matching is found Trier et al, 1996.

CHAPTER FOUR

EXPERIMENTATION

4.1 INTRODUCTION

This chapter presents the result of experiment carried out to apply a statistical line fitting approach to extract features of handwritten Amharic characters that would be used in recognition of the characters used in Postal addresses writing. It also includes procedures, processes, and results from all the automation, Training, and testing of the system. Sample algorithms from the automation phase and sample training data from the training phase and confusion matrices from the testing phase are major components building up this chapter.

Result of a survey conducted to determined the most frequently used Amharic Characters in writing the Addresses used in this research is tabulated and presented(see table 4.1).

The Neural Network classifier was trained by using features extracted from 415 character images from the handwriting of three writers: 175 character images from writer D, 136 character images from writer A, and 104 character images from Writer Y). Then matrices of features were prepared for all the

three writers (49x175 for writer D, 49x136 for writer A, and 49x104 for writer Y). The training and the testing in this research were organized in such a way that the system is trained in one of the handwritings using a ten fold cross validation technique to evaluate its performance on the handwriting of single writer. Moreover, the system was also tested on the other three datasets individually in order to see how robust the system is and on the combination of the three writers to see the scalability of the system (see table 4.2).

4.2 DATA COLLECTION

The input of this Amharic Handwritten Character Recognition system is image of 196 handwritten destination addresses written in Amharic Language. They were collected from different address books of fellow students and friends and copied on A4 paper by the students themselves.

A survey has been conducted to determine the most frequently used characters in destination addresses written in Amharic in order to exclude the less frequently used characters from the training. According to the result found from the survey the most frequently used Amharic Characters were ል, ሰ, ን, ም, ር, ታ, አ, ስ, በ, ብ, ባ, ዲ, ራ

4.3 DESIGN OF AMHARIC CHARACTER RECOGNITION

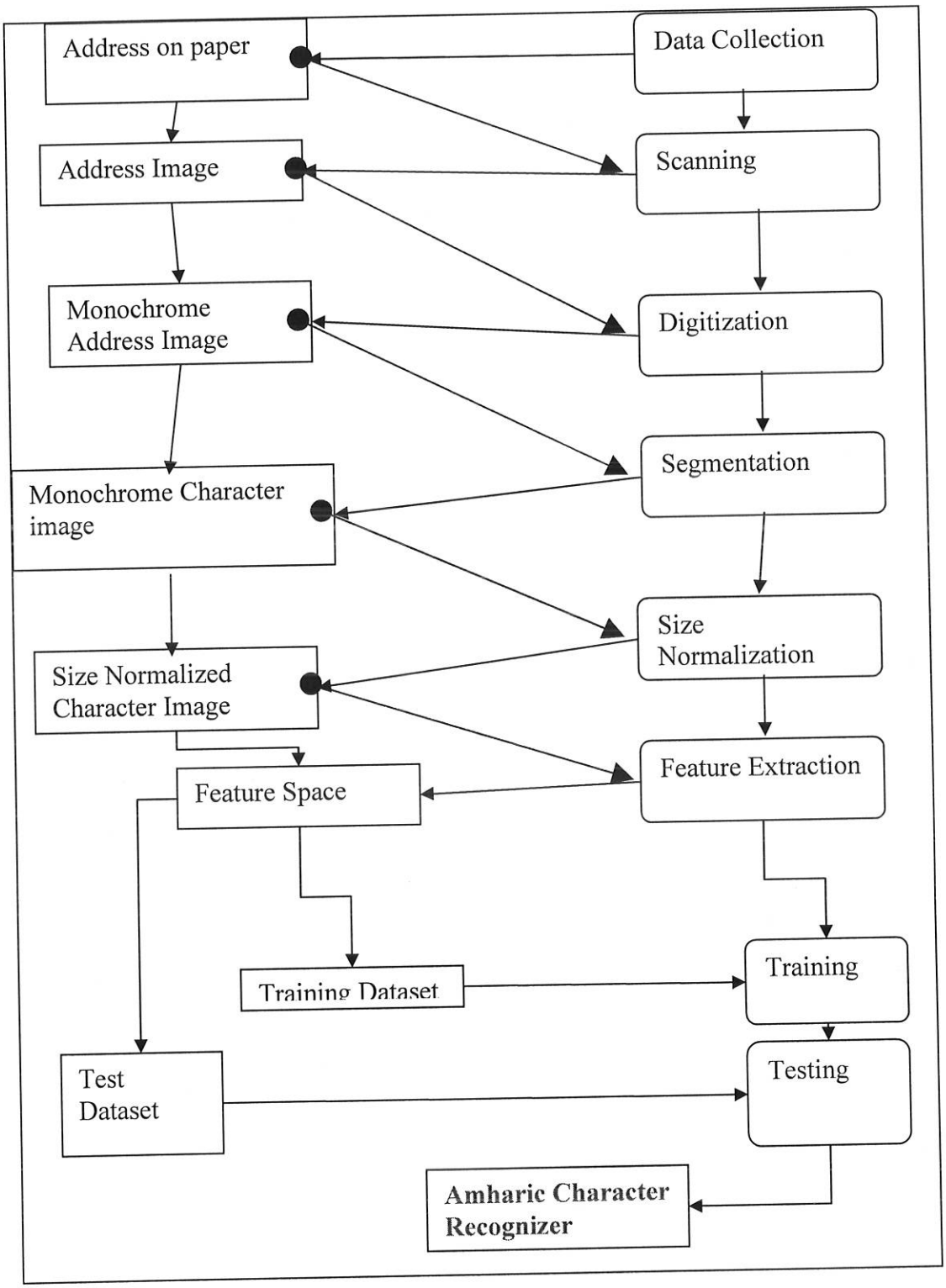

Fig 4.1 Design of the Amharic Character Recognition System

4.4 PREPROCESSING

The accuracy of recognition step in OCR systems highly depends on the effectiveness of their preprocessing steps. The goal of preprocessing steps, from the perspective of this research, is to reduce the noise (undesired artifacts) from the image data to some acceptable degree and prepare a refined image for further tasks in the recognition of characters. Some of the necessary analyses to perform prior to recognizing scanned image are: Thresholding (the task of converting a gray – scale image into a binary black – white image), noise removal (filtering out background textural matters, interfering strokes, shades and dots introduced due to input devices), line segmentation (the separation of individual lines of text), word segmentation (the isolation of textual word), and character segmentation (the isolation of individual characters)

The images of handwritten addresses used in this research are scanned in such a way that the out put would be black and white document image. HP 3500C scanner, used to scan all the address images at a resolution of 300 dpi, supplies a facility of saving the image in black and white and Microsoft paint accessory program, used to save and process the images, enables saving of the image as a Monochrome (black and white). Thus, Thresholding was not

considered for this recognition system because gray scale image of the addresses were not used.

Noise removal, however, was important due to some interference of the strokes of some of the characters into other characters. These interferences of strokes from neighboring characters were cropped by hand before the character image was considered for further treatment. Manually, additional adjustments on the sharpness and contrast of an image were made while scanning; the contrast was maximized so that the distinction between the foreground and the background would be unambiguous. Automation of noise removal is not in the scope of the research.

(a) (b) (c)

Fig 4.4. A character that has an interfering stroke to those characters written to the left. (b) Character whose stroke interferes to those written to the right. (c) Character that interfere to those written adjacent to it.

The method pursued was manual cleaning of the image by analyzing each image of addresses and eliminating some unnecessary strokes, dots, and black regions introduced due to the image capturing processes.

Thinning and Underline removal are also important Preprocessing steps in the development of OCR systems. Nevertheless, thinning was not considered because the features that would be extracted (pixel density, slope and intercept of line of regression) are proved to be invariant with the thickness of the strokes. (LLF: chapter three of this research).

4.5 DIGITIZATION

In order to process data collected on paper which is an analog medium, data should be converted to a digital image by the use of scanners. In this research, HP 3500C digital scanner is used to scan the addresses. The images were scanned by resolution of 300 dpi, contrast was made to be 100%, shadows were removed, and the output of the images was selected to be in black and white. Additionally, after the image is segmented into characters, it was made sure that the image is saved in a monochrome bitmap formats.

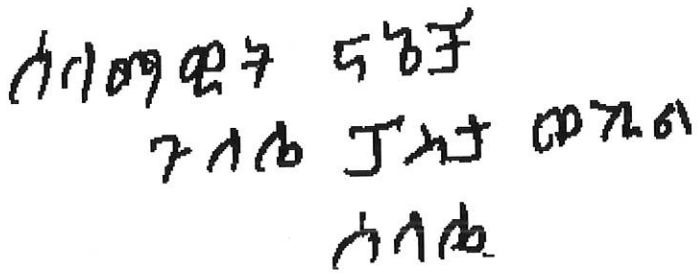

Fig 4.4.1 Sample Addresses as scanned and saved as black and white

4.6 SEGMENTATION

In order to recognize any document image, one of the approaches is to segment the document image into some manageable sub – images. In this research, segmentation algorithm that was used by Worku (1997), Ermias (1998), Dereje (1999), Million (2000), Nigussie (2000), and Yaregal (2002) was adopted. This algorithm was selected primarily because it was proved successful for printed Character recognition and secondly it is adapted well to the Amharic OCR systems [Yaregal, 2002] and it worked well for unconnected and non skewed characters [Nigussie, 2000].

In the algorithm, there are three main procedures: Line segmentation, Word segmentation, and Character segmentation. The algorithm mainly assumes a space between words and characters. The detailed code is appended on appendix I.

After the document image is segmented into character images, the next most important task was to determine the rectangular region containing the character. That is done in order to use the dimension for normalizing the image into a square region of (32X32 Pixels) and divide it to other square regions called cells (8X8 pixels) from which to extract the three basic features and determine the region to which the best fitting line is restricted.

	Trained using			
Tested using	Mr. D	Mr. A	Mr. Y	Combined
Mr. D	**89.7%**	2.3%	34.9%	88.6%
Mr. A	13.9%	**91.9%**	8.9%	84.7%
Mr. Y	9.6%	5.8%	**78.5%**	48.1%
Combined	44.3%	33.25%	36.9%	**73.25**

Table 4.6.1 Results of the Experiment(bold is result of cross validation)

The performance of the system using cross validation is sufficiently good. It also performed well when trained using the combination of the data and tested on individual writers. Further works are required to get explanations on the significant deference in recognizing handwriting of Mr. Y. From the inspection of the handwriting of Mr. Y, it is observed that his handwriting is abnormally small in size and highly irregular.

From the confusion matrix of the training Mr. A, it could be observable that 28 of the 31 'Be' characters were correctly classified as 'Be'(ন) where as one 'Be' is classified as 'B'(ন), one 'Be' (ন) character is classified as 'Aa" (ঁ) and one ' Be'(ন) is classified as 'S'(ঁ). Thus, the character wise recognition rate of

'Be' (ก) is 90.323%. (confusion matrices of the others is attached in appendix A)

The classifier was able to recognize 'B' (ก), 'Ba' (ก), 'S' (ก) and 'Ra' (ก) correctly. Given hereunder is the confusion matrix of the training copied from WEKA.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	I	j	<-- classified as
28	1	0	1	0	0	1	0	0	0	a = Be
0	16	0	0	0	0	0	0	0	0	b = B
0	0	7	0	0	0	0	0	0	0	c = Ba
2	0	0	10	0	0	0	0	0	1	d = Aa
0	0	0	0	4	0	0	0	0	0	e = Me
2	0	0	0	0	14	0	0	0	0	f = Di
0	0	0	0	0	0	26	0	0	0	g = S
0	0	0	0	0	1	0	11	0	0	h = Ta
0	0	0	1	0	0	0	1	5	0	i = R
0	0	0	4	0	0	0	0	0	0	j = Ra

with the statistical regression model (and the Least square method to model it) is the fitting of a local linear model to the black Pixels in a 8x8 square region. Here more attention is given to the feature extraction, training, and testing rather than the preprocessing of the image.

In previous researches done in the department on Amharic Character Recognition, low accuracy was obtained and network classifier was concluded to be not satisfactory [Nigussie, 2000]. To the contrary, in this research, however, a highly motivating result [91.9%] that would inspire further studies in the field was obtained. The system would be more versatile if sufficient training data was obtained on the classified characters since machine learning, due to the curse of dimensionality, requires a large amount of training dataset.

5.3 RECOMMENDATIONS

On the basis of the experiment and the constraints of this research, and due to the assumptions made about some concepts, the following recommendations were forwarded to improve the research.

- Further studies would be to determine the effect of using gray scale, under line removal, thinning, and skew detection and removal, slant detection and removal on the recognition using these set of features
- Further studies are possible by considering different size of cells, and / or overlapping cells

- The research is on Handwritten Amharic characters applied to the characters written on postal addresses by hand using pens (offline). Thus further works could be on the application of Local line fitting for feature extraction in other field of handwriting or in other areas of pattern recognition using other domains, colored pens, textured backgrounds, etc

- The robustness of this technique in handling noisy images introduced due to patterned and colored backgrounds needs further work

- It used only a linear model of regression to fit the distribution of foreground pixels in a cell. Using non linear models are recommended for further investigation

- Other classifiers were not tested using the features extracted by this technique. Hence, further works are encouraged on this line

- Machine learning approaches other than classification were not considered in this research. Thus, research works could be done on this area using the same technique of feature extraction methods

- Evaluation on this research was not made using evaluation data set thus repeating the same research with adequate amount of training, test, and evaluation data sets is also one area of research.

- The application of line fitting for extraction of global feature of words is not in the scope of this research. Thus, studies investigating the application of line fitting at global level are appreciated

- Analysis on the power of the features was not made (like principal component analysis) to determine which features are really good in maximizing inter – class variability and intra – class similarity. Therefore, by using the same technique of feature extraction, further works on such areas are recommended

- Impact of overlapping the cells on the performance of the system could be studied further

- Impact of size of handwriting, and different resolutions of scanning the image should be tested further

13. Lallican,M., Yong Haur Tay, Kahlid M., Guadian C. Knerr S. (2000). **Offline Handwritten Word Recognition Using a Hybrid Neural Network and Hidden Markov Model.** URL

14. Lecun, Y., L. Battou, Y. Bengio, P. Haffner(1998). **Gradient – Based Learning Applied to Document Recognition,** Proceedings of IEEE, vol.86, no 11, pp. 2278 -2324

15. Martin De Lesa (2001). **Document Image Binarization Based on Texture Features.** URL

16. Million Meshesha (2000). **A Generalized Approach to Optical Character Recognition.** (Masters Thesis). Addis Ababa: School Of Information Studies for Africa, Addis Ababa University.

17. Mori, S., H. Nishida and H. Yamada (1999). **Optical Character Recogntion.** New York: John Wiley & Sons, Inc.

18. Nigussie Taddesse (2000). **Handwritten Amharic Text Recognition applied to Bank Cheques.** (Masters Thesis). Addis Ababa: School Of Information Studies for Africa, Addis Ababa University

25. Srihari, S. and S. Lam (1996). **Character Recognition**: Amherst, NY: Center of Excellence for Document Analysis and Recognition, State University of New York at Buffalo. URL:

26. Srihari. S. N., **Recognition of Handwritten and Machine Printed Text for Postal Address Interpretation.** Pattern Recognition Letters vol. 14, 1993, pp. 577 – 584.

27. Tamhane, C. A., and Dunlop,D.D.(2000).**Statistics and Data Analysis.** Upper Saddle River: Princess Hall

28. Thomas M. Breuel (2002). **Segmentation of Handprinted Letter Strings Using Dynamic Programming Algorithm.** URL

29. Timar G., Karacs. K, and Rekeczky C. (2002). **Analogic Preprocessing and Segmentation Algorithms for Offline HandWriting Recognition. URL**

30. Ullenderoff, E.(1973). **The Ethiopians: An Introduction to the Country and People,** 3rd ed., London: Oxford University Press

31. Wen – Tsong Chen and Gader.P (2000). **A Word Level Discriminative Training for Handwritten Word Recognition. URL**

32. Witt, I. H. and Frank, E(2000).**Data Mining: Practical Machine Learning tools and Techniques with Java Implementaion.** San Diago: Academic press

33. Worku Alemu (1997). **The Application of OCR techniques to the Amharic Script,** (Masters Thesis). Addis Ababa: School Of Information Studies for Africa, Addis Ababa University.

34. Yaregal Assabie (2002).**Development of Versatile Character Recogntion System for Amharic Text.** (Masters Thesis). Addis Ababa: School Of Information Studies for Africa, Addis Ababa University.

35. Yonas A. et. al. (1966 E.C) **አማርኛ፡ ለኮሌጅ ደረጃ የተዘጋጀ.** College of Social Science: Addis Ababa University .

Appendix I

```
// AMHAddressRecognizerView.h : interface of the
   AMHAddressRecognizerView Aclass
/////////////////////////////////////////////////////////////////////////////
#if
   !defined(AFX_AMHADDRESSRECOGNIZERVIEW_H__0FB3422D_2B0F_
   4FA8_8B00_1D8B0BA2CBF7__INCLUDED_)
#define
   AFX_AMHADDRESSRECOGNIZERVIEW_H__0FB3422D_2B0F_4FA8_8B
   00_1D8B0BA2CBF7__INCLUDED_
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
class CAMHAddressRecognizerView : public CScrollView
{
protected: // create from serialization only
   CAMHAddressRecognizerView();
   DECLARE_DYNCREATE(CAMHAddressRecognizerView)
// Attributes
public:
   CAMHAddressRecognizerDoc* GetDocument();
// Operations
public:
```

```
    CAMHAddressRecognizerDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    CClientDC dc(this);
    if (m_initialized) {
    BITMAP bm;
    m_bitmap.GetBitmap(&bm);
    CDC dcImage;
    if (!dcImage.CreateCompatibleDC(pDC))
    return;
    if(dcImage.GetDeviceCaps(RC_STRETCHBLT)){
    dcImage.SetStretchBltMode(BLACKONWHITE);
    CBitmap* pOldBitmap = dcImage.SelectObject(&m_bitmap);
    pDC-
    >StretchBlt(0,0,64,64,&dcImage,0,0,bm.bmWidth,bm.bmHeight,
    SRCCOPY);
    dcImage.SelectObject(pOldBitmap);
    }
}
}
    // TODO: add draw code for native data here
void CAMHAddressRecognizerView::OnInitialUpdate()
{
    CScrollView::OnInitialUpdate();
```

```
    return (CAMHAddressRecognizerDoc*)m_pDocument;
}
#endif //_DEBUG
////////////////////////////////////////////////////////////////////////////
// CAMHAddressRecognizerView message handlers
void  CAMHAddressRecognizerView::OnUpdate(CView*  pSender,
    LPARAM lHint, CObject* pHint)
{
// TODO: Add your specialized code here and/or call the base class
m_loaded=FALSE;
m_initialized=FALSE;
}
void CAMHAddressRecognizerView::OnDisplay()
{
// TODO: Add your command handler code here
//Loads the scanned document for processing
CClientDC dc(this);
if(!m_loaded)
{
    if (!m_bitmap.LoadBitmap(IDB_BITMAP1))
        {
        AfxMessageBox("Cannot Open bitmap");
        return;
```

```
    CClientDC dc(this);
    CDC dcImage;
    if(!dcImage.CreateCompatibleDC(&dc))
    return;
//select the bitmap object
CBitmap* PBitmap = dcImage.SelectObject(&m_bitmap);
// extract the feature of a character that begins at loc
//count the total number of black pixels of character
N=countN(&dcImage);
if(N==0)N=1;// exception handler
i=0;
j=0;
// now for each character
//for each row of the character,
// make the feature file ready for input
if( (FtrFile = fopen( "c:\\characterFeatures.doc", "a" )) == NULL )
    wsprintf("the file could not be opened","%s");
//start on a new line
fprintf(FtrFile, "%c",'\n');
do
{
 //for each row
     do
```

```
{
// determine the number of black pixels in the character image
loc=(i,j);
// determine the number of black pixels in a cell
n=countn(&dcImage,loc);
if(n==0) n=1;      //exception handler
// determine the values for the least square method
sumx=sumOfIndependent(&dcImage,loc);//    sum    of    the
independent variables
if(sumx==0) sumx=1;// adjusting the sum
avgx= sumx/n;// average of the independent variables
sumy=  findSumOfDependent(&dcImage,loc);//  sum  of  the
dependent varibles
avgy=sumy/n;// average of the dependent varibles
// Sxy
sumxsumy=sumx*sumy;
// Sx2
sumOfxSquared= sumOfxsquared(&dcImage,loc);
//Sxx
sumxx=sumOfxSquared - (sumx*sumx/n);
if(sumxx==0) sumxx=1;
// the following is the applicattion of the Least square method
// to the Handwritten character recognition
```

```
// this function will segment the document image into characters
    bool TopLine, BottomLine,LeftLine,RightLine;
    int BlackPixelsInRow, BlackPixelsInColumn;
    int TopHorLine, BottomHorLine,LeftVerLine, RightVerLine;
    int i, j, q, r;
//  FtrFile=fopen("Featrure.doc","a");
        m_Black=0x00000000;
        m_White=0x00FFFFFF;
        m_Red=      0x000000FF;
    TopLine=TRUE;
    BottomLine=FALSE;
    for (j=0; j<=bm.bmHeight;j++)
    {
        BlackPixelsInRow=0;
        for(i=0;i<=bm.bmWidth;i++)
                if(pDC->GetPixel(i,j)==m_Black)
                        BlackPixelsInRow++;
        if(BlackPixelsInRow!=0) // Top Line Segmentation
        {
        TopHorLine = j;
        BottomLine=TRUE;
        TopLine=FALSE;
        }
```

```
    else if ((BlackPixelsInRow==0) && (BottomLine))// Bottom
Line Segmentation
    {
            BottomHorLine = j-1;
            TopLine=TRUE;
            BottomLine=FALSE;
            LeftLine=TRUE;
            RightLine=FALSE;
            for (q=0;q<=bm.bmWidth; q++)
            {
                    BlackPixelsInColumn=0;
                    for (r=TopHorLine; r<=BottomHorLine; r++)
                            if(pDC->GetPixel(q,r)==m_Black)
                                    BlackPixelsInColumn++;
                    if((BlackPixelsInColumn!=0)                    &&
(LeftLine))//Character segmentation from left side
                    {
                            LeftVerLine=q;
                            LeftLine=FALSE;
                            RightLine=TRUE;
                    }
                    else    if    ((BlackPixelsInColumn==0)    &&
(RightLine))//Character segmentation from right side
```

```
                    {
                        RightVerLine=q-1;
                        LeftLine=TRUE;
                        RightLine=FALSE;
                    }
                }
            }
        }
    }
    void CAMHAddressRecognizerView::markCharacter(CDC *pDC)
    {
        int i,j;
        //int CharTop,CharBottom,CharLeft,CharRight;
        CharLeft=0;
        CharRight=64;
        CharTop=0;
        CharBottom=64;
        CPoint loc;
            for(i=CharLeft;i<=CharRight; i++)
            {
                pDC->SetPixel(i,CharTop,RGB(0,255,255));
                pDC->SetPixel(i,CharBottom,m_Red);
            }
```

```
//DEL  n=countn(&dcImage,loc);
//DEL
//DEL
//DEL  sumx=sumOfIndependent(&dcImage,loc);
//DEL
//DEL  avgx= sumx/n;
//DEL  sumy= findSumOfDependent(&dcImage,loc);
//DEL  sumxsumy=sumx*sumy;
//DEL  sumxx=sumOfxsquared(&dcImage,loc)-(sumx*sumx/n);
//DEL
//DEL  // the following is the applicattion of the Least square method
//DEL  // b is the slope of the line of regression
//DEL  b=sumxsumy/sumxx;
//DEL  d= b*b+1;
//DEL  feature1=n/N;
//DEL  feature2=2*b/d;
//DEL  feature3=(1-b*b)/d;
//DEL
//DEL  //call a function that saves the features into a word file;
//DEL  //saveFeatures(feature1,feature2,feature3);
//DEL
//DEL
//DEL          }
```

```
//DEL }
//DEL
//DEL }

double CAMHAddressRecognizerView::countN(CDC *pDC)
{

int i,j;
double N=0;
        m_Black=0x00000000;
        m_White=0x00FFFFFF;
        m_Red=      0x000000FF;
CPoint loc;

for(i=0;i<=63;i++)
{
    for(j=0;j<=63;j++)
    {
        if(pDC->GetPixel(i,j)==m_Black)
        N++;
    }

}
```

```
return N;
}

double CAMHAddressRecognizerView::countn(CDC *pDC, CPoint
    loc)
{
// this function conts the number of black pixels in the cell
// that starts at loc
int k=0; int l=0;
int m=loc.x;
int n=loc.y;
double num=0;
m_Black=0x00000000;
m_White=0x00FFFFFF;
m_Red=      0x000000FF;
for(k=m;k<m+8;k++)
{
    for(l=n;l<n+8;l++)
    {
        if(pDC->GetPixel(k,l)==m_Black)
                num=num+1;
    }
```

```
}
return num;
}
double     CAMHAddressRecognizerView::sumOfIndependent(CDC
    *pDC, CPoint loc)
{
// this function adds the independent values
double sum=0;
int k,l;
int m=loc.x;
int n=loc.y;
        m_Black=0x00000000;
        m_White=0x00FFFFFF;
        m_Red=      0x000000FF;
// for each pixel in the cell
for(k=m;k<m+8;k++)
{
    for(l=n;l<n+8;l++)
    {
        // if the cell is black
        if(pDC->GetPixel(k,l)==m_Black)
        sum=sum+(k+1)%8;
    }
```

```
}

return sum;
}
double  CAMHAddressRecognizerView::findSumOfDependent(CDC
    *pDc, CPoint loc)
{
double sum=0;
int k,l;
int m=loc.x;
int n=loc.y;
for(k=m;k<m+8;k++)
{
    for(l=n;l<n+8;l++)
    {
        if(pDc->GetPixel(k,l)==m_Black)
                sum=sum+(l+1)%8;
    }
}
return sum;
}
```

```
double          CAMHAddressRecognizerView::sumOfxsquared(CDC
    *pDC, CPoint loc)
{
double sum=0;
int k,l;
int m=loc.x;
int n=loc.y;
m_Black=0x00000000;
m_White=0x00FFFFFF;
m_Red=      0x000000FF;
// for each pixel in the cell
for(k=m;k<m+8;k++)
{
    for(l=n;l<n+8;l++)
    {
        // if the cell is black
        if(pDC->GetPixel(k,l)==m_Black)
                sum=sum+((k+1)%8)*((k+1)%8);
    }
}
 return sum;
}
```