Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Network

A thesis submitted to the School of Electrical and Computer Engineering in partial fulfilment of the requirements for the Degree of Master of Science in Computer Engineering

By Siranesh Getu Endalamaw

March 2016

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Network

By
Siranesh Getu Endalamaw

Advisor: Menore Tekeba

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Network

By

Siranesh Getu Endalamaw

Approval By Board of Examiners

Dr. Yalemzewd Negash _____
Dean, School of Electrical signature
and Computer Engineering

Menore Tekeba _____
Advisor signature

_____ _____
Internal Examiner signature

_____ _____
External Examiner signature

Abstract

The recognition of handwritten documents, which aims at transforming written text into machine encoded text, is considered as one of the most challenging problems in the area of pattern recognition and an open research area. Especially ancient manuscripts, like Ethiopic Geez scripts, are different from the modern documents in various ways such as writing style, morphological structure, writing materials and so on. This brings the necessity to make research works on characetr recogntion of those scripts. Geez is one of the ancient languages which has been used as a liturgical language in Ethiopia. Manuscripts written using this language contains many unexplored content which is the base of the current Ethiopic scripts; however, only few researches have been done on these valuable documents.

A number of algorithms have been proposed for handwritten character recognition such as support vector machine, hidden Markov model, and neural network.In this research the design and implementation of character recognition system for ancient Ethiopic manuscript using deep neural network is presented. Deep learning, is employed and trained using a Restricted Boltzman Machine (RBM), a greedy layer-wise unsupervised training strategy.

The complete system employs image acquisition, preprocessing, character segmentation, and classification and recognition. Efficient and effective algorithms were selected and implemented in each step. A dataset was also prepared to train and test the system, which consists of 24 base characters of Geez alphabet with 100 frequencies. Overall, a recognition accuracy of 93.75 percent was obtained using 3 hidden layers with 300 neurons. Analysis of results obtained

from each step of the recognition process shows that the system can be extended and fine-tuned for practical application.

Key words: Ancient Ethiopic Manuscript, Handwritten Recognition, Preprocessing, segmentation, Deep Neural Network, Restricted Boltzmann Machine.

Declaration

I,the undersigned, certify that research work titled Ancient Ethiopic Manuscripts Recognition Using Deep Learning Artificial Neural Network is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources, it has been properly acknowledged.

Siranesh Getu Endalamaw signature ——————

Date of submission: March, 2016

Place: Addis Ababa

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

Advisor: Menore Tekeba signature ——————————

Acknowledgements

First of all I would like to thank the Almighty God for giving me the strength through all challenging thesis journey.

My deepest gratitude is to my advisor , Menore Tekeba, for his encouragement, guidance, support, and enthusiasm with his knowledge. His guidance helped me in all the time of research and writing of this thesis.

Mr. Enyachew Tamir has been always there to listen and give advice. I am deeply grateful to him for the long discussions that helped me sort out the technical details of my work and writing of this thesis.

I thank you my husband Fiker and my daughter Afomi for your patience to spend those painful days and nights, for supporting me throughout all my studies and help me in writing and editing the thesis.

Last, but not least, I would like to offer my regards and blessings to all my families, friends and of those who supported me in any respect during the completion of the thesis.

Table of Contents

List of Symbols

v	system input
ϵ	learning rate
σ	transfer function
b	bias function
f	frequency of black pixels
h	hidden layer
k	sensitivity parameters
ℓ	Number of layer
m	mean of black pixels
s	standard deviation
w	weight function
y	system output
Z	partition function

Abbreviations

ANN	**A**rtifical-**N**eural-**N**etwork
ASCI	**A**merican **S**tandard **C**ode for **I**nterchange
CD	**C**ontrastive **D**ivergence
CR	**C**haracter **R**ecognition
CNN	**C**onvolutional **N**eural **N**etwork
DNN	**D**eep **N**eural **N**etwork
HMM	**H**idden **M**arkov **M**odel
IGT	**I**terative **G**lobal **T**hresholding
LSTM	**L**ong **S**hort **T**erm **M**emory
MCDNN	**M**ulti **C**olumn **D**eep **N**eural **N**etwork
NALA	**N**ational **A**rchives and **L**ibrary **A**gency
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SVM	**S**upport **V**ector **M**achines

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Character recognition is the process of detecting and recognizing characters from input image and converting into American Standard Code for Information Interchange (ASCII) or other equivalent machine editable form. The input to this system can be handwritten text or printed text. A handwritten text recognition system aims at transforming large amount of documents, handwritten into machine encoded text. Technically, this system includes procedures like scanning documents or taking photograph using scanner or imaging device respectively, pre-processing to clear unwanted pixels, segmenting each character form scanned or imaged document, extracting features of a character and finaly recognizing characters by training the network using the prepared dataset. This contributes immensely to the advancement of automation process of information storage, processing and search or retrieval, and improves the interface between man and machine in numerous applications, which include, reading aid for blind, bank cheques and conversion of any hand written document into structural text form [1].

There are two types of handwriting recognition: off-line and on-line. Off-line recognition is performed on images of printed or handwritten text where as in online handwriting, the location of the pen- tip on a digitizing tablet surface is recorded at regular intervals, and the task is to map from the sequence of pen positions to the sequence of words [2]. The on-line methods of recognition

have been shown to be better than to their off-line counter parts in recognizing handwritten characters due to the temporal information available at time of writing. However, in the off-line systems, comparably high recognition accuracy levels could be achieved [1].

Handwritten document or text recognition is considered as one of the most challenging problems in the area of image processing by many researchers. Different algorithms and systems have been proposed and implemented in the area of off-line character recognition [1]. Offline recognition system follow holistic and segmentation based approaches [3]. The holistic approach is used in recognition of limited size vocabulary where global features extracted from the entire word image are considered. As the size of the vocabulary increases, the complexity of holistic based algorithms also increases and correspondingly the recognition rate decreases rapidly. The segmentation based strategies, on the other hand, employ bottom-up approaches, starting from the stroke or the character level and going towards producing a meaningful word. After segmentation the problem gets reduced to the recognition of simple isolated characters or strokes and hence the system can be employed for unlimited vocabulary. Many of the approaches proposed mainly focus on recognizing characters or words of specific language, which consists of its own specific features [3].

Offline methods consider machine printed text in the modern recognition system for various application. It also consider the already existing bulk of handwritten text, especially documents or books that are handwritten well before automatic printing started. These classes of writings are usually called ancient manuscripts and the text writings exist mostly written in ancient world languages. In Ethiopia, Geez is one of the world's ancient languages [4] and

enormous amounts of handwritten material are found in Churches, Monasteries, museums, libraries, and many places of Europe and America snitched by travelers and invaders. These scripts are mostly written in parchments, and consist of intrinsic values and unexplored content until today.

Geez language (the liturgical language of Ethiopia) is a syllabic language, which is rich with characters and words. This language contains 26 base characters with six orders. Handwritten words in Geez mostly consist of isolated characters, which are composed of unique and artistic strokes. The fact that shows in the language and the character richness and similarity in morphology of characters (e.g. "ሰ" and "ለ") in the language, together with the ill-posedness of the recognition problem itself, will pose a great challenge in the recognition of handwritten Geez text.

The purpose of automatic recognition of ancient texts is to convert texts stored in a paper or other media, e.g. codex (branna), magic scroll [5], to a standard encoding scheme representing the texts. Ancient Ethiopic handwriting are unique and different from the modern handwriting in many ways like writing style, size and shape of the character, document type, the background, writing materials, morphology of characters and so on. In addition, to use those ancient scripts the following problems should be addressed:

- Some of them are huge in size and small in numbers
- They are old and difficult to maintain and replace
- Some of them are kept in museums for only tourists
- Difficult and time consuming for searching words as a reference

- In some places there is handling problems and scripts will be lost

If those scripts documented in digital form the above problems can be minimized. Therefore there should be an efficient recognition system for those handwritings in different from the modern handwriting, to transfer to next generation as well as for document preservation, in well formatted structural text form. This well formatted structural text can be used also for further application, like text search, research, data mining, and for translation purpose. Therefore by considering the mentioned attributes and problems; efficient recognition system must be designed and implemented to transfer so many information like cultures, histories and educations to next generation in preserved and well organized way.

A number of algorithms have been proposed for handwritten character recognition such as support vector machine, hidden Markov model, and neural network. Neural networks, a biologically-inspired programming paradigm which enables a computer to learn from observational data. In recent years, neural networks are becoming popular to solve problems of classification with many features [6]. Deep learning, a powerful set of techniques for learning in neural networks, is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. It contains fast and reliable set of algorithms which are composed of multiple layers that increases the accuracy of the recognition.

Although many efforts have been made in the area of pattern recognition especially for ancient scripts, the recognition accuracy is not as such good. Very few researches have been done in Amharic (derived from Geez) recognition [7, 8].

Therefore great effort and special treatment is required to recognize ancient Ethiopic scripts, which is an active research area.

1.2 Problem Statement

The challenges in recognition of handwritten text mainly comes from cursiveness of handwriting, difficulty of detecting text lines, non-uniformity of spaces between characters and words, inconsistency of a writer, and variability in writing styles of different writers. However, several handwritten recognition are made for different languages such as Latin, Chinese, Japanese and Greek etc [3]. Researches on the recognition of Amharic characters have been made in different perspectives [3, 7, 8]. Even if they are few and not standardized yet they establish a great way for further research on Geez characters. In addition to the common problems pointed above, in Geez language there is character richness and similarity in morphology of characters which create a great challenge in the recognition of handwritten Geez text. Therefore Geez handwriting recognition is still an active research area that needs more investigation; particularly there is a large gap on the recognition of ancient scripts, which contains large amount of information like history, culture, customs, religious and other important values.

1.3 Objectives

General Objective

The general objective of this thesis is to design and implement character recognition system for ancient Ethiopic manuscripts (Geez) using deep learning artificial neural network(ANN).

Specific Objectives

- Study the general and basic step of character recognition system
- Understand the concept of ANN in general and deep learning neural network in particular
- To address training and testing character dataset
- Design the network parameters of deep learning neural network and its architecture
- Selecting best training algorithm based on literatures to train deep neural network
- We come up with an accurate and efficient system/algorithm that leverages unique features of ancient Ethiopic/Geez scripts (like structural or morphological, hierarchical and the like) that will be the first one for this application
- Implementation and developing a program to test the algorithms and make analysis of the result.

1.4 Methodology

The methodology employed consists of collecting image of ancient manuscripts from churches and museums, data acquisition using imaging devices, and system design for data processing. The important steps that we followed are listed below

Literature Review - Reading books, articles, research papers, proceedings and materials related to the subject matter which help us for selecting efficient

algorithms.

Data Acquisition - Collecting and scanning different ancient manuscripts from churches, museums and libraries for extracting characters for training and testing.

Preprocessing Data - Making appropriate input data for the recognition phase like filtering the noise, binarization, edge detection, etc.

System Modelling and Design - Design the Deep ANN used, selecting training algorithm and modelling the system using Deep ANN.

Implementation - Matlab software has been implemented for the proposed system recognition.

Interpretation and Discussion - Analysis and discussion on the results, problems faced recommendation and future works.

1.5 Scope of the thesis

An Ethiopic Ancient script includes many characters, numbers and different diacritics. They are aged and are not well preserved. The security, diversity of data locations and access problems make it very challenging to collect training data for on Geez characters. Therefore, in this thesis we focused only the 26 base Geez characters and undamaged scripts. Preparing the datasets for training and testing is very difficult task because there is no automatic and efficient character segmentation software so far that can handle Ethiopic ancient documents. Because of this, a segmentation system was devised. The data sets used in this research work have been prepared by using this segmentation system for all 2400 characters (100 characters for each base Geez characters) that was so difficult and routine. From the 26 base characters only 24 characters have rich presence in manuscripts and two characters ”ጸ” and ”ፐ” are not widely

found on the ancient Geez documents. Also, the research is limited to show performance of Deep Neural network (DNN) for this particular task; so complete system for processing and recognition is not done in this research work.

1.6 Thesis Organization

The organization of this thesis is summarized as follows: chapter two is about literature review on character recognition technology which has been done by different researchers. Chapter three is about Ancient Ethiopic Manuscripts recognition. Chapter four is devoted to system designing and algorithms on each phase (preprocessing, segmentation and classification) and also the implementation of the system. Chapter five concentrates on testing and evaluation of the result. And the final chapter (chapter six) is conclusion and recommendation.

Chapter 2

Literature Review

There are different ancient languages in the world with their own alphabet for writing. Books that are primed by those ancient languages reveal much information and technology to the current situation of our world. Especially in Ethiopia, Geez scripts contain many unexplored contents. One of the tasks in processing of these documents is recognition of texts so that they can be converted to forms that can easily be processed by or streal in machine. The advent of computing machines and the need for processing large volumes of data motivated research and development for automatic recognition of those ancient and up to date texts. On the other hand one language differs from another in writing styles, character shape, space, overlaps, and the connection of characters and also the material used to write. These problems are challenging many researchers in producing solution in converting to computer readable format. At present many techniques have been developed for language however, the developed techniques are not be comprehensive enough to address all handwritings and languages. Especially, the case of ancient Ethiopic manuscript processing is almost untouched .This section presents the pervious works on ancient script, modern scripts and finally Amharic, Ethiopian language that is derived from Geez, recognition using different techniques.

Basically handwritten character recognition (HCR) includes different stages. It starts by acquiring the input image, preprocessing, segmentation, feature extraction, classification and recognition. Depending on the type of script and

style of characters one of the mentioned stages may be done with training and without training. But all are very necessary and depends on the result of pervious phases.

2.1 Ancient Manuscript Recognition

In a research [9] feed forward back-propagation neural network is applied to a handwritten recognition of Pali cards of Buddhadasa Indapanno. The images are improved by contrast adjusting, gray scale converting, and noise removing. The characters in the improved images are then segmented using connected component labeling and projection profile. The features of each character are extracted by zoning method. The experimental results show that the proposed method yielded satisfying results but there are some characters that cannot be classified due to the similarity between characters, difference in the color of backgrounds, the thickness of characters, and the size of characters.

The work in [10] investigates ancient Slavonic manuscripts from the 11th century is investigated. They propose a binarization-free approach based on local descriptors to minimize the consequences of false character segmentation. Initially Scale Invariant Feature Transform (SIFT) features are extracted which are subsequently classified using Support Vector Machines (SVM). The system was evaluated on real world data, a dataset that consists of highly degraded Glagolitic characters. Experiments on this dataset proved the systems capability to recognize degraded characters and the difference to well preserved characters.

Another approach in [11] aims at supporting and facilitating current and fu-

ture efforts in Early Christian Greek manuscript digitization and processing. They propose a novel digital image binarization scheme for low quality historical documents allowing further content exploitation in an efficient way. Additionally, they present a novel methodology that assists recognition of early Christian Greek manuscripts. They strive toward an assessment of the recognition procedure by tracing and recognizing the most frequently appearing characters or character ligatures, using a segmentation-free, quick and efficient approach. Apart from the tools for processing the manuscripts, an innovative educational software tool has been developed to support the process of studying and translating the manuscripts by experts and students. From the result their focus was only on the closed cavities of old Greek handwritten i.e. the system couldn't recognize open and large size Greek letters. The recognition accuracy for characters with closed cavities is approximately 95%.

2.2 Modern Handwritten Recognition

The research in [1] focuses on recognition of English alphabet in a given scanned text document with the help of Neural Networks. Using Matlab Neural Network toolbox, they tried to recognize handwritten characters by projecting them on different sized grids. The steps they use in their work start acquiring scanned image followed by noise filtering, smoothing and normalization of scanned image and decomposing the scanned image into sub images. And finally they use character extraction and edge detection algorithm for training the neural network to classify and recognize the handwritten characters. After experimentation, they propose those artificial neural networks are used to perform character recognition due to their high noise tolerance. And the ability to get excellent result depends on selection of feature extraction algorithm.

The writers in [12] applied Convolutional Neural Networks (CNNs) for offline handwritten English character recognition. They used by modifying the common model of CNN which is LeNet-5CNN model, with special settings of the number of neurons in each layer and the connecting way between some layers. Experiments were done based on lower case and upper case section. These two sections contain 28069 samples for uppercase and 61351 samples for lowercase, respectively from UNIPEN dataset. In order to obtain offline character images, they employ some preprocessing steps like connecting the adjacent points, extending the width of strokes and anti-aliasing. For training of the CNN, an error-samples-based reinforcement learning strategy is developed. Experiments are evaluated on UNIPEN lowercase and uppercase datasets, with recognition rates of 93.7% for uppercase and 90.2% for lowercase, respectively.

In [13], researchers uses several multi column deep neural networks (MCDNN) to classify handwritten Chinese characters from the dataset used in their previous work. The data consists of plain images (offline, no temporal information) of isolated Chinese characters (already segmented out from text). From their result MCDNN signicantly improved the recognition rate over single DNN. And also they tried to speed up the recognition by testing their system on the GPU. Therefore Multi-Column Deep Neural Networks achieve best known recognition rates on Chinese characters, approaching human performance.

In [14] on the other hand, they analyze transfer learning with Deep Neural Networks (DNN) on various character recognition tasks. In particular, transfer learning from Latin letters to Chinese characters works as well as pre train-

ing a net with 1% of the classes of the Chinese character training task, despite the lower apparent complexity of Latin letters. Advantages of transfer learning include: less training time is needed to obtain good results, and much better results are obtained when only few labeled samples per class are available for the destination task. To do this their DNN consists of a succession of convolutional and max-pooling layers. It is a hierarchical feature extractor that maps raw pixel intensities of the input image into a feature vector to be classied by a few, they generally use 2 or 3, fully connected layers. All adjustable parameters are jointly optimized, minimizing the misclassication error over the training set. They use the GPU implementation of a DNN for experiments and they achieve promising results.

In [15] the author describes an offline Arabic handwriting recognition system based on recurrent neural networks. The system is trained directly on raw images, with no manual feature extraction, and therefore requires minimal changes to be used for languages with different alphabets. Their recognition system has three components: (1) multidimensional recurrent neural networks, and multidimensional Long Short-Term Memory (LSTM) in particular; (2) the Connectionist Temporal Classification output layer; and (3) the hierarchical structure. These components fits together to form a complete system. Their experiment is done using a data that consists of 32,492 images: 30,000 for training and 2,492 for validation and they achieve good result with better recognition rates.

2.3 Geez or Amharic Character Recognition

There are very few researches made on Ethiopic character recognition, and almost none on ancient Ethiopic scripts to the best of our knowledge. Facts show that this is potentially rich research area and a lot of works could be done on Ethiopic scripts.

The work in [8] presents writer-independent offline handwritten character recognition for Ethiopic script. The recognition is based on the characteristics of primitive strokes that make up characters. They classify characters based on the characteristics of their primitives, resulting in grouping of characters in a five-dimensional space. They developed a database to standardize the evaluation of research works on recognition of handwritten Ethiopic script. The database is grouped in to two: the first group is collected from 114 pages of Ethiopian Orthodox Church documents, and the second group contains documents from ordinary writers. The second group is also partitioned in to two subgroups: characters collected from various issues and participants writing each character in the Ethiopic alphabet three times. The total characters set in the first sub group is 379,800 character samples from 177 writers and 120,840 character samples from 152 participant in the second subgroups. But source of documents and characters that are extracted from the church are not mentioned so its difficult to compare with our work. On the other hand they use thinning of characters to get the primitive strokes of each character, which doest consider the morphological nature of the church scripts. Finally they use sequence matching and knowledge base system for recognition, the system does not need training because the knowledge base stores possibly occurring sequences of primitives and connectors for each handwritten character, and

shows promising result in each groups.

Another approach on Ethiopic scripts [16] tries to recognize offline handwritten Amharic words based on lexicon. The system computes directional fields of scanned handwritten documents, from which pseudo characters are segmented. They developed and proposed an algorithm for such character and word segmentation, and also script-independent text line detection tasks using direction field image. The system is tested by a database of unconstrained handwritten Amharic documents collected from various sources. They prepared the lexicon from words appearing in the collected database. Form their result, for good quality texts, they achieved a recognition rate of 87% and for poor quality texts, the recognition rate was 58%.

The other literature on this regard is the one mentioned in [3]. The researchers in this paper proposed unconstrained offline Amharic handwriting system that uses Hidden Markov Model (HMM) based on two approaches. The first approach builds word models from concatenated features of constituent characters and in the second method HMMs of constituent characters are concatenated to form word model. They use structural features of characters as the building blocks of the recognition system and the two methods for recognition strategies. In feature level concatenation method, training samples for a given word are generated from a character feature list which stores possibly occurring sample features for each character and this sample with the input word will be used for HMM initialization which sets a prototype for HMM of the word to be trained including its model topology, transition and output distribution parameters. In HMM level concatenation method, HMMs are built for each character from the stored sample features of characters, and the word

model is made up of the concatenation of the constituent character HMMs. Because there is no dataset of handwritten Ethiopic documents for Amharic languages the researchers collected a dataset of handwritten Amharic documents and tested their system. Their results show that feature-level concatenation method consistently performs better than HMM-level concatenation across different document qualities and varying sizes of training and test data. The limitation we can see in this paper is that they try to build word models, which make their system so complex, and possibly reduce accuracy in case of Ethiopic language, which consists of huge amount of words with complex set of word formation. In addition, their approach does not consider hierarchical nature of the Ethiopic syllabus, which could be well suited for neural network classifiers. Still another problem is that, they apply thinning for feature extraction. However, this approach looses inherent and unique artistic (image like) features of Ethiopic letters, which helps in uniquely identifying one character from another. In ancient manuscripts, the artistic features are observed to be, in addition to their inherent morphology, the results of writing style which uses pieces of sharpened reeds ”meqa- beer”.

2.4 Summary

In the research works revised in this chapter, character recognition system use different approaches: training based or non training approach, statistically and machine learning, or neural network approaches. In the recognition of ancient scripts different mechanism is applied to preserve the morphological nature of the scripts. It is also pointed that special care and treatment should be done not to lose the very important features of the characters because compared to modern handwriting they are different in writing style, shape, and

material used to write. They also proposed different algorithms for historical documents based on the nature of characters which is very helpful to our work. The second categories show modern handwritten recognition system with lot of techniques and tool with advanced technology under pattern recognition. The last one shows the level of Ethiopic scripts recognition with character and word recognition.

From the experimental result of the revised papers , CNN and HMM approaches showed good performance , DNN and MCDNN can be extended to Geez and other languages but feed forward back propagation and SVM showed poor performance and requires improvement. In general deep neural approach with deep layer and optimized unit of neurons can be applied in Geez character recognition and are expected to provide better result.

Chapter 3

Ancient Ethiopic Manuscripts Recognition

3.1 Ancient Ethiopic Manuscripts - Overview

Ethiopia has the most ancient tradition of written culture in sub-Saharan Africa. Until today old monasteries and churches, scattered all over the country, hold thousands of precious manuscripts. It is exceptional in that it had its own written tradition [5] in Geez, an indigenous Semitic language from a very early period, while Ethiopian Christian literature in Geez is unique in terms of quantity and quality of the works. There are collections of scripts that are written in geez, including biblical and liturgical texts, hagiographies, legal documents and local historical writings. All these documents are currently insufficiently researched and are in danger of disappearing because of improper storage conditions, fire and thefts. Therefore there should be a mechanism to save the remaining scripts and document analysis has to be done. The current technology leads to solve those problems in different approach.

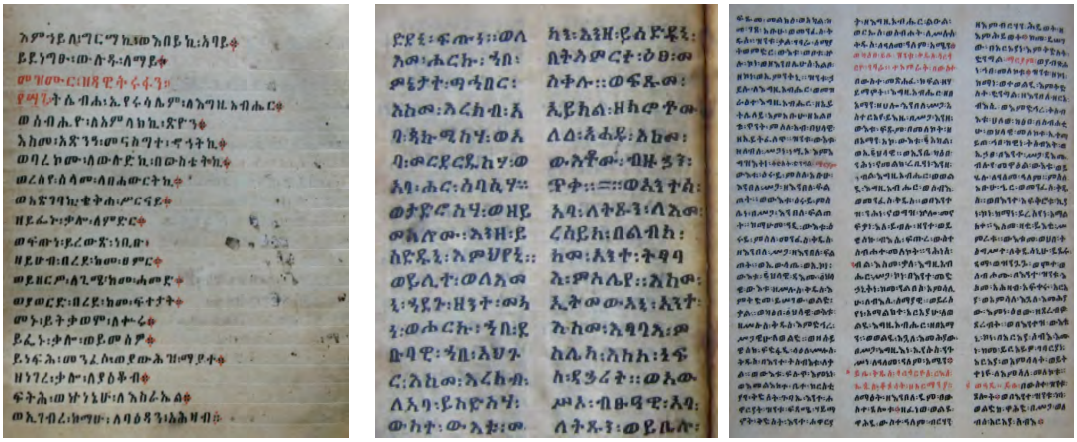
The common writing surface of ancient Geez is codex (branna), a parchment made from animal skin, or folded parchment leaves that are collected in gatherings, sewn together, and given covers. Because of its organic nature the codex is subject to degradation over long periods of time. The codex (branna) has dominated the local manuscript culture throughout its history. It is impossible today to establish the exact time when it was first introduced to Ethiopia, but the earliest known Ethiopian manuscripts are codices. The size of the codex varies greatly, depending on the time of preparation and the given text: from

pocket-size books to volumes more than 45 cm in height, so heavy that a grown man could hardly carry them. It was a fairly common practice to transfer aging text onto new branna in order to preserve the written word.

Usually Ethiopian ancient documents pages written in columns, each page containing one, two or three columns depending on the scripts. Some examples are show in Figure 3.1.

3.1.1 Geez Writing System

Geez is written with Ethiopic or the Geez abugida characters, a script that was originally developed specifically for this language. In languages that use it, such as Amharic and Tigrignya, the script is called Fidel (ፊደል) , which means script or alphabet. Geez is written/read from left to right.



(a) (b) (c)

Figure 3.1: Document pages written in: a) one column b) two columns c) three columns

There are essentially 26 main syllographs [4], all consonants, in Geez; while the rest are essentially those with additional strokes and modifications added on to the main forms to indicate a vowel sound associated with it or to make

aural adjustments in the basic consonant sound as shown in Figure 3.2. It must be acknowledged also that there are no upper or lower case distinctions in Geez. The total geez characters are 182 without geez numbers and diacritics.

Figure 3.2: Geez Characters

3.1.2 Characteristics of Geez Alphabet

In Geez characters there is a similarity between each Fidel morphologically which creates great challenges in recognition. Table 3.1 shows the characteristics of the basic geez characters

Table 3.1: Characteristics of Geez alphabet

Characters with one straight long leg	ቀተነገፐ
Characters with two straight long legs	ሰበከዘ
Characters with three straight long legs	ሐጠ
Characters with rounded bottom	ህመሠ
Characters with horizontal bottom line	ረፈ
Characters with rounded ring at upper	ጸጸ
Characters with full rounded ring	ዐፀ

3.1.3 The Difference between Ancient and Modern Ethiopic Scripts

Ancient Ethiopic script are unique by using its own written tradition in Geez and artistic nature of writing style. There are a lot of differences with the modern script that are written in Amharic, derived from Geez. We can describe their differences based on the following terms:

Writing materials - ancient scripts are written in parchments which is called branna, made from animal skin, and a sharpened reeds ”meqa” pens [5]. Where as the modern script used a plane paper with European pen.

Size and shape of characters - character shapes vary according to the scribe, neighboring characters and writing materials (type of ink varies their shape in ancient script) which results in a challenging recognition task.

Writing style - ancient writing uses art style to make the text attractive where as in modern writing free writing style is used.

Morphology of characters - the artistic nature and similarity in morphology of characters makes classification and recognition difficult.

Background noise - ancient scripts contain highly distributed background

noise because of their aged, ink speage, ink faded out and handling problem. Hence, ancient scripts need further preprocessing compared to modern scripts. **Binding** - the main binding and threads for binding are made from wood and guts respectively. This makes taking image of scripts through scanner difficult. Scanning helps to minimize skew problem.

3.2 Handwritten Character Recognition System

The task of handwritten character recognition system uses the following basic steps: image acquisition, preprocessing, segmentation, feature extraction, classification and recogntion [17]. Figure 3.3 shows those major components of handwritten character recognition system

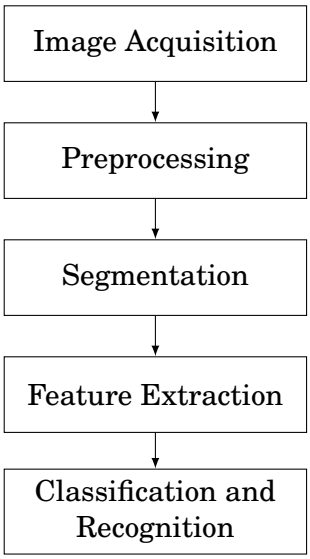


Figure 3.3: Block diagram of character recognition system

3.2.1 Image Acquisition

The input image can be taken through camera or scanner. The first stage in character recognition is image acquisition, which involves getting digital image of manuscript pages for input to the system. In places where scanning document is difficult, like ancient Ethiopic scripts, image acquisition is done through digital camera.

3.2.2 Preprocessing

Preprocessing involves operations performed on the scanned or photographed input image [18]. It essentially enhances the image rendering suitable for segmentation. The various tasks performed on the photographed image in preprocessing phase are described below:

Binarization :-Documents can be a valuable source of information but often they suffer degradation problems, especially in the case of historical documents [19], such as strains, background of big variations and uneven illumination, ink seepage, etc. All these are unwanted foreground elements and considered as noise. A sample image with background noise is shown in Figure 3.4. It removes the noise and improves the quality of the documents by converting the gray-scale document images to black and white (binary) ones. Based on some threshold value to decide whether a gray scale value of a pixel should be grouped as black and white. In the transformed binary image the background is represented by white pixels and the text by black ones.

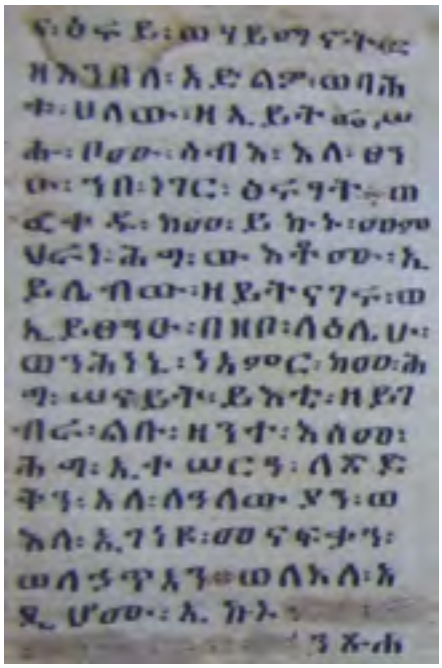


Figure 3.4: Sample document image with background noise

There are two general techniques for binarization based on the threshold values used [20, 21]: Global binarization and local binariztion thresholding. In global thresholding techniques, the pixels of the image are classified into text or background according to a global threshold that is determined by considering gray level pixel values of the entire image. Usually, such techniques are fast. This method fails to remove non uniformly distributed noise in the image. Local thresholding techniques, classify the pixel of the image into text or background according to a local threshold determined by their neighborhood area. Such techniques are more robust to the presence of different kinds of noise in the image. On the contrary, they are significantly more complex and time-consuming.

Existing binarization techniques focus either on finding an appropriate global threshold or adapting a local threshold of the noisy image in order to remove

smear, strains, uneven illumination etc. A technique for ancient or historical documents specifically is proposed in [19] and shows a very good result. This method is called hybrid binariztion technique. This approach first applies a global thresholding technique and, then, identifies the image areas that are more likely to still contain noise. Each of these areas is re-processed separately to achieve better quality of binarization. There is no algorithm that works well for all types of images but some work better than others for particular types of images suggesting that improved performance can be obtained by selection or combination of appropriate algorithm for the type of document image under investigation. This technique can be implemented with some modification to increase the quality of Ethiopic manuscripts for the next phase which is segmentation.

Skew detection and correction:- During scanning or taking the photo of documents, it often happens that the document page is not aligned correctly, the deviation of the dominant orientation of the text lines from the horizontal axis. The relative inclination angle of the page being acquired must be detected and accounted for as it can cause serious performance deterioration of segmentation and recognition stage of document processing system [22].

In literature, a variety of skew detection techniques are available and fall broadly into the following categories according to the basic approach they adopt: Projection profile, Fourier method, Nearest neighbor clustering, Correlation, and Hough transform technique [23]. Projection profile methods are the commonly used techniques, and usually work well for text -only documents. In the Fourier method, the direction for which the density of the Fourier space is the largest gives the skew angle. And very often for a document image, the largest

density direction of the Fourier space is on a vertical line and the true density direction may not be the largest. This makes the skew detection/search difficult. In Nearest neighbor clustering to skew detection all the connected components in the document are found and the directions of the nearest neighbor for each component. A histogram of the direction angle is computed, the peak of which indicates the document skew angle. On the other hand skew angle of an image using cross -correlation between lines at a fixed distance has been introduced [22]. It is based on the observation that the correlation between two vertical lines in an image of a skewed document is maximized in general if one line is shifted relatively to the other line such that the character base line levels for the two lines are coincident.

Skew problem can occur globally, i.e. the entire document can be shifted from the normal location during capturing the image or due to the position of the camera. On the other hand it can also be occurring during writing; this is local skewness problem. In Ethiopic manuscripts each line is observed to be parallel and there is no local skeweness problem because of the very careful writing style and tradition used. The global skew problem that occurs due to capturing the scripts through digital camera can be detected and corrected using Bounding Box Technique. The technique used is very simple and better compared to the methods listed above.The method does not involve any expensive computation like Hough transform and others to determine skew angles for documents.

Bounding Box technique [22] is a way of finding the extreme corners of text image. If the four extreme points are supposed to be found correctly forming a perfect rectangle, it can easily find out the estimated angle in much less time. The advantage of this Bounding box algorithm is that if any two of the four

corner points detected correctly, it will give the accurate skewed angle. Then skew correction is performed by rotating the document through an angle θ with respect to the horizontal line, where θ is the detected angle of skew. In order to prevent the image being rotated off the image plane, the skewed image is first translated to the center and the new image dimensions are computed.

Noise reduction:- The input image might be contaminated by additive noise and these low quality images will affect the next step of document processing. Due to this noise there can be disconnected line segments, large gaps between the lines etc. So it is very essential to remove all of these errors so that the required character features can be retrieved to the best possible way. There are many kinds of noise in images like Salt and Pepper Noise, the black points and white points sprinkled all over an image [18]. These can be reduced using filtering and applying morphological operations.

3.2.3 Segmentation

Segmentation of hand written text document into individual character or digit is an important and crucial phase in document analysis, character recognition and many other areas. In character recognition techniques segmentation is done to make the separation between the individual characters of an image, the most important process. Correct segmentation of individual symbols decides the accuracy of character recognition technique.

There are various factors that affects the process of text image segmentation [24]. The quality of the image is a significant factor for text segmentation. Presence of noise in the image results in degradation of accuracy and

efficiency. In Ancient Ethiopic scripts due to ageing the text is highly affected by background noise, drop of ink and handling problem. Therefore care must be taken to preserve the accuracy of the recognition system.

There are three steps in character segmentation: the first is line segmentation, separation of each line in the document. The second one is word segmentation- after identifying each line the word in each line is separated to make suitable for identifying individual characters. The last one is character segmentation, separation of individual characters form each word [25].

Line segmentation:-Line segmentation is the first and a preliminary step for handwritten text image segmentation, the process of identifying lines in a given image. It includes horizontal scanning of the image, pixel -row by pixel -row from left to right and top to bottom [24, 26]. Taking its top and bottom pixel rows that are identified during transformation of pixels from background to foreground and vice versa during scanning identifies a line.

Word segmentation:- Word segmentation is the next level of segmentation, at each line word is separated with a distance. It includes vertical scanning of the image, pixel -row by pixel -row from left to right and top to bottom [24]. At each pixel the intensity is tested.

Character segmentation:- Character segmentation is the final level for text based image segmentation. It is similar to in operations as word segmentation. Precaution should be taken under this event, i.e. after the scripts are preprocessed there might be opening of character that leads to segmentation of one character as multiple characters because of the gap between the open-

ings. This incorrect segmentation affects the recognition accuracy. Therefore to avoid this problem the open or unconnected character should be filled first using morphological operations dilation and erosion [25].

Several techniques for segmentation are reported in literature [25, 27]. These techniques may be classified into three groups:(i) Projection profile based or histogram projection techniques, (ii) Hough transform based techniques, and (iii) Thinning based approach.

Projection profile methods are of two types: vertical projection profile and horizontal projection profile. In the vertical projection method, the document is divided into vertical strips and vertical projection profile can be obtained by pixel by pixel along each value on the x axis for each y value. From this gaps between the lines can be observed. After finding the projection values, the curves can be smoothed by applying some filtering techniques. Horizontal projection gap between the lines can be found by searching the continuous white pixel and the line can be drawn. These methods work very fine on the unconnected handwritten and printed documents. In the case of ancient Ethiopic manuscripts, because the gap between each line and character are clear, this algorithms might yield better results.

Hough transform is employed in the field of document analysis in many research areas such as skew detection, slant detection, text line segmentation, etc. Thinning operation is also used by researchers for text line segmentation from documents [25].

3.2.4 Feature Extraction

Feature extraction is the process used to retrieve the most important data from the raw data. The most important data is the part in which the characters can be represented accurately. The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements [28]. Depending on the type of network used to recognize the characters, the feature of the whole segmented character can also be taken as a feature vector for classification.

3.2.5 Classification and Recognition

This stage is the decision making part of recognition system and it uses the features extracted in the previous stage or the segmented binary image. When input image is presented to HCR system, its features are extracted or the entire segmented binary image and given as an input to the trained classifier like artificial deep neural network. Classifiers compare the input feature with stored pattern and find out the best matching class for input. Classification is expected to perform one of the following two tasks [29]:

i. Supervised classification (e.g., discriminant analysis), where a given pattern has to be identified as a member of already known or predefined class. The system designer defines classes.
ii. Unsupervised classification (e.g., clustering), where a pattern needs to be assigned to a so far unknown class of patterns. Classes are learned based on the similarity of patterns.

3.3 Techniques of Handwritten Character Recognition

Character Recognition (CR) systems extensively use the methodologies of pattern recognition, which assigns an unknown sample into a predefined class. Numerous techniques for CR can be investigated in four general approaches of pattern recognition [30]: Template Matching, Statistical Techniques, Structural Techniques, and Neural Networks.

Template matching:- Template matching is a system prototype that is useful to recognize the character or alphabet by comparing two images of the alphabet. In other words it is the process of finding the location of a sub image called a template inside an image [18]. Once a number of corresponding templates is found their centers are used as corresponding points to determine the registration parameters. Template matching involves determining similarities between a given template and windows of the same size in an image and identifying the window that produces the highest similarity measure.

Classification is performed by comparing an input character image with a set of stored templates (or prototypes) from each character class. Each comparison results in a similarity measure between the input character and the template. The similarity measure, often a correlation, may be optimized based on the available training set.

Template matching approach is effective in some application domains; however it has a number of disadvantages. For instance, it will fail if the patterns are distorted due to the imaging process, view-point change, or large intra-class variations among the patterns [29]. Template matching can be extended in

to direct matching, deformable templates and elastic matching, and relaxation matching [18, 30].

Statistical Techniques:- Statistical method is concerned with statistical decision functions and a set of optimality criteria, which maximizes the probability of the observed pattern given the model of a certain class [30]. Mostly, there are three assumptions: the first one is distribution of the feature set is Gaussian or in the worst case uniform, the second one is there are sufficient statistics available for each class, and the last is given ensemble of images $\{I\}$, one is able to extract a set of features $\{f_i\} \in F, i \in \{1, \ldots, n\}$, which represents each distinct class of patterns.

The measurements taken from n-features of each word unit can be thought to represent an n-dimensional vector space and the vector, whose coordinates correspond to the measurements taken, represents the original word unit [30].

Generally, in applications involving patterns that can be represented meaningfully, using vector notations the statistical approach is ideal. However, this approach lacks a suitable formalism for handling pattern structures and their relationships citeoptical.

Structural Techniques:- Structural techniques are concerned with the recursive description of a complex pattern in terms of simpler patterns based on the shape of the object [30]. The characters are represented as the union of the structural primitives. It is assumed that the character primitives extracted from writing are quantifiable and one can find the relations among them.

Structural methods express characters as compositions of structural primitives such as lines, curves, and loops and then identify the characters by matching representations of its primitives with those of a reference character, or by parsing the representations according to sets of syntactic rules. These methods are more tolerant of irregularities than the statistical methods [29]. The difficulties associated with making the syntactic approach work for practical problems has been outlined as (i) It is difficult to identify a comprehensive set of primitive patterns; (ii) It is difficult to find a grammar which generates all and the only valid strings; (iii) It is not possible to utilize context in the well defined framework of context free grammars; and (iv)The syntax analysis is slow.

Neural Network Techniques:- Neural network is a computing system inspired from biological neurons and is defined as a computing architecture that consists of massively parallel interconnection of adaptive 'neural' processors. Because of its parallel nature, it can perform computations at a higher rate compared to the classical techniques. Because of its adaptive nature, it can adapt to changes in the data and learn the characteristics of input signal [30]. A neural network contains many nodes. The output from one node is fed to another one in the network and the final decision depends on the complex interaction of all nodes. Neural networks can be trained for any pattern recognition problem and they have been one of the most successful learning algorithms for character recognition. Section 3.4 presents detail discussion about neural networks and their application for ancient script recognition, focusing on approaches for training deep neural network architectures, which is a new area of machine learning.

3.4 Handwritten Character Recognition using Deep Neural Network

3.4.1 Artificial Neural Network (ANN)

A neural network is composed of a number of interconnected neurons, and the manner of interconnection differentiates the network models into feed forward networks, recurrent networks, self-organizing networks, and so on [31].

In neural networks, a neuron (processing element) is also called a unit or a node. A neuronal model has a set of connecting weights (corresponding to synapses in biological neurons), a summing unit, and an activation function, as shown in Figure 3.5. The output y of the summing unit is a weighted combination of input signals (features):

$$y = \sum_{i=1}^{d} w_i x_i + b \tag{3.1}$$

where w_i are the weights of each link between input nodes x_i with hidden nodes and b is a bias.

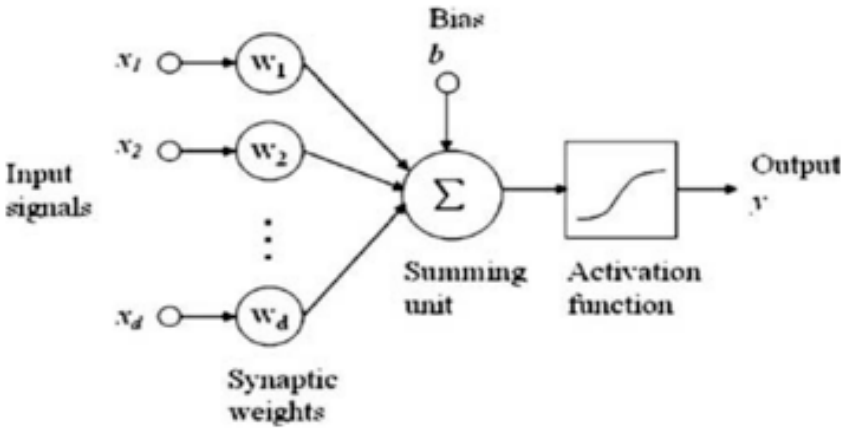


Figure 3.5: Model of a neuron

Architecture of neural network depends on nature and complexity of applications. However multilayer neural network with proper choice of parameter is capable enough to classify almost any pattern. The back propagation model or multi-layer perceptron, which is the most widely used model, is a neural network that utilizes a supervised learning technique applied for character recognition. In multi layer networks typically there are one or more layers of hidden nodes between the input and output nodes [17].

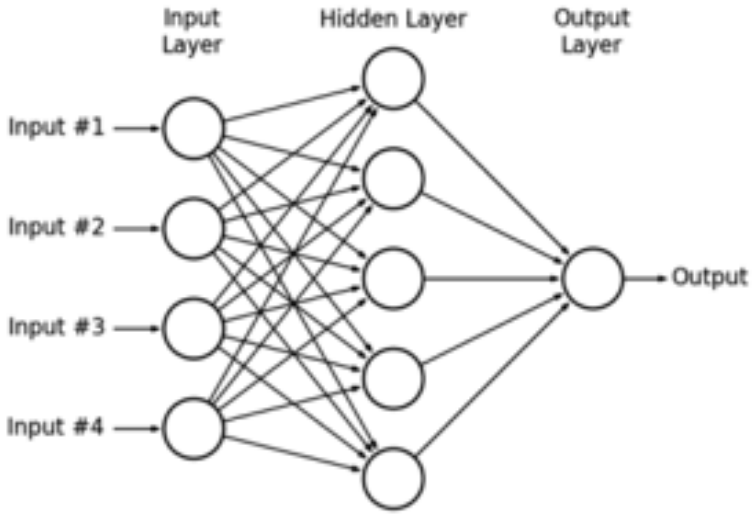


Figure 3.6: Architecture of multilayer neural networks

As shown on Figure 3.6 the network has an input layer (on the left) with four neurons, one hidden layer (in the middle) with five neurons and an output layer (on the right) with one neuron [31].

Input Layer:- A vector of predictor variable values ($x_1 \dots x_p$) is presented to the input layer. The input layer (or processing before the input layer) standardizes these values so that the range of each variable is -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the bias that is fed to each of the hidden layers; the bias is multiplied by a weight and added to the sum going into the neuron.

Hidden Layer :- Arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight (w_{ji}), and the resulting weighted values are added together producing a combined value u_j. The weighted sum (u_j) is fed into a transfer function, σ , which outputs a value h_j. The outputs from the hidden layer are distributed to the output layer.

Output Layer: Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight (w_{kj}), and the resulting weighted values are added together producing a combined value v_j. The weighted sum (v_j) is fed into a transfer function, σ, which outputs a value y_k. The y values are the outputs of the network.

There are so many parameters that control the performance of neural network [17], like

- Number of layers
- Number of neurons in each layer
- Transfer function used between two layers
- Learning algorithm
- Number of epochs

Training Artificial Neural Network:- Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins.

There are two approaches to training ANNs: supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by manually ”grading” the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help. The majority of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs [32].

3.4.2 Deep Learning Neural Network

Deep learning, a powerful set of techniques for learning in deep neural networks, is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers with complex structures, or otherwise composed of multiple non-linear transformations. An observation (e.g., an image) can be represented in many

ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc.. Some representations make it easier to learn tasks (e.g., face recognition or facial expression recognition). One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction [34].

There are a number of ways that the field of deep learning has been characterized. Deep learning is a class of machine learning algorithms that:

- use a cascade of many layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. The algorithms may be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised).
- are based on the (unsupervised) learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation.
- are parts of the broader machine learning field of learning representations of data.
- learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

Shallow architectures [33] have been shown effective in solving many simple or well-constrained problems, but their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications involving natural signals such as human speech, natural sound

and language, and natural image and visual scenes. However, deep learning algorithms are contrasted with shallow learning algorithms by the number of parameterized transformations a signal encounters as it propagates from the input layer to the output layer, where a parameterized transformation is a processing unit that has trainable parameters, such as weights and thresholds. They can be applied for large and complex real data such as for Ethiopic character data.

Deep learning is a name for a certain set of stacked neural networks composed of several layers. The layers are made of nodes. A node is a place where computation happens, loosely patterned on the human neuron, and firing when it encounters sufficient stimuli. It combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to it in the task the algorithm is trying to learn. These input-weight products are summed and the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal progresses further in the net to affect the ultimate outcome, say, an act of classification [34].

3.4.3 Training Deep Neural Network

Training deep multi-layered neural networks is known to be hard. The standard learning strategy consisting of randomly initializing the weights of the network. Applying gradient descent using back propagation is known empirically to find poor solutions for networks with 3 or more hidden layers. For that reason, artificial neural networks have been limited to one or two hidden layers [35]. However complexity theory of circuits strongly suggests that

deep architectures can be much more efficient (sometimes exponentially) than shallow architectures, in terms of computational elements and parameters required to represent some functions.

In the basic sense, a deep neural network contains an input layer and an output layer, separated byℓ layers of hidden units. Given an input sample clamped to the input layer, the other units of the network compute their values according to the activity of the units that they are connected to in the layers below. We will consider a particular sort of topology here, where the input layer is fully connected to the first hidden layer, which is fully connected to the second layer and so on up to the output layer. Before describing the learning and training mechanism, the mathematical notations used for representing DNN are explained as follows referring Figure 3.7.

Given an input x, the value of the j^{th} unit in the i^{th} layer is denoted by $h_j^i(x)$; if $i = 0$ it refers input layer and if $i = \ell + 1$ refers output layer. The size of a layer is denoted as $|h_i(x)|$. The default activation level is determined by the internal biasb_j^i of that unit. The set of weights W_{jk}^i between $h_k^{i-1}(x)$ in layer $i - 1$ and unit $h_j^i(x)$ in layer i determines the activation of unit $h_j^i(x)$ as follows:

$$h_j^i(x) = sigm(a_j^i) \tag{3.2}$$

where $a_j^i = b_j^i + k \sum w_{jk}^i h_k^{i-1}(x) \forall i \in \{1, \ldots, \ell\}$ with $h^0(x) = x$

and $sigm(a) = \frac{1}{1+e^{-a}}$ is sigmoid function. Given the last hidden layer, the output layer o(x) is computed similarly by

$$O(x) = h^{\ell+1}(x) = f(a^{\ell+1}(x)) \tag{3.3}$$

where $a^{\ell+1}(x) = b^{\ell+1} + W^{\ell+1}h^{\ell}(x)$

and $f(.)$ is the activation function, and for the j unit described as:

$$f_j(a) = softmax_j(a) = \frac{e^{a_j}}{\sum_{k=1}^{K} e^{a_k}} \tag{3.4}$$

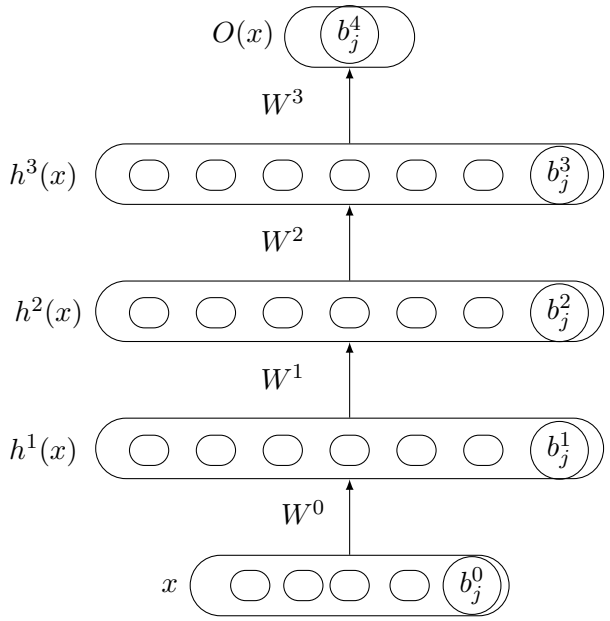


Figure 3.7: Illustration of a deep network and its parameters.

Training DNN as a conventional neural network has been found to perform worse than neural networks with one or two hidden layers. One reason for this is random initialization of weight might lead to gradient decent to get stuck in poor local minima. The other reason is when the number of hidden layers increased the cost function is going complex and it might produce many multiple local minima, which yields different solutions with gradient decent that might give low training errors but can have very different generalization errors.

In general, deep multi-layer neural networks have many levels of non-linearity allowing them to compactly represent highly non-linear and highly-varying functions. However, until recently it was not clear how to train such deep networks, since gradient-based optimization starting from random initialization appears to often get stuck in poor solutions. Hinton et al. recently introduced a greedy layer-wise unsupervised learning algorithm, a generative model with many layers of hidden causal variables [36]. Therefore the best paradigm to train DNNs is greedy layer wise approach using unsupervised learning and supervised fine tuning.

Greedy layer wise training is proposed to train a network taking one layer at a time, i.e. train layers sequentially starting from bottom (input) layer. Unsupervised training makes each layer learn a higher-level representation of the layer below. The greedy layer wise unsupervised strategy provides an initialization procedure, after which the neural network is fine-tuned to the global supervised objective. Figure 3.8 shows this greedy training procedure. The most common algorithms to train each layer in deep neural network using greedy layer wise unsupervised strategy are Restricted Boltzmann Machine (RBM) and auto-encoder

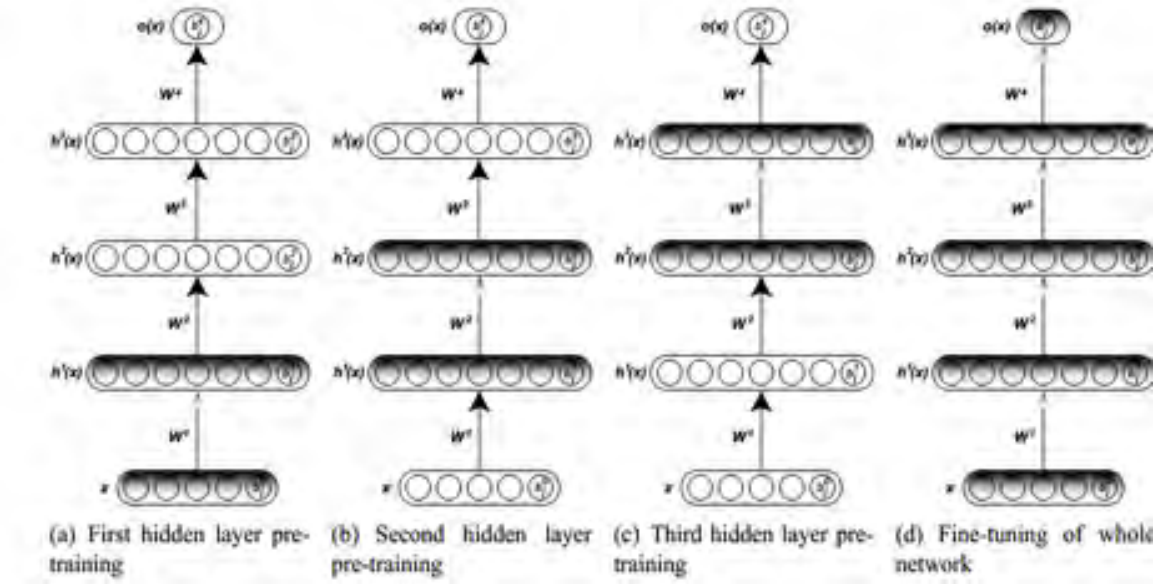


Figure 3.8: Unsupervised greedy layer-wise training procedure.

Restricted Boltzmann Machine is a generative model that uses a layer of binary variables to explain its input data [35, 37], undirected bipartite graphical model with connections between visible nodes and hidden nodes as shown on Figure 3.9 . The pixels correspond to visible units of the RBM because their states are observed; the feature detectors correspond to hidden units. A joint configuration, $(\mathbf{v}, \mathbf{h})$ of the visible and hidden units has an energy given by:

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \tag{3.5}$$

Z is called partition function and is given by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{3.6}$$

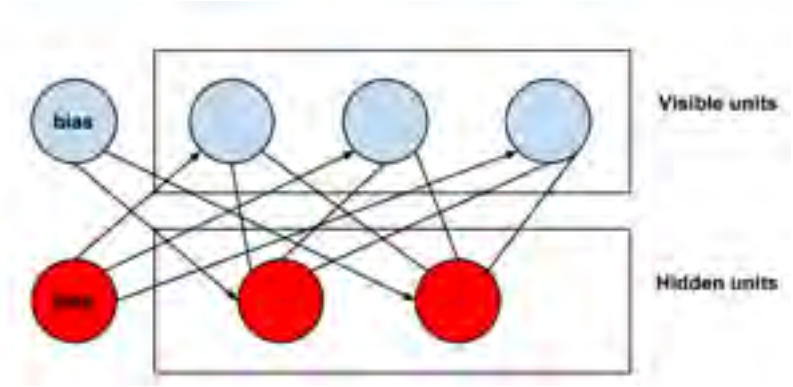


Figure 3.9: Illustration of RBM with three visible units and two hidden units (and biases).

The probability that the network assigns to a visible vector, $\mathbf{v}$, is given by summing over all possible hidden vectors:

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{h} e^{-E(\mathbf{v},\mathbf{h})} \tag{3.7}$$

The derivative of the log probability of a training vector with respect to a weight is surprisingly simple.

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{3.8}$$

This leads to a very simple learning rule for performing stochastic steepest ascent in the log probability of the training data:

$$\Delta w_{ij} = \in \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{3.9}$$

where $\in$ is a learning rate.

Given a randomly selected training image, $\mathbf{v}$, the binary state, h_j ,of each hid-

den unit, j , is set to 1 with probability

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i v_i w_{ij}) \tag{3.10}$$

where $\sigma(x)$ is the logistic sigmoid function $\frac{1}{(1+\exp(-x))}$. $v_i h_j$ is then an unbiased sample. Because there are no direct connections between visible units in an RBM, it is also very easy to get an unbiased sample of the state of a visible unit, given a hidden vector

$$p(v_j = 1|\ \mathbf{h}) = \sigma(a_i + \sum_j h_j w_{ij}) \tag{3.11}$$

Once binary states have been chosen for the hidden units,a reconstruction is produced by setting each v_i to 1 with a probability given by equation (3.11). The change in a weight is then given by

$$\Delta w_{ij} =\in (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recom} \tag{3.12}$$

This is the final weight using the reconstruction and the given data.
An auto-encoder or autoassociator is, another algorithm used to train DNN, a three layer neural network with unsupervised learning algorithm that applies back propagation, setting the target values to be equal to the inputs. I.e., it uses $y^{(i)} = x^{(i)}$.

An auto-encoder has two parts: an encoder function f that maps the input x to a representation $\mathbf{h} = f(x)$, and a decoder function g that maps h back in the space of x in order to reconstruct x. In the regular auto-encoder the reconstruction function $r(\mathbf{.}) = g(f(\mathbf{.}))$ is trained to minimize the average value of a recon-

struction loss on the training examples. Note that reconstruction loss should be high for most other input configurations. The regularization mechanism makes sure that reconstruction cannot be perfect everywhere, while minimizing the reconstruction loss at training examples digs a hole in reconstruction error where the density of training examples is large [38].

3.5 Summary

In this section the overall algorithms, methods and issues of handwritten character recognition system are discussed in detail. The input image is acquired through digital camera form different places and pre processed to eliminate unwanted pixels. In binerization of the gray scale image hybrid thresholding methods is best and recommended for ancient scripts and we expect better result for Ethiopic manuscripts. Skew detection and correction can be solved using the simple and less complexity algorithm which is bounding box method. Among the different algorithm of line and character segmentation vertical and horizontal histogram profile projection method is recommendable for the elegant Ethiopic manuscripts respectively. Deep neural network were generally found to be not better, and often worse, than neural networks with one or two hidden layers in reason that it was difficult to train but Hinton introduces a greedy layer-wise unsupervised learning algorithm, a generative model with many layers of hidden causal variables. Because of this DNN is very strong, and recommendable strategy for real and complex data than shallow networks and we expect also to get better result for ancient Ethiopic manuscripts.

Chapter 4

Design and Implementations

4.1 System Description

The proposed hand-written ancient Geez character recognition system involves training of a deep neural network using binary images of individual characters segmented from original document image. The trained network is then used to recognize character inputs in to a defined set of output classes (Geez characters).

The system accepts hand-written document images in colour (JPEG) format collected using digital camera. The colour image is then converted to gray-scale image, which is required for pre-processing which isolates foreground characters from background noise, and removes unwanted pixels form the important images. There are three sub processes in this stage that are applied on the gray-scale input image: binerization, skew detection and correction, and morphological operation. Binerization separates foreground (important pixels) of the image from the background and unwanted pixels. During taking the picture of the scripts there might be shift of the alignment form the original position, and also optical distortion. This problem is solved using skew detection and correction algorithms. The last step in pre-processing which is morphological operation that performs opening and clothing operation is applied to connect and fill opened texts.

In order to obtain individual characters which are inputs to the recognition

step, segmentation process is applied on the cleaned image. In the proposed system, this process involves line and character segmentation steps. In line and character segmentation, each line of the text is detected and segmented, and then individual characters are extracted from each segmented line. Before the characters are input to the training process, individual character images are labelled in to similar classes.

Finally, a deep neural network is trained, using a dataset prepared by us, to be used for recognizing character images into one of the Geez character alphabets. The output of the system is editable individual geez characters.

4.2 System Model

The proposed system includes four basic stages as shown in Figure 4.1 data collection or acquisition, preprocessing, segmentation, and recognition.

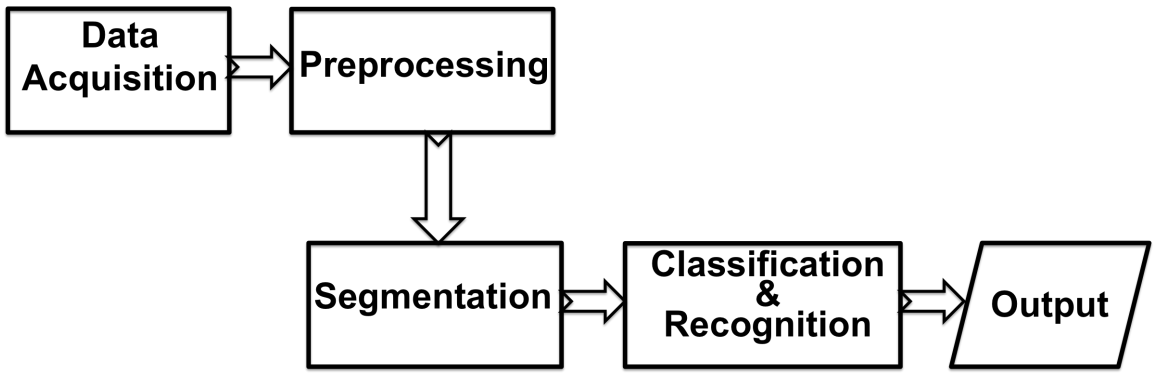


Figure 4.1: Block diagram of the proposed system.

4.2.1 Data Acquisition

The input image for the proposed system is collected through digital camera(see Appendix A) from different sources; Ethiopian Orthodox Tewahido Churches

(EOTC), monasteries and also from National Archives and Library Agency (NALA). We have collected images of twenty four different types of ancient manuscripts form NALA and six different types from churches and monasteries. The collected number of manuscripts and the number of pages in each manuscript are very limited because they are highly secured and it is difficult to take a picture of the entire manuscript pages even with legal permission. For security reasons, it is not allowed to take more than eight pages from one manuscript so that, on average five to eight pages were collected from each manuscript. Totally, 200 pages of ancient Ethiopic scripts were collected for preparing the data set to be used for training and testing.

4.2.2 Preprocessing

This stage is applied on the collected manuscript images to clean unwanted pixels and to remove background noises. There are four steps in this stage: gray scale conversion, binerization, skew detection and correction, and morphological operation.

Gray Scale Conversion:-The input images captured from digital camera, which are in RGB colour format are converted in to gray scale format using the luminosity or luma method. It converts RGB values to gray scale value (Luminance) by forming a weighted sum of the Red (R), Green (G), and Blue (B) components based on the following formula:

$$Luminance = 0.299 * R + 0.587 * G + 0.114 * B \tag{4.1}$$

Binerization:- Binerization involves converting gray scale document image in to black and white (binary), where the white and black pixels indicate the

background and text images respectively. Various binerization algorithms have been proposed for document image binerization(see chapter 3). The hybrid binerization technique is used in the proposed system because of its better performance, it is also recommended in various literatures for historical documents based on the good result it shows [19].

The hybrid approach attempts to combine the advantages of both global and local thresholding, first applies a global thresholding technique and, then, identifies the image areas that are more likely to still contain noise. Each of these areas is re-processed separately to achieve better quality of binarization. The proposed hybrid binerization framework is summarised as follows:

- Application of Iterative Global Thresholding (IGT) Algorithm,
- Detection of Noisy Areas (local), and
- Application of IGT Algorithm to the detected areas.

The output of the IGT Algorithm applied to the detected areas still contains some little background noise, which requires further cleaning. For this, additional step is implemented. This step combines the detected and cleaned noised area with the rest of the image and calculates the average black pixels of the entire image. The entire image is then thresholded using the average value obtained to get better binerized image.
The detailed steps of the algorithm is described below:

1. Read the gray scale input image represented by the following equation

$$I(x, y) = r, r \in [0, 1] \tag{4.2}$$

where x and y are the horizontal and vertical coordinates of the image, and r can take any value between 0 and 1 with 0 values representing a black pixel and 1 value a white pixel.

2. Calculate the average pixel value of the entire image, (Threshold T_I) for $M \times N$ document image using the following formula

$$T_i = \frac{\sum_x \sum_y I_i(x, y)}{M \times N} \tag{4.3}$$

where $I_i(x, y)$ is the image after the $(i-1)^{th}$ repetition.

3. Subtract T_i from each pixel and get an image Is

$$I_s(x, y) = I_i(x, y) - T_i + 1 \tag{4.4}$$

4. Stretch the gray scale histogram so that the remaining pixels are distributed in all the grey scale tones. The relation used for the histogram stretching is:

$$I_{i+1}(x, y) = 1 - \frac{1 - I_s(x, y)}{1 - E_i} \tag{4.5}$$

where E_i i is the minimum pixel value in the image I_s during the i^{th} repetition, just before the histogram stretching

5. Repeat steps 3-5 till the termination condition is fulfilled; the condition is given by:

$$|T_i - T_{i-1}| < 0.05 \tag{4.6}$$

Now the document image is binerized using IGT but still there might remain noise in certain areas of the image which is not cleaned by IGT. As a result, local thresholding is applied on the output of this image, as

described in the following steps.

6. Divide the image in to segments, each segment (S) of fixed size nxn window and calculate the frequency of black pixel in each segment. The segments that satisfy the following criterion are, then, selected as more noisy segments and are therefore candidates for further IGT:

$$f(S) > m + ks \tag{4.7}$$

where $f(S)$ is the frequency of the black pixels in the segment S while m and s are the mean and the standard deviation of the black pixel frequency of the entire page, respectively.

The parameter k in the formula determines the sensitivity of the detection method. The higher the k, the less segments will be detected. This could mean that some of the areas that may still need further improvement will not be selected. The window size (nxn) is also an important factor which affects the detection of noisy area. If it is so small, false alarm might be produced, i.e. segments that contain more character pixels than background pixels might be selected. Larger window size may lead to unfulfilment of the above criteria, which essentially means more background pixels than foreground.

7. Apply IGT (local thresholding) on the selected segment.

8. Combine the selected and cleaned image segments with the unselected image segments to form the binary image

9. Finally calculate the average pixel value of the entire binary image and

threshold the whole image to clean the remaining noise. This produces the final improved binary image.

Skew Detection and Correction:- Although ancient Ethiopic manuscripts are written with an elegant formatting, during image acquisition, the positioning of camera (if not fixed) might not align with placement of the manuscript, so the captured image might be skewed with some angle, which highly affects the line and character segmentation process. Skew detection and correction algorithms are used for addressing this issue. The Bounding Box Technique is used to detect and correct skew angle problem. The technique used is very simple compared to other methods mentioned in Chapter 3, since the method do not involve any expensive computation like Hough transform and others to determine skew angles for documents.

Bounding Box technique starts by finding the extreme corners of text image to determine a bounding box for the text only area, and then applies geometric transformation on the bounding box. The detailed steps of the algorithm are discussed below:

1. Read the number of rows and columns of the binary image
2. Find and mark the coordinates of the first and the last black pixel in the first row of the text image, as the two corners of the text image
3. Again find and mark the coordinates of the first and last black pixel in the last row of the text image, as the other two corners of the text image
4. Calculate the distance between the four corners forming rectangle and the diagonal distance taking the four extreme corners
5. Calculate skew angle from calculated distance

6. De-skew the image by rotating the black pixels (p) by ($-\theta$) using the following transformation equation.

$$Q = \begin{pmatrix} x' \\ \\ y' \end{pmatrix} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} . \begin{pmatrix} x \\ \\ y \end{pmatrix} \tag{4.8}$$

Where (x, y) are the coordinates of p in the input image and (x', y') are the coordinates of p in the output image.

Morphological Operation :- Morphology involves the process of images based on their shapes. The proposed system uses the most basic morphological operation that is dilation and erosion. Dilation adds pixel to the boundary of objects in an image while erosion removes pixels on object boundary. Binarization and skew correction produces opening of characters or unwanted connection between characters, this problem solved by using the two rules of morphological operation: opening and closing commands.

4.2.3 Segmentation

In this stage individual characters that are used for training are segmented using the cleaned binary image. Each line text of the binary image is extracted using horizontal projection profile method and then individual characters are segmented from line text image by using vertical projection profile method. The detail algorithm of each method is discussed below:

Line segmentation:- The horizontal projection method is used to compute sum of all black pixels on every row and construct corresponding histogram.

The steps for line segmentation are as follow:

1. Read the binary image
2. Count the black pixel in each row
3. Construct the Horizontal Histogram for the image
4. Using the Histogram, find the rows containing white pixel (space line).
5. Using the start and end line of white pixel mark the Bounding Box for text lines using standard Matlab functions (region props and rectangle).
6. Copy the pixels in Bounding Box and save in separate file.
7. Repeat the above steps to segment the whole line in the page

Character segmentation: The vertical projection method is used to compute the sum of all black pixels on a segmented line which forms one character and construct the corresponding histogram. The steps for characters segmentation are discussed below:

1. Read the segmented line image
2. Count the black pixel in each column
3. Construct the vertical histogram
4. Using the histogram, find the column containing white pixel which separates the characters.
5. Find the position containing single white pixel
6. Mark the Bounding Box for characters using the single white pixel
7. Copy the pixels in the Bounding Box and save in separate file.

4.2.4 Classification and Recognition

In this section the network architecture, the model parameters, and the training algorithm of deep neural network used for classifying and recognition of Geez characters in our proposed system is presented.

Network Architecture:- The proposed network architecture consists of a 900 input features each having binary values that are obtained from segmented character images normalized to 30x30 windows. The number of output nodes or units is determined by the number of unique classes, in our case the number of unique characters in Geez alphabet are 26. However, as will be described in section 5.2 we have only 24 class so the number of output nodes is made to be 24. In deep neural networks the number of hidden layers is hard to decide, in our system we will try experimentally to set it empirically that give the optimal result. The basic architecture of the network is shown in Figure 4.2.

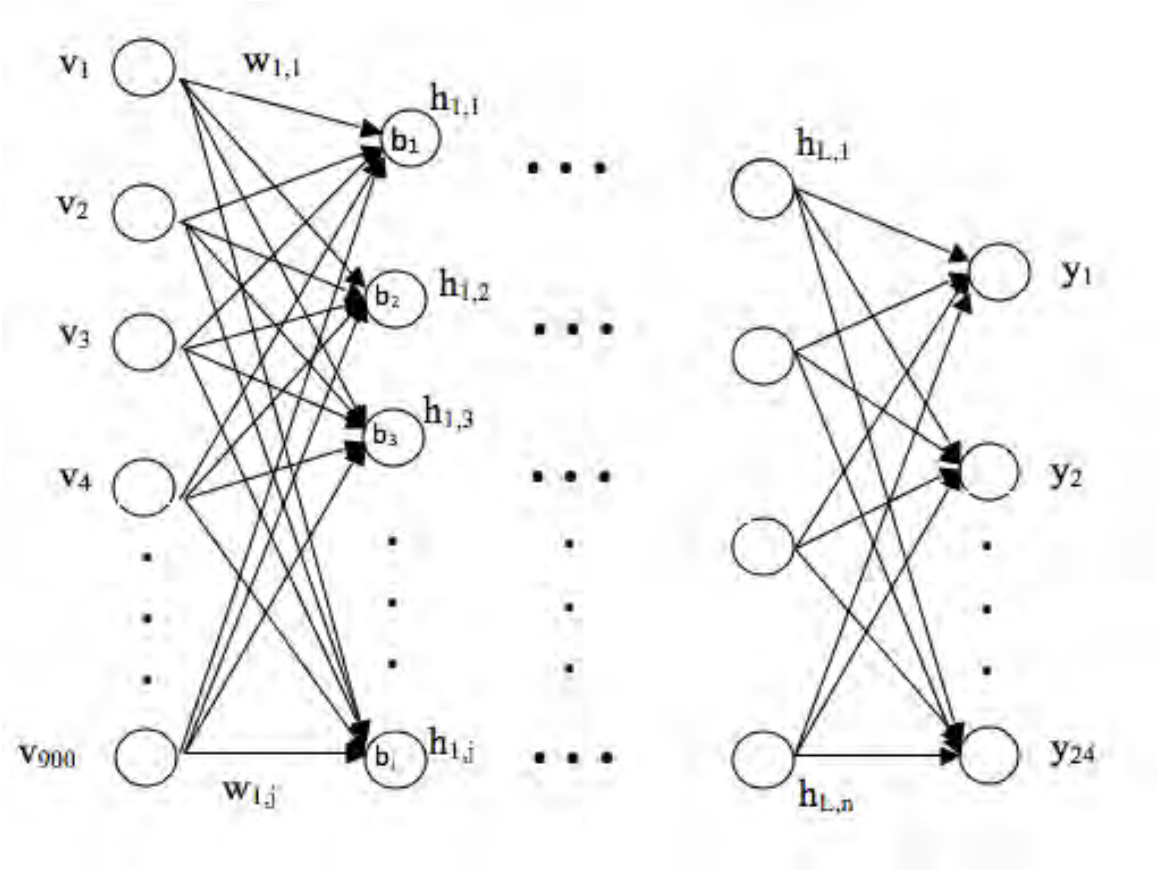


Figure 4.2: Architecture of the proposed deep network with 900 input, ℓ hidden, and 24 output layers

Model Parameters:- During designing of the neural network the number of different parameters of the network need to be decided. The model parameters [6] that are required for the proposed network are described below:

- Number of neurones in the hidden layer: number of processing unit or nodes in the hidden layer
- Learning rate: training parameter that controls the size of weight and bias changes in learning of the training algorithm.

  Recommended value: Real Domain: [0, 1] typical value: 0.3
- Batch size: the number of training instance per batch. The typical value depends on the training data.

- Momentum: momentum simply adds a fraction m of the previous weight update to the current one. The momentum parameter is used to prevent the system from converging to a local minimum or saddle point. A high momentum parameter can also help to increase the speed of convergence of the system. However, setting the momentum parameter too high can create a risk of overshooting the minimum, which can cause the system to become unstable. A momentum coefficient that is too low cannot reliably avoid local minima, and can also slow down the training of the system. Recommended value: Real Domain: [0, 1] typical value: 0.9

- Training epoch: when this value is zero it means train by epoch, and when the value is one means train by minimum error.
  Recommended value: integer Domain: [0, 1] typical value: 1

- Epoch: determines when training will stop once the number of iterations exceeds epochs. When training by minimum error, this represents the maximum number of iterations.
  Recommended value: Integer Domain: [1, ∞)

Training Algorithm of DNN:- The following steps are our procedure used to train deep neural network

1. Pre-training one layer at a time in a greedy way;

2. Using unsupervised learning at each layer in a way that preserves information from the input and disentangles factors of variation;

3. Fine-tuning the whole network with respect to the ultimate criterion of interest.

For the first one RBM are used for greedy pre-training step. RBM are trained to maximize the product of probabilities assigned to some training set V (a matrix, each row of which is treated as a visible vector v),

$$arg\max_{w}\prod_{v\in V}P(v) \tag{4.9}$$

or equivalently, to maximize the expected log probability of :

$$arg\max_{w}\mathbb{E}\left[\sum_{v\in V}\log P(v)\right] \tag{4.10}$$

where arg max is the argument of the maxima, the probability of the input vector is maximum at the given weight during training.

The algorithm most often used to train RBMs, that is, to optimize the weight vector , is the contrastive divergence (CD) algorithm as proposed by Hinton. The algorithm performs Gibbs sampling and is used inside a gradient descent procedure (similar to the way back propagation is used inside such a procedure when training feed forward neural nets) to compute weight update. The basic, single-step contrastive divergence (CD-1) procedure for a single sample can be summarized as follows:

1. Take a training sample v, compute the probabilities of the hidden units and sample a hidden activation vector h from this probability distribution.

2. Compute the outer product of v and h and call this the positive gradient.

3. From h, sample a reconstruction v' of the visible units, then resample the hidden activations h' from this. (Gibbs sampling step)

4. Compute the outer product of v' and h' and call this the negative gradient.

5. Let the weight update to $w_{i,j}$ be the positive gradient minus the negative gradient, times some learning rate:

$$w_{i,j} =\in (vh^T - v'h'^T) \tag{4.11}$$

The update rule for the biases a, b is defined analogously. For the second step using unsupervised learning each layer is stacked together in a way that by giving the output of the previous layer to the next layer by preserving every information form the input.

Finally the whole network is fine tuned in supervising manner using soft-max function criteria. The soft max activation function is useful predominately in the output layer [6]. Softmax function convert a raw value in to a posterior probability as pointed in section 3.4.

4.3 Implementation

This section describes the tools, languages and libraries used to implement the proposed system. The dataset preparation for the system input and the experimental set up are also presented in detail.

4.3.1 Developing tools

The proposed system to recognize ancient Ethiopic manuscript is implemented using the following description

1. Programming languages and tools:

(a) Matlab 2010b uses different function to prepare the dataset

(b) Matlab 2014a uses to integrate additional libraries to implement DNN

2. Libraries:

   DeepLearnToolbox: is a library for training and recognition of deep learning neural network.

3. Experiment setup:

   A laptop computer was used to conduct the experiment and digital camera for image acquisition. The specification is given in Table 4.1

Table 4.1: Experimental specification for the experiment

Process	Intel (R) Core (TM) i3-2350M CPU (2.3GHz, 3MB L3 cache)
RAM	3GB DDR3 Memory
HDD	500GB
Operating system	Window 7
Digital Camera	Sony, 16 MP

4.3.2 Dataset preparation

To train and test the proposed system we have prepared our own dataset. The collected input images are totally 200 pages. Form each pages character images were segmented and normalized in to 30 by 30 pixels. Our dataset contains 7065 characters that are extracted from the collected images. However the frequency of each character are not same, even there are characters which are not found in the dataset.

Because of the aforementioned reason, we have only used 100 samples for each base Geez character for training (70 percent) and testing (30 percent) which totals to 2,400 character samples. The remaining segmented characters which contain families of the base characters can be organized with additional segmented characters from additional sources to prepare more complete data-set for future research developments in this direction.

Chapter 5

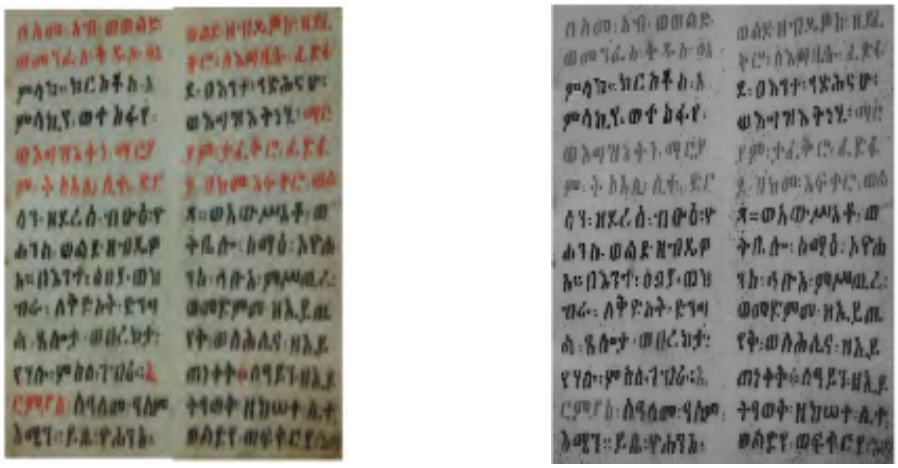
Results and Discussion

This chapter presents the results obtained from the experiments and analysis of the selected system of Ethiopic manuscript recognition. It was tried to run tests at each stage of the system using appropriate data derived from the manuscript images. In the first section, the results and discussion of preprocessing stage is presented. The second section shows the output of segmentation stage. The outputs of segmentation, which are individual characters, are then used to train the main component of the recognition system. A test scenario was used to verify the operation of the selected system. This test result which shows the classification of characters and recognition of Ethiopic manuscripts are also discussed at the third section.

To the best of our knowledge, since there is no other research result in ancient Ethiopic manuscripts, it was very difficult to present the comparison of the selected system with other recognition algorithms. Discussion and analysis of results are mostly based on percentage measures, and to some extent comparison with results of well known algorithms are made wherever applicable.

5.1 Results of preprocessing

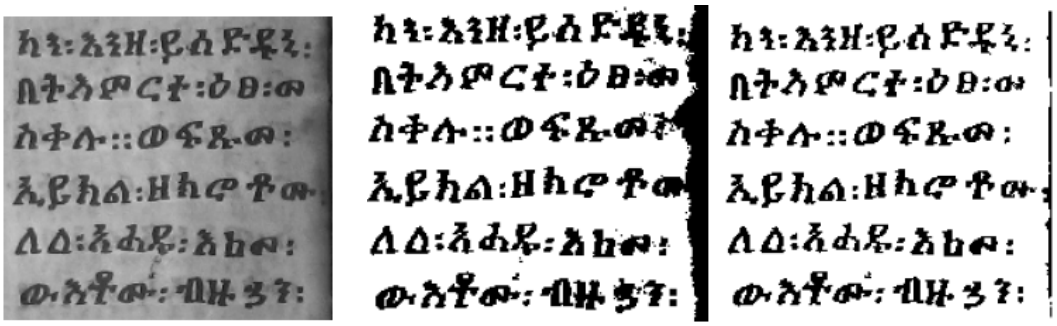
The first step in preprocessing is converting the input color image in to gray scale image. We used the Matlab function rgb2gray to get the gray scale image. The result of this function is shown in Figure 5.1



(a) (b)

Figure 5.1: Output of RGB to gray scale conversion: (a) color image (b) gray scale image

The gray scale image is then converted in to binary image in binarization step using the selected technique, hybrid binarization algorithm. Since there is no ground truth data for our test image, we only used subjective evaluation and we try to compare the output of the most commonly used binarization algorithm that is implemented in Matlab (Global Otsu binarization) with hybrid binarization. As shown in figure 5.2, the Otsu technique introduced some background noise, but the hybrid system tried to remove almost all of the background noise. The shapes of the characters are almost similar in each case, which shows that the proposed system performs comparably better, particularly in removing noise, which influences the segmentation stage.



(a) (b) (c)

Figure 5.2: Output of binarization: (a) gray scale image (b) binarization using Otsu (c) binarization using hybrid

The next stage of preprocessing under test was skew detection and correction. Ethiopic manuscripts are written in elegant format, by making straight lines as base line for writing each line of text on white codex, so it can be said that this does not create local skew problem. The only skew problem detected is during taking picture of Ethiopic manuscripts using digital camera, this is global skew problem. It was tried to take image of manuscripts without creating this skew problem, however there are some inclinations in some pages, and rotating with negative angle using bounding box algorithm solves this. In many document analysis problems, skew angles as high as 23° are observed. Even though we did not get skewed document as high as this, in order to test the algorithm we increased the skew angle of a document to 23° and the result is shown in Figure 5.3.

(a) (b)

Figure 5.3: Output of skew detection: (a) skewed image (b) deskewed image

In handwritten character recognition, preprocessing has a great effect on the accuracy of recognition. In the selected system we have tried to implement it carefully, to obtain a good result to make the recognition better and efficient (see Appendix B).

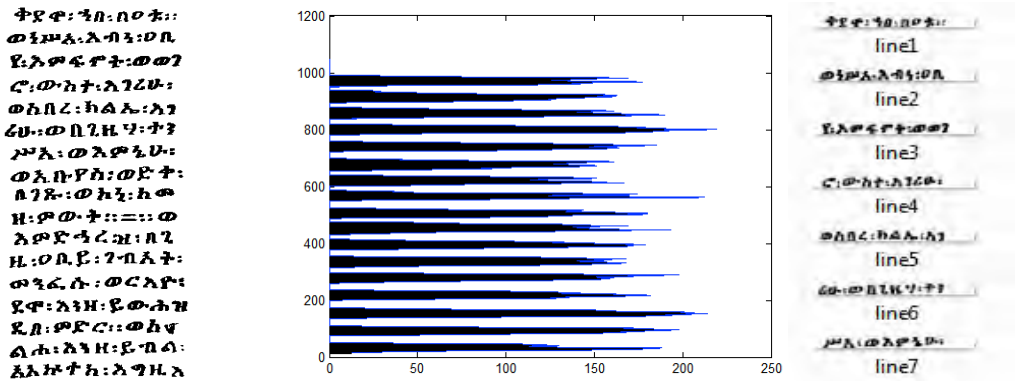
5.2 Result of Segmentation

For segmenting lines and characters in images, vertical (Y histogram) and horizontal (X histogram) projections is implemented respectively.

5.2.1 Result of Line Segmentation

The result of vertical projection for a sample page is shown in Figure 5.4. As shown in the figure, the lines of the sample images are segmented accurately. The algorithm was tested with a number of document pages, and all tests produced perfect results; this shows that the selected algorithm works very well.

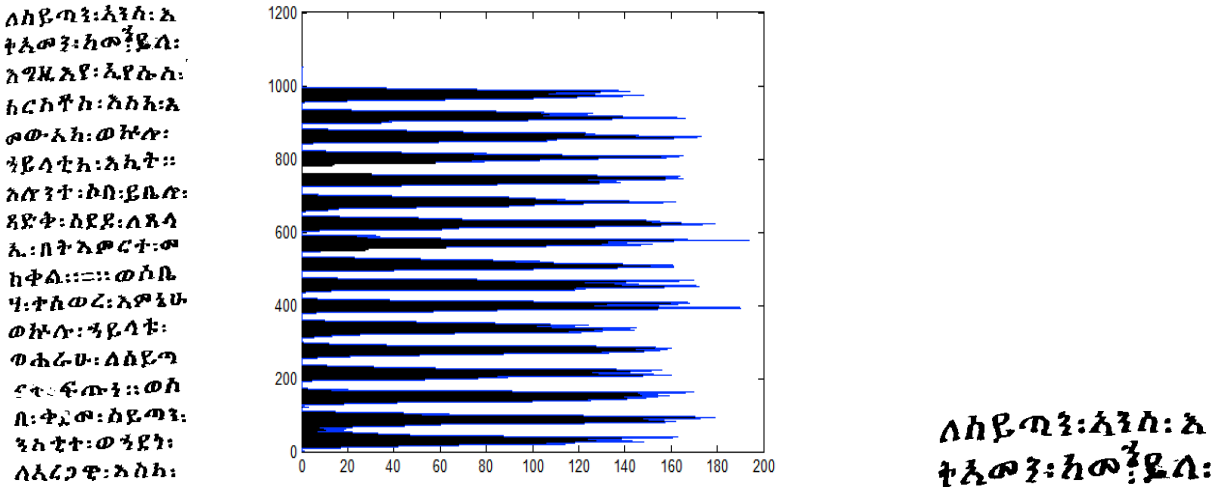
The processing speed was also observed to be fast and good for handwritten Ethiopic manuscripts.For example the processing speed which takes to segment the text image shown on Figure 5.4(a)is 0.202610 seconds.



(a) (b) (c)

Figure 5.4: Output of line segmentation (a) input image (b) Y histogram (c) segmented line

One limitation of line segmentation observed was that it segments two lines together if there are characters written over a word, between two lines, usually placed as correction when mistakes were made during writing, as shown in the first two lines of a document page and lower part of its histogram Figure 5.5 (a) and (b) respectively. In this case, the algorithm will segment the two lines incorrectly as a single line, as shown Figure 5.5 (c). it is hard to address such a problem using automatic segmentation, and manual intervention was used to address the situation; in fact it occurs very rarely, and the manual intervention has minimal overhead.

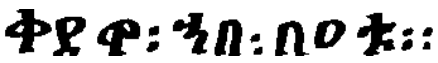


(a) (b) (c)

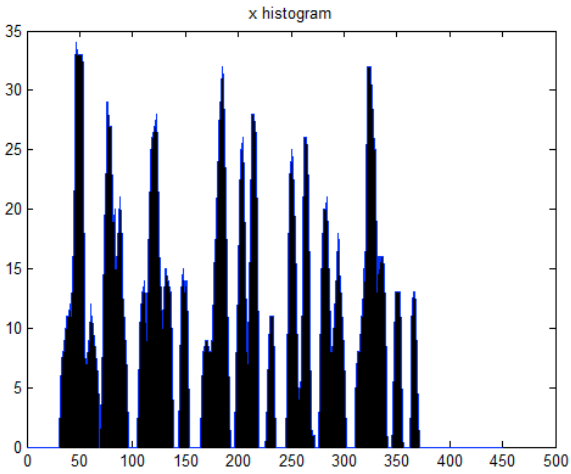
Figure 5.5: Line segmentation error (a) input image (b) Y histogram (c) error segmented

5.2.2 Result of Character Segmentation

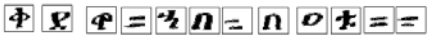
Character segmentation of the segmented lines is done using X histogram. From the given segmented line, each of the characters are extracted automatically using the X histogram. Figure 5.6 (c) shows sample results of segmented characters from segmented lines (Figure 5.6(a)) using the horizontal histogram projection (Figure 5.6 (b)). The result shows that each character is segmented accurately.



(a)



(b)



(c)

Figure 5.6: Output of character segmentation (a) input image (b) X histogram (c) segmented characters

One limitation of segmenting individual characters in our system using X histogram is when characters are in touch with each other, or when they are interleaved with each other; i.e. when there is no space between each characters. In this case, it segments those characters as one character as shown in Figure 5.7 (b). Another challenge is when a character has unconnected component, in which case, a single character is split into two characters due to opening problem, as shown in Figure 5.7 (d). These two problems created error in character segmentation results. However, it was observed that this happens rarely because Ethiopic scripts are written by preserving the space between characters. These problems were addressed by applying morphological operation.We have

(a) (b)

(c) (d)

Figure 5.7: Output of errored segmentation: (a) input image (b) segmenting two characters as one (c) input image (d) segmenting one character by splitting in to two characters

used a matlab function ”bwmorph (image, ’close’, n)” to connect the opened characters n times. However the function can not connect completely the extracted characters when the space is large. Therefore we have trace each characters and connect manually.

5.2.3 Discussion on the Prepared Dataset

For training and testing the system, we prepared dataset which consists of images of 7065 segmented characters extracted from 200 pages of input images of ancient Ethiopic manuscripts. Out of the total segmented characters, 24 out of the 26 Geez base characters with their derived characters were found. The remaining two characters ”ጸ” and ”ፐ” appeared four times and none respectively. The frequencies of some of the derived characters were also small and were not equally distributed compared to the frequency of the base characters. Base characters appeared on average from 100 up to 165. Due to this reason, the dataset consisted of only base characters and the frequency was set to 100 for each character. Each character was labeled taking 100 characters for each base character forming one similar class, and totally the dataset contains 2400 base characters for training and testing (see Appendix C). Among the total data set

70% was used for training and 30% used for testing the system.

5.3 Result of Training and Recognition

The proposed deep neural network was trained and tested by using the following experimental setup:

- The training set is provided as an input image, arranged in 1680 x 900 pixel mat file and 1680 x 1 label mat file. Sample binary input image in mat file format is shown at Figure 5.8.
- Each input image character is 30x30pixels, so number of input neurons is 900
- Number of characters for classification is 24, so the number of output neurons is set to 24
- Learning rate was set to 0.3
- Batch size was set to 50
- Momentum was set to 0.5
- Epoch was set to be variable; values of 50, 100, and 150 were tested
- Number of hidden layers and number of neurons in each layer was also set to variable and different values were set and tested for each
- Network model used was RBM
- Finally the proposed network was tested using 720 x 900 pixel mat file

data

1680x900 double

	16	17	18	19	20	21	22	23	24
102	1	1	0	0	0	1	1	1	1
103	1	1	1	1	1	1	1	1	1
104	1	0	0	0	0	0	1	1	1
105	1	1	1	1	1	1	1	1	1
106	1	1	1	0	0	0	0	1	1
107	1	1	1	1	1	0	0	1	1
108	1	1	1	1	1	1	1	1	1
109	0	1	1	1	1	1	1	1	1
110	1	1	1	1	1	1	1	0	1
111	1	1	1	1	1	1	1	1	1
112	1	1	1	1	1	1	1	1	1
113	1	1	0	0	0	0	1	1	1
114	1	1	0	1	1	1	1	1	1

Figure 5.8: Sample binary input image in mat file

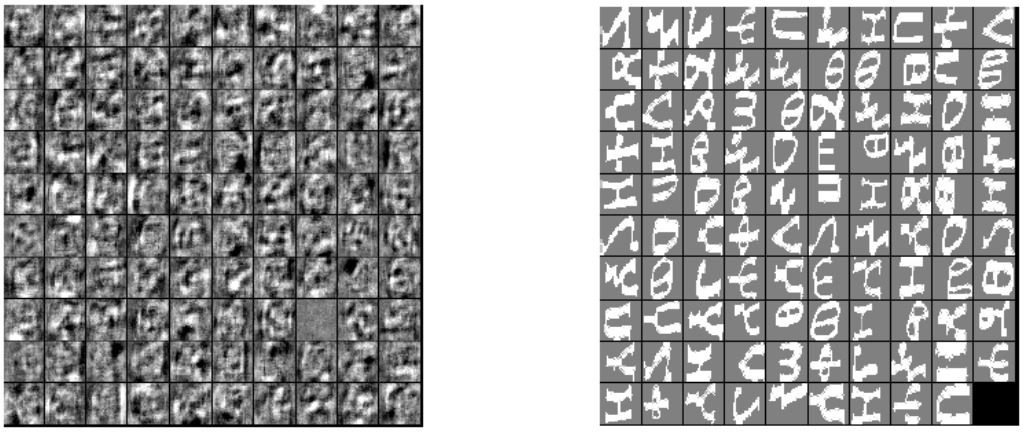
Various experiments were done by using different values of epoch, number of hidden layer, and number of hidden units. The first output to observe from the experiment is visualization of the learned weights for the 100 hidden units at the first hidden layer of the RBM. As shown in figure 5.8 (a), very coarse values of weights are observed which shows that only primitive features of characters are learned. This is expected since the first layers of deep networks perform only mapping of input features to low-level representation of images; so it is not expected to see character shapes at this stage.

To test how accurately the first RBM maps the input features to the first hidden layer, the first RBM is tested using the test data and output of the reconstructed characters are observed as shown in Figure 5.9 (b). The output shows that out of 100 characters used for testing, only one characters is miss classified. We tried to observe the output of reconstruction error, the sum of unpredictable testdata divided by the number of test data, by using different epochs as shown in Table 5.1.

Table 5.1: Trained RBM with 100 hidden neuron results with different epochs

Epoch	Classification error
50	0.093056
100	0.08333
150	0.075

From Table 5.1 the classification error is reduced as the number of epochs increased. Better result was obtained by using 150 epochs (0.075), this shows that the number training iterations has a significant impact on training RBM.



(a) (b)

Figure 5.9: RBM training result with 100 epochs: (a) learned weight (b) classification error for RBM

In a similar manner the proposed network was trained using entire training dataset by stacking various RBM networks which is produced by adding more hidden layers. The RBM stack was then fine tuned in supervised manner using softmax.

After training the network, we used the 720 character image test data to test

it, and see the recognition error by varying the number hidden layers, number hidden units and also the epochs.

The results of recognition errors for different hidden layers are summarized in Tables 5.2, 5.3 and 5.4. Tables 5.2 to 5.4 show the recognition error using 2, 3, and 4 hidden layers each with 100, 200, and 300 units for 50, 100, and 150 epochs respectively. It can be observed that, generally, the recognition error decreases as the number of epochs increases. This is expected, since the network enforces what it has learnt in each epoch. However, the rate of decrement of the error is slow, as can be deduced from the difference between two successive epochs. This again indicates that the error decrease will come to a point where no more decrease is observed, in which case the training is said to over fit.

Comparing the error values in the three tables, smallest recognition error (that is 0.0625 or 93.75% accuracy) was obtained for a network with three hidden layers, 300 hidden units and 150 epoch (Tables 5.3)

Table 5.2: Recognition error with two hidden layers

Epoch	2 hidden layers		
	100 units	200 units	300 units
50	0.141667	0.0875	0.08333
100	0.113889	0.076389	0.065278
150	0.097222	0.070833	0.063889

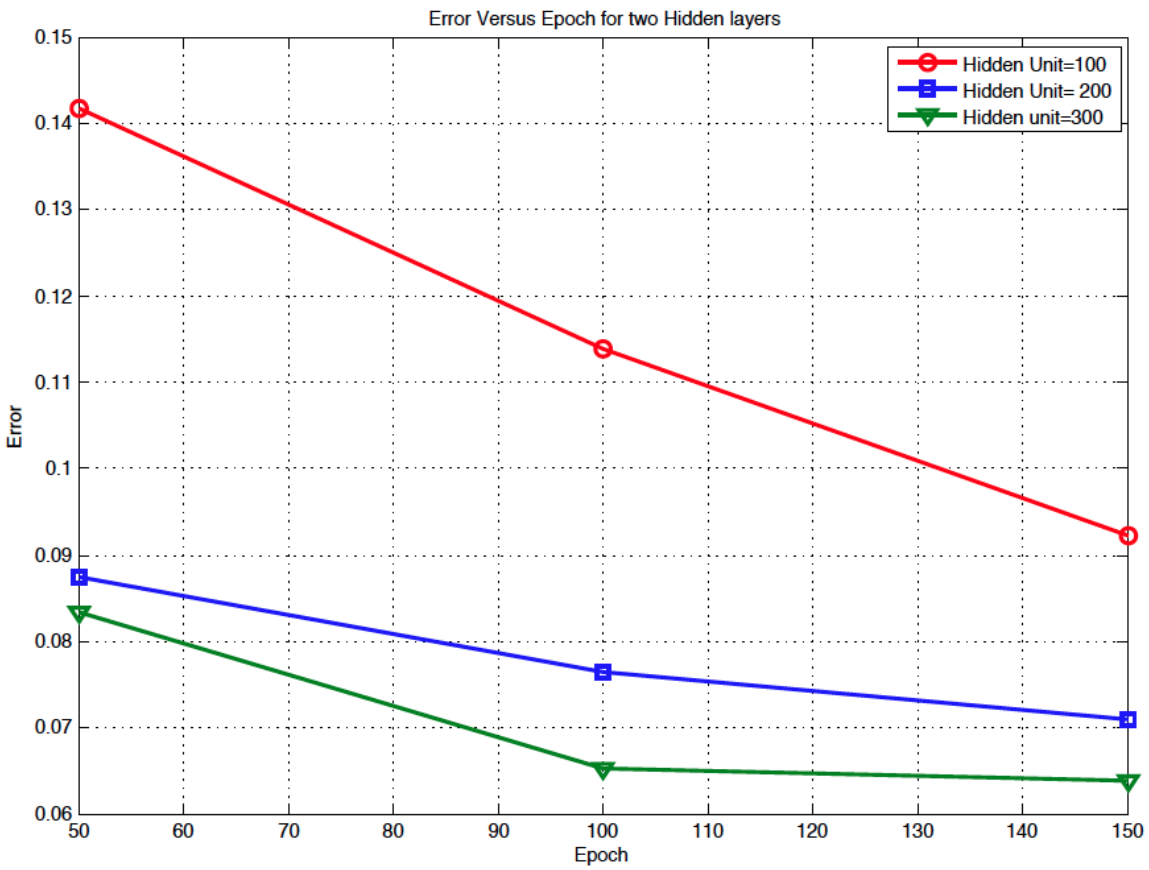


Figure 5.10: Graph of recognition error using two layers

The above graph, Figure 5.9, describes how the recognition error declines as the number of epochs and number of neurons increases using 2 hidden layers. From the graph best result is recorded at 300 neurons with 150 epochs.

Table 5.3: Recognition error with three hidden layers

Epoch	3 hidden layers		
	100 units	200 units	300 units
50	0.16667	0.098611	0.079167
100	0.151389	0.090278	0.072222
150	0.141667	0.06944	0.0625

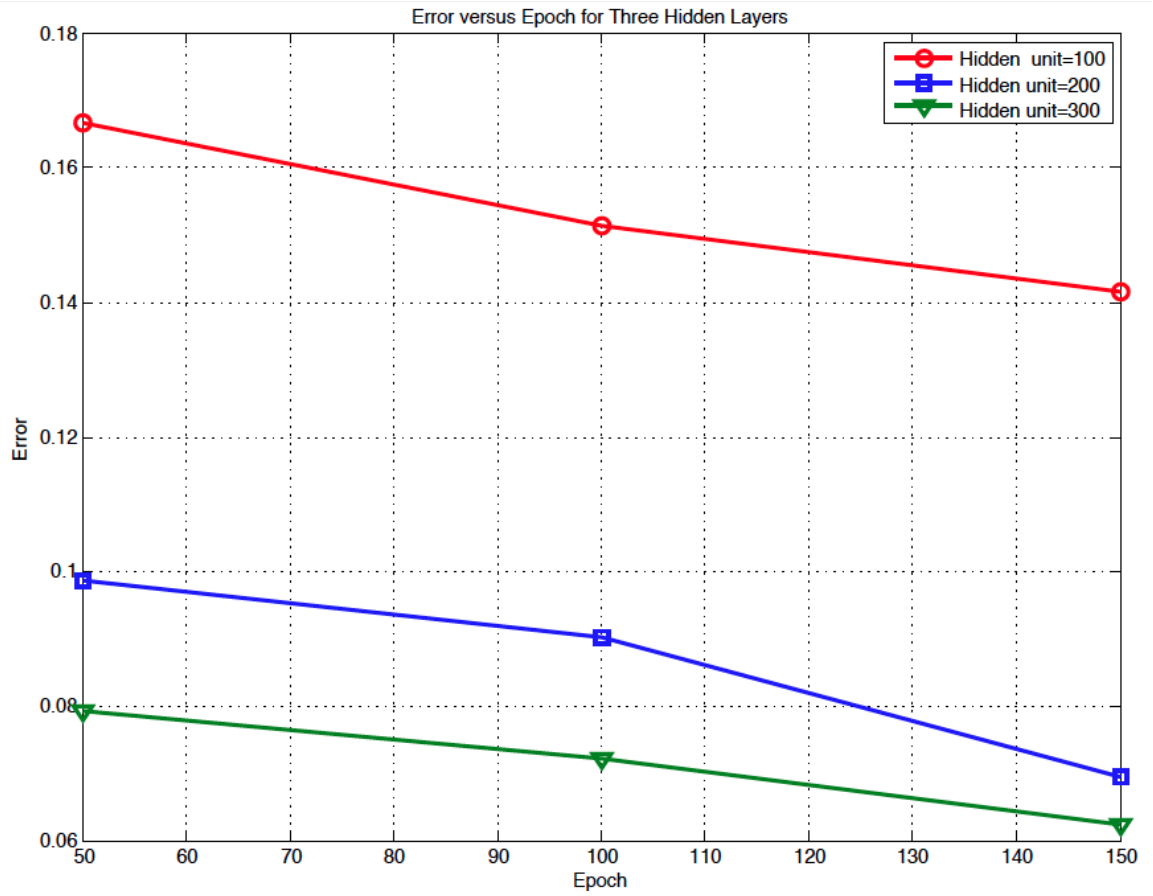


Figure 5.11: Graph of recognition error using three layers

The graph, Figure 5.10, shows that the recognition error decreases more at 300 neuron on 150 epochs using 3 hidden layers.

Table 5.4: Recognition error with four hidden layers

Epoch	4 hidden layers		
	100 units	200 units	300 units
50	0.218056	0.129167	0.105556
100	0.179167	0.1	0.0875
150	0.172222	0.0875	0.077778

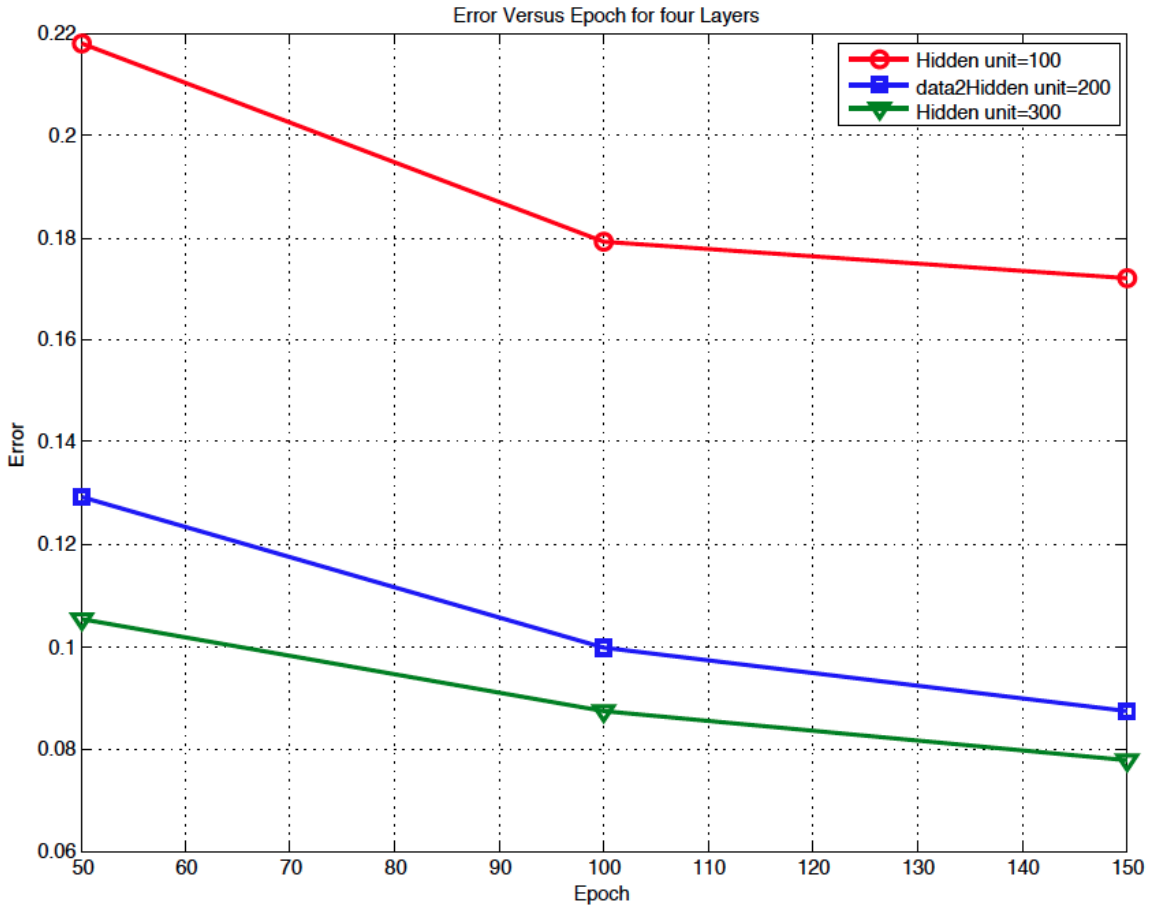


Figure 5.12: Graph of recognition error using four layers

Using four layers, Figure 5.11, the error decreases also but compared to the above layers the error starts at larger value, at 150 epoch with 300 units the error is larger than the 2 and 3 hidden layers.

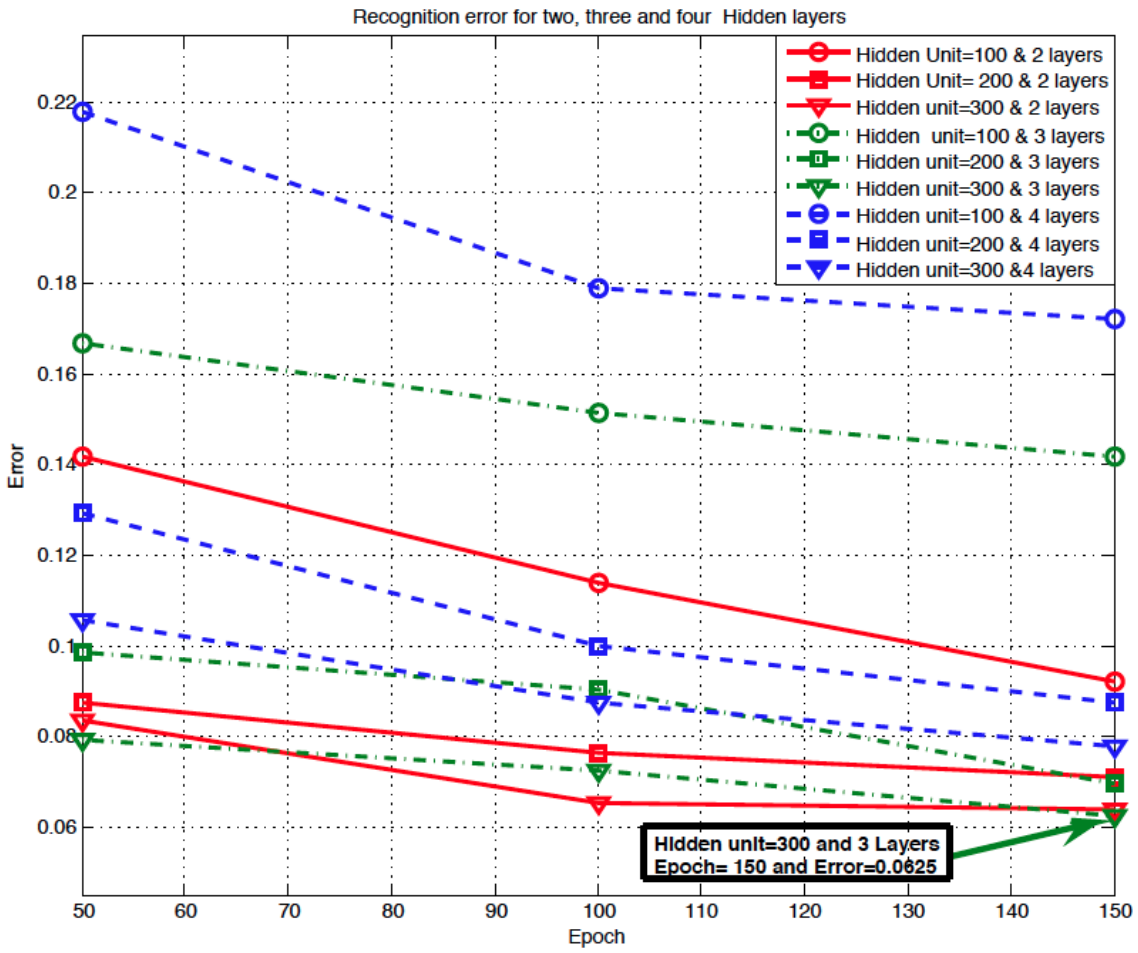


Figure 5.13: Graph of overall recognition error with two, three and four layers

The above graph illustrates the overall comparison of the three network types of layers i.e. between 2, 3 and 4 hidden layers. At 50 epochs the 4 hidden layer with 100 neurons gets larger error than 2 and 3 layers. when the number of epochs increased, at 100 epochs, the error at all layers become small. However, at 150 epochs the error using 4 layers starts to increase again, may be due to over fitting. Therefore the final good result relays on 3 layers, 300 neurons with 150 epochs as shown in Figure 5.12.

Chapter 6

Conclusions and Future work

6.1 Conclusions

This research work aimed to propose a system for recognition of Ancient Ethiopic manuscript recognition using the advanced machine learning technique which is DNNs. The recognition system consists of data acquisition, preprocessing, segmentation, and classification and recognition. Ancient manuscripts were collected from different sources,form NALA, EOTC and monasteries, through digital camera. and the collected image. The collected image were preprocessed to enhance the document images using binarization, morphological operation and skew correction. From these images individual characters were extracted using histogram profile segmentation method to prepare the dataset for training and testing the DNN.

The proposed system was trained using a data set of Geez characters. Our dataset consists 24 base characters of 100 frequencies, totally 2400 characters are prepared in mat file. Among the total characters 1680 used for training and 720 characetrs used for testing the system.

DNN are gaining importance in pattern recognition applications due to the recently proposed layer wise training procedure which relays on training RBMs to get improved recognition accuracy. In our experiment we have experimented to show the performance of DNN by varying a number of parameters. All the three test scenarios showed comparable and similar results (not less than 92%)

even though the best result obtained was 93.75% accuracy using 3 hidden layers with 300 hidden neurons at 150 epochs. This confirms that if properly trained with enough amounts of data DNN can be promising techniques for ancient handwritten document recognition with very good accuracy. Also analysis of results obtained from each step of the recognition process shows that the system can be extended to further research work to include all other base character families along with other two remaining base characters. The analysis made in this research work is promising that it can be fine-tuned for practical applications as well.

6.2 Future Work

A number of tasks can be recommended as future work. First in order to design a complete recognition system for ancient Ethiopic manuscript recognition, a complete dataset should be prepared that consists of the entire Geez alphabets, since there is no standard dataset for ancient geez manuscripts. This might require continuous effort to collect large amount of the necessary manuscripts.

Another task is testing the system with more number of hidden layers and epochs to get optimal network parameters and reduced errors. The impact of varying the number and value of model parameters like learning rate, batch size etc can be further studied.

Finally the recognition system can be extended to include diacritics and Geez number in ancient Ethiopic manuscripts documents to transfer those precious documents to next generation.

References

[1] K. Prasad, D. C. Nigam, A. Lakhotiya, D. Umre, and B. Durg, "Character recognition using matlab s neural network toolbox," *International Journal of u-and e-Service, Science and Technology*, vol. 6, no. 1, pp. 13–20, 2013.

[2] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Processing in Advances Neural Systems*. Curran Associates, Inc., 2008, pp. 577–584.

[3] Y. Assabie and J. Bigun, "Offline handwritten amharic word recognition," *Pattern Recognition Letters*, vol. 32, no. 8, pp. 1089–1099, 2011.

[4] G. F. Scelta and P. Quezzaire-Belle, "The comparative origin and usage of the geez writing system of ethiopia," *Unpublished manuscript, Boston University, Boston. Retrieved July*, vol. 25, p. 2009, 2001.

[5] D. Nosnitsin, "Ethiopian manuscripts and ethiopian manuscrit studies," *Gazette du livre médiéval*, no. 58, p. 1, 2012.

[6] wikibooks, "Artificial neural network." wikibooks.org, 2014.

[7] A. Yaregal, "Optical character recognition of amharic text: an integrated approach," Ph.D. dissertation, aau, 2002.

[8] Y. Assabie and J. Bigun, "Writer-independent offline recognition of handwritten ethiopic characters," *Proc. 11th ICFHR*, pp. 652–656, 2008.

[9] T. Phienthrakul and W. Chevakulmongkol, "Handwritten recognition on pali cards of buddhadasa indapanno," in *Computer Science and Engineering Conference (ICSEC), 2013 International*. IEEE, 2013, pp. 191–195.

[10] M. Diem and R. Sablatnig, "Recognizing characters of ancient manuscripts," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 753 106–753 106.

[11] S. Perantonis, B. Gatos, K. Ntzios, I. Prattkakis, I. Vrettaros, A. Drigas, C. Emmanouilbdis, A. Kesidis, and D. Kalomirakis, "A system for processing and recognition of old greek manuscripts(the d-scribe project)." *WSEAS Transactions on Computers*, vol. 3, no. 6, pp. 2049–2057, 2004.

[12] A. Yuan, G. Bai, L. Jiao, and Y. Liu, "Offline handwritten english character recognition based on convolutional neural network," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 125–129.

[13] D. Ciresan and U. Meier, "Multi-column deep neural networks for offline handwritten chinese character classification," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–6.

[14] D. C. Cireşan, U. Meier, and J. Schmidhuber, "Transfer learning for latin and chinese characters with deep neural networks," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–6.

[15] A. Graves, "Offline arabic handwriting recognition with multidimensional recurrent neural networks," in *Guide to OCR for Arabic Scripts*. Springer, 2012, pp. 297–313.

[16] Y. Asabie and J. Bigun, "Lexicon-based offline recognition of amharic words in unconstrained handwritten text," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[17] M. Shah and G. B. Jethava, "A literature review on hand written character recognition," *Indian Streams Research Journal*, vol. 3, no. 2, pp. 1–19, 2013.

[18] V. L. Sahu and B. Kubde, "Offline handwritten character recognition techniques using neural network: A review," *International journal of science and Research (IJSR)*, vol. 2, no. 1, pp. 87–94, 2013.

[19] V. Sokratis, E. Kavallieratou, R. Paredes, and K. Sotiropoulos, "A hybrid binarization technique for document images," in *Learning Structure and Schemas from Documents*. Springer, 2011, pp. 165–179.

[20] N. Ntogas and D. Veintzas, "A binarization algorithm for historical manuscripts," in *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, no. 12. World Scientific and Engineering Academy and Society, 2008.

[21] N. V. R. A. S. Rao, S. Balaji, and L. P. Reddy, "Cleaning of ancient document images using modified iterative global threshold," *Proceedings of International Journal of Computer Science Issues November*, 2011.

[22] M. H. Ramappa and S. Krishnamurthy, "Skew detection, correction and segmentation of handwritten kannada document," *International Journal of Advanced Science and Technology*, vol. 48, 2012.

[23] A. Papandreou, B. Gatos, G. Louloudis, and N. Stamatopoulos, "Icdar 2013 document image skew estimation contest (disec 2013)," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1444–1448.

[24] N. Dave, "Segmentation methods for hand written character recognition," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 4, pp. 155–164, 2015.

[25] V. J. Dongre and V. H. Mankar, "Devnagari document segmentation using histogram approach," *arXiv preprint arXiv:1109.1247*, 2011.

[26] N. Anupama, C. Rupa, and E. S. Reddy, "Character segmentation for telugu image document using multiple histogram projections," *Global Journal of Computer Science and Technology*, vol. 13, no. 5, 2013.

[27] R. P. dos Santos, G. S. Clemente, T. I. Ren, and G. D. Cavalcanti, "Text line segmentation based on morphology and histogram projection," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 651–655.

[28] M. Rani and Y. K. Meena, "An efficient feature extraction method for handwritten character recognition," in *Swarm, Evolutionary, and Memetic Computing*. Springer, 2011, pp. 302–309.

[29] A. Yaregal, "Optical character recognition of amharic text: an integrated approach," Ph.D. dissertation, aau, 2002.

[30] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 31, no. 2, pp. 216–233, 2001.

[31] M. Shrivastava, M. Sahu, and M. Rizvi, "Artifical neural network based character recognition using backpropagat," *International Journal of Computers & Technology*, vol. 3, no. 1, p. 44, 2012.

[32] [Online]. Available: http://www.psych.utoronto.ca/users/reingold/courses/ai /cache/neural3.html, accessed on 12/20/2015

[33] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[34] [Online]. Available: http://deeplearning4j.orgneuralnet-overiew.html, accessed on 01/17/2016

[35] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *The Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.

[36] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[37] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[38] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.

Appendix

Appendix A: Sample of photographed Ancient Ethiopic Manuscript images

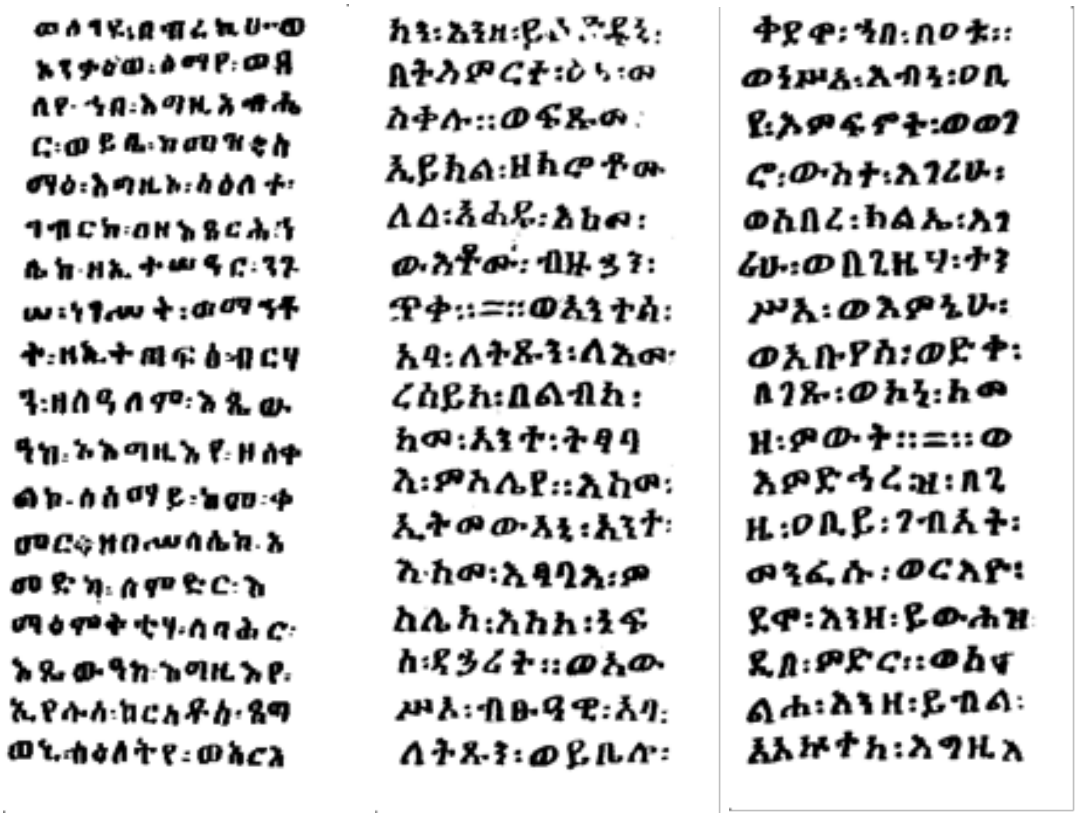
(a)

(b)

(c)

(d)

Appendix B: Result of Preprocessed stage of Photographed images



(a) (b) (c)

Appendix C: Sample of Geez characters used in training and testing

Appendix D: Sample Implemented Algorithm Codes

1. Hybrid bineriaztion algorithm

```
% Read true color image
rgbimage=imread('D:\OrgImgColJpg\MZG-FE-01-2.jpg');

%convert the rgb image to grayscale image
gimage=rgb2gray(rgbimage);
figure,imshow(rgbimage),title('Orginal Image');
figure,imshow(gimage),title('Graysacle Image');
tic
%binarize using otsu methods
level = graythresh(gimage);
BW = im2bw(gimage,level);
figure,imshow(BW),title('otsu binarization');
toc
function IH=itr(I)
    %calculate the intial threshold value
    T=sumofpixels/(M.*N);
    % Apply itrative Global methods using for loop
    for g=1:20
    IS=I-T+1;           % each intensty value of the grayimage
subtructed by T
    %figure, imshow(IS), title('subtructed image');
    A=1-IS;
    E=min(IS(:));
    ET=1-E;
    IH=1-(A/ET);        % histogram streched image
    %figure, imshow(IH), title('streched image');
    I=IH;               % final streched image
    T=(sum(sum(I)))/(M.*N);
```

```
    end
% segment the entire image using 50 by 50 window size
a=1; b=1; c=0; z=0; q=1;

for i=1:windowsize:M1-windowsize+1
    for j=1:windowsize:N1-windowsize+1
       for m=i:i+windowsize-1
            for n=j:j+windowsize-1
                if I(m,n)==0
                    c=c+1;
                end
                k(a,b)=I(m,n);
                kn(a,b)=IN(m,n);  %segment intermidiate image
                if k(a,b)==0
                    z=z+1; % no of black freq of segmented image
                end
                b=b+1;
            end
             a=a+1;
             b=1;
       end
          fs(q)=z;
          kn_mean(q)=mean(mean(kn));
        % reset a, b and z
            a=1; b=1; z=0; q=q+1;

    end
end
fs;
m=mean(fs);
s=std(fs);
k=0.2;
w=m+s*k;

.
.
.
.
.
%return (Ifinal);
toc
figure,imshow(Ifinal),title('Hybrid binarization')
cd D:\binery
        %filename=sprintf('win%d.bmp',f);
        imwrite(Ifinal,'YT-.bmp','bmp');

end
```

2. Y histogram algorithm

```
rimage=imread('D:\binery\DE-7.bmp');
. . .
% find or plot y histogram of the entire image
cnt=0;
for i=1:M
    for j=1:N
        %cnt=0;
        if I(i,j)==0
            cnt=cnt+1;
        end
    end
    v(i)=cnt;
    n(i)=i;
    cnt=0;
end
figure, plot(v,n),title('using for loop');
. . .
%.....................new.............................
i=1; f=1;
while (i<M-9)
    while(v(i)==0)     %space b/n lines
        i=i+1;
        if i==M-1
            break;
        end
    end
    k=1; a=1;
    I2=1;
    while(v(i+k)>0)

        for a=1:N
            I2(k,a) =I(i+k-1,a);
        end
        k=k+1;
    end
    cd D:\seg
    %fn = fullfile('D:\seg', [num2str(f), '.bmp']);
   filename=sprintf('line%d.bmp',f);
   imwrite(I2,filename,'bmp');
   %imshow(I2);
   % I2
    %imwrite(fn,I2);
    f=f+1;
    i=i+k;  end
```

3. X histogram algorithm

```
function charseg(~)
clear all;
close all;
% character segmentation
testfile=dir('D:\seg\*.bmp');
ls=length(testfile);
. . .
[M1,N1]=size(I);
M1;
N1;
cnt1=0;
for j=1:N1
    for i=1:M1

        if I(i,j)==0
            cnt1=cnt1+1;
        end
    end
    vn(j)=cnt1;
    nn(j)=j;
    cnt1=0;
end
%vn
figure, plot(nn,vn),title('x histogram');
%thresholding

. . .
I3=imresize(I2,[30 30]);
            %I3=imdilet(I3,se);
             cd D:\char
             filename=sprintf('char%d-%d.bmp',s,f);
             imwrite(I3,filename,'bmp');
              f=f+1;
              j=j+k;
end

end
```

4. RBM training and testing algorithm

```
load geez_compl_dataset

%% Train RBM for classification
%train rbm with 100 hidden units

m=rbmFit (data,300,labels,'verbose',true);
yhat=rbmPredict(m,testdata);

function prediction = rbmPredict(m, testdata)


fprintf('Classification error using RBM with 100 hiddens is
%f\n', ...
    sum(yhat~=testlabels)/length(yhat));


%% Train a DNN
op.verbose=true;
models=dnnFit(data,[300 300 300 ],labels,op,op,op);
yhat2=dnnPredict(models,testdata);

. . . .
%accumulate reconstruction error
        err= sum(sum( (data-negdata).^2 ));
        errsum = err + errsum;
    end

    errors(epoch)= errsum;
    if (verbose)
        fprintf('Ended epoch %i/%i, Reconsruction error is
%f\n', ...
            epoch, maxepoch, errsum);
    end
end

%output of the trained network
model.W= Wavg;
model.b= bavg;
model.c= cavg;
model.Wc= Wcavg;
model.cc= ccavg;
model.labels= u;
```

5. Fine-tuning using soft-max algorithm

```
function[oneofn] = softmax_sample(probmat)
oneofn = zeros(size(probmat));
probmat = probmat./repmat(sum(probmat,2),1,size(probmat,2));
for i=1:size(probmat,1)
    probs = probmat(i,:);
    sample = cumsum(probs);
    sample = sample>rand();
    index = find(max(sample) == sample);
    index = min(index);
    sample = zeros(1,length(probs));
    sample(index) = 1;
    oneofn(i,:) = sample;
end
```

…..

```
function mu = softmaxPmtk(eta)
% Softmax function
% mu(i,c) = exp(eta(i,c))/sum_c' exp(eta(i,c'))


tmp = exp(eta);
denom = sum(tmp, 2);
[D, C] = size(eta);
%mu = tmp ./ repmat(denom, 1, C);
mu = bsxfun(@rdivide, tmp, denom);

end
```