# ANCIENT ETHIOPIC MANUSCRIPTS CHARACTER RECOGNITION USING DEEP BELIEF NETWORKS

Siranesh Getu[1], Eneyachew Tamir[1] and Menore Tekeba[2]
Corresponding Author's Emails: saragetu8@gmail.com

[1] School of Electrical and Computer Engineering, Bahir Dar IOT, Bahir Dar University
[2] School of Electrical and Computer Engineering, AAiT, Addis Ababa University

## ABSTRACT

*Very large proportion of Ethiopian literature is found in ancient ge'ez manuscripts in the form of old scriptures with papers from animal hides and skins (Branas) on which the ancient Ethiopic knowledge and civilization is recorded. This knowledge can be extracted and made usable by applying optical character recognition (OCR) systems on document images. Little efforts have been done for OCR of Ethiopic ancient manuscripts. Hand written OCR process is considered as one of the most challenging problems in the area of image processing . The unique morphology of ge'ez hand-writing system (known as "Kum Tsihfet"), the degraded quality of the documents, and non-uniform background of the Branas poses additional challenges. Because of this, the OCR technique employed can't be addressed directly by using OCR systems designed for modern printed and handwritten documents. Machine learning techniques like deep belief networks (DBNs) are becoming powerful set techniques that attempt to model complicated morphological features of handwritten texts. In this research we developed an OCR system using DBNs. The system was trained and tested using our own segmented datasets of ancient ge'ez characters containing 24 base characters only. The test result shows that a recognition accuracy of 93.75% was obtained, which is a promising result.*

**Keywords**: OCR, Deep Belief Networks, Ancient Ge'ez documents

## INTRODUCTION

OCR is the process of detecting and recognizing characters from input image and converting into machine editable text. There are two types of handwriting recognition: off-line and on-line. The off-line OCR uses image documents from computers and coverts those document images to texts. The on-line OCR systems directly takes the writers pen strokes from writing pads and converts those pen-strokes and lines directly to sequences of words [1]. As per the research work given in [2], the online hand-written OCR systems performs better accuracy using temporal pen stroke directions and sequences for recognition task.

Handwritten document 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 [2]. Offline recognition system follow holistic and segmentation based approaches [3]. The holistic approach is used to recognize limited size vocabulary of words where global features extracted from the entire word image are considered. As the size of the vocabulary of words increases, the complexity of holistic based algorithms also increases and correspondingly the recognition rate decreases rapidly [4]. 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 [4].

Many of the approaches proposed mainly focus on recognizing characters or words of specific language, which consists of its own specific features [3].

The off-line OCR systems are used to covert modern printed documents, hand-written documents, and ancient hand-written documents which are produced before printing technology starts. Geez is one of the world's ancient languages [5] and enormous amounts of handwritten material are found in churches, monasteries, museums, libraries, and many places of Europe and America stolen by travelers and invaders. These scripts are mostly written in parchments, and consist of intrinsic values and unexplored content until today.

To the best of our knowledge, there is no a research work done to address OCR issues associated with Ethiopic ancient manuscript characters of having "Kum Tsihfet" style and the rich cultural, technology, artistic, governance, science, religious and other knowledge and wisdom left unexplored due to lack of such solutions. The aim of this study is, therefore, to develop an OCR system using DBNs, train and test the network with our own segmented dataset of 24 (out of 26) base characters of Ge'ez which are shown in the first column of Figure 1.

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.

Researches on offline OCR systems are generally done for two types of documents: printed and handwritten. A reach set researches on printed text exist, [6, 7, 8] for Amharic and [2, 9] for other languages with promising results. However only limited attempts were made [3, 10, 11] in the area of Amharic handwritten document, which reveals itself open to research.

| | | | | | | |
|---|---|---|---|---|---|---|
| ሀ | ሁ | ሂ | ሃ | ሄ | ህ | ሆ |
| ለ | ሉ | ሊ | ላ | ሌ | ል | ሎ |
| ሐ | ሑ | ሒ | ሓ | ሔ | ሕ | ሖ |
| መ | ሙ | ሚ | ማ | ሜ | ም | ሞ |
| ሠ | ሡ | ሢ | ሣ | ሤ | ሥ | ሦ |
| ረ | ሩ | ሪ | ራ | ሬ | ር | ሮ |
| ሰ | ሱ | ሲ | ሳ | ሴ | ስ | ሶ |
| ቀ | ቁ | ቂ | ቃ | ቄ | ቅ | ቆ |
| በ | ቡ | ቢ | ባ | ቤ | ብ | ቦ |
| ተ | ቱ | ቲ | ታ | ቴ | ት | ቶ |
| ኀ | ኁ | ኂ | ኃ | ኄ | ኅ | ኆ |
| ነ | ኑ | ኒ | ና | ኔ | ን | ኖ |
| አ | ኡ | ኢ | ኣ | ኤ | እ | ኦ |
| ከ | ኩ | ኪ | ካ | ኬ | ክ | ኮ |
| ወ | ዉ | ዊ | ዋ | ዌ | ው | ዎ |
| ዐ | ዑ | ዒ | ዓ | ዔ | ዕ | ዖ |
| ዘ | ዙ | ዚ | ዛ | ዜ | ዝ | ዞ |
| የ | ዩ | ዪ | ያ | ዬ | ይ | ዮ |
| ደ | ዱ | ዲ | ዳ | ዴ | ድ | ዶ |
| ገ | ጉ | ጊ | ጋ | ጌ | ግ | ጎ |
| ጠ | ጡ | ጢ | ጣ | ጤ | ጥ | ጦ |
| ጰ | ጱ | ጲ | ጳ | ጴ | ጵ | ጶ |
| ጸ | ጹ | ጺ | ጻ | ጼ | ጽ | ጾ |
| ፀ | ፁ | ፂ | ፃ | ፄ | ፅ | ፆ |
| ፈ | ፉ | ፊ | ፋ | ፌ | ፍ | ፎ |
| ፐ | ፑ | ፒ | ፓ | ፔ | ፕ | ፖ |

Figure 1: Geez Characters

As a contribution we developed an OCR system based on machine learning technique that implements DBN. In order to train and test the system, we prepared our own data sets that consists of a total of 2400 characters (100 images of the 24 base characters) extracted from 200 pages of selected ancient Ethiopic manuscripts. Subjective evaluation using percentage accuracy of performance of the system was done by examining outputs the different stages.

The rest of this paper is organized as follows: section 2 presents literature review on OCR system. Section 3 describes the methodology and implementation approach followed in this research. Test and results at different stages of the entire system is described in section 4. Finally, conclusion and future works are indicated in section 5.

## 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 a 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 have become challenging for many researchers in producing solution in converting to computer readable format. The research of ancient Ethiopic manuscript processing is almost unexplored. This section presents the pervious works on ancient script, modern scripts and finally Amharic recognition using different techniques.

The work in [12] investigates ancient Slavonic manuscripts from the 11th century. 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 [12].

The writers in [9] applied Convolutional Neural Networks (CNNs) for offline handwritten English character recognition. They modified 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 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 [9].

Another approach on Ethiopic scripts [10] 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% [10].

## METHODOLOGY AND IMPLEMENTATION DETAILS

The proposed system includes the basic steps shown in Figure 3.1 namely, image acquisition, preprocessing, segmentation, classification and recognition [11, 13]. The details of the methods are given in [11].

### Dataset Preparation

We have prepared a dataset both for training and testing the OCR of ancient characters of Ethiopic manuscripts. The dataset preparation involved several steps from image acquisition followed by different image processing algorithms and finally segmentation of characters using vertical and horizontal histograms projection. The steps used to prepare the dataset are given in the following steps:

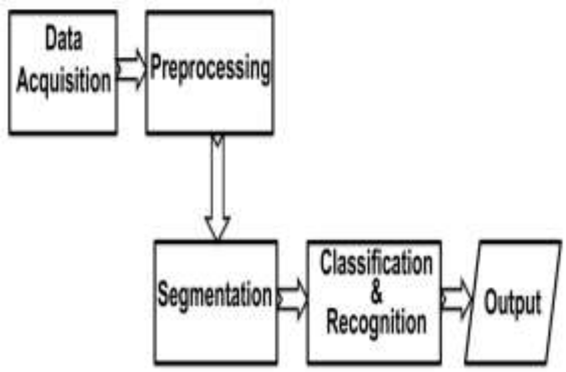



Figure 3.1 Block diagram of proposed system

**Image Acquisition**

The input image can be taken through camera or scanner. The first stage in character recognition (CR) is image acquisition, which involves getting digital image of manuscript pages for input to the system. We collected about 200 pages of ancient Ethiopic manuscripts from different sources such as Ethiopian Orthodox Tewahido Churches, monasteries and also from National Archives and Library Agency for data preparation.

**Preprocessing**

This step involves the following tasks:

**Grey Scale Conversion and Binarization:** Images which are in color are converted to grayscale before binarization is applied. Binarization is used to remove the noise and improves the quality of the documents by converting the gray-scale document images to black and white (binary) ones. Binarization is required because documents can often suffer degradation problems, especially in the case of historical documents due to unwanted foreground information (noise) [14]. Based on the threshold values used, there are two general techniques for binarization [15, 4]: Global binarization and local binariztion thresholding. But a technique for ancient or historical documents specifically is proposed in [14] and shows a very good result. This method is called hybrid binariztion technique and we have used for our images.

**Skew Detection and Correction:** The relative inclination angle of the page being acquired during scanning or taking photo of the page must be detected and accounted for as it can cause serious performance deterioration of segmentation and recognition stage of document processing system [16].

Since skew observed in our input images are global skew problem which occurs due to capturing the scripts through digital camera or rotated scanning, it can be detected and corrected using Bounding Box Technique. Bounding Box technique [16] is a way of finding the extreme corners of text image. 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 and it is also computationally inexpensive when compared to other methods.

**Noise Reduction:** There are many kinds of noise in images like Salt and Pepper Noise, the black points and white points sprinkled all over an image [17]. These can be reduced using filtering and applying morphological operations. We have applied morphological operation at this stage.

**Segmentation**

Segmentation of hand written text document into individual character or digit is an important and crucial phase in document analysis and character recognition. There are various factors such as noise and disconnected characters that affects the process of text image segmentation [18]. The quality of the image is a significant factor for text segmentation. In Ancient Ethiopic scripts due to ageing the text is highly affected by background noise, drop of ink and handling problem. We have used vertical profile projection for line segmentation and horizontal profile projection for individual character segmentation as it is given in [18, 19, 20].

**Classification and Recognitions**

**DBNs for OCR of Ancient Ethiopic Manuscripts:** OCR systems extensively use the methodologies of pattern recognition, which assigns an unknown sample into a predefined class. Numerous techniques for OCR can be investigated in four general approaches of pattern recognition [21]: Template Matching, Statistical Techniques, Structural Techniques, and Neural Networks but each of these approaches except the neural network approach lacks flexibility to adapt to new unseen challenges and we have selected neural network approaches for our particular problem.

From neural networks, deep learning neural networks are powerful set of techniques. 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. 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 [22].

According to Bengio et al. [14], shallow architectures 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. To solve this problem, it is recommended to represent a highly-varying function compactly (with few parameters) through the composition of many non-line arities, i.e., with a deep architecture. [14]

DBNs are one type of deep learning algorithms which uses a greedy layer-wise unsupervised pre-training and a light-weight supervised fine-tuning any back-propagation training algorithms [14].

**Training DBNs:** 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 [23].

Hinton et al. recently introduced a greedy layer-wise unsupervised learning algorithm, a generative model with many layers of hidden causal variables [14]. 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. Then neural network is fine-tuned to the global supervised objective. 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 (AE). AE uses back propagation algorithm for semi-supervised training of each layer pairs and computationally intensive. Therefore, we have used RBM as our training algorithm.

RBM is a generative model that uses a layer of binary variables to explain its input data [23, 24], undirected bipartite graphical model with connections between visible nodes and hidden nodes. The pixels correspond to visible units of the RBM because their states are observed; the feature detectors correspond to hidden units. A joint configuration, (**v, h**) of the visible and hidden units has an energy given by:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})} \quad (1)$$

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

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v}.\boldsymbol{h})} \quad (2)$$

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

$$p(\boldsymbol{V}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})} \quad (3)$$

The derivative of the log probability of a training vector with respect to a weight is given by:

$$\frac{\partial logp(V)}{\partial w_{ij}} = \langle V_i h_j \rangle_{data} - \langle V_i h_j \rangle_{model} \quad (4)$$

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

$$\Delta w_{ij} = \in \langle V_i h_j \rangle_{data} - \langle V_i h_j \rangle_{model} \quad (5)$$

where ∈ is learning rate.
Given a randomly selected training image, **V**, the binary state, $\mathbf{h_j}$, of each hidden unit, j, is set to 1 with probability:

$$p(h_j = 1 | v) = \sigma(b_j + \textstyle\sum_i v_i w_{ij}) \quad (6)$$

Where $v_i h_j$ is then an unbiased sample and σ(x) is the logistic sigmoid function given by:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \quad (7)$$

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 by using:

$$p(v_j = 1 | h) = \sigma(a_i + \textstyle\sum_i h_j w_{ij}) \quad (8)$$

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 8 and the change in a weight then given by equation 5 above is the final weight change using the reconstruction and the given data.

**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 better result. The basic architecture of the network is shown in Figure 2.

During designing of the neural network the number of different parameters of the network need to be decided. The model parameters that are required for the proposed network are described below. The values used for some parameters are typical values used in research [25] and others tuned during experimentation.

Number of neurons in the hidden layer: number of processing unit or nodes in the hidden layer.

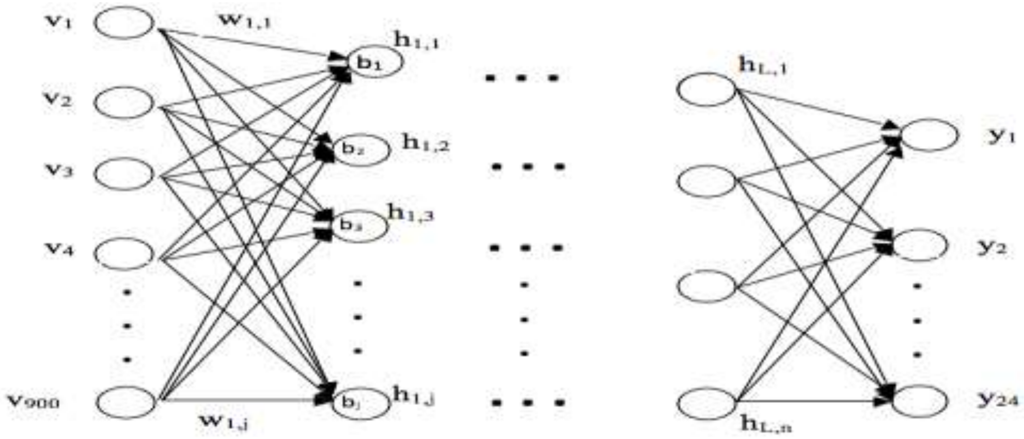



Figure 2: Architecture of the proposed deep network with 900 input, L hidden and 24 nodes of layers

**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] and its 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] and typical value is 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. Value Selection domain: Integer Domain: [1,∞). The values practically used are as low as 50 and as high as tens of thousands. The following steps are our gross steps used to train deep neural network.

Pre-training one layer at a time in a greedy way; using unsupervised learning at each layer in a way that preserves information from the input and disentangles factors of variation; Fine-tuning the whole network with respect to the ultimate criterion of interest.

Each RBM layer is 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) as given by:

$$\arg\max_w \prod_{v\in V} P(v) \tag{9}$$

Or equivalently, to maximize the expected log probability given by:

$$\arg\max_w E\left[\sum_{v\in V} logP(v)\right] \tag{10}$$

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 to compute weight update [15]. For computational simplicity and since it is the most widely used approach, we have used single-step contrastive divergence (CD-1) procedure which is given in the following steps:

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 as given here with equation 11:

$$w_{i,j} = \in \left(vh^T - v'h'^T\right) \quad (11)$$

The update rule for the biases a and b is defined analogously.

After each layer of RBM consisting of pairs of layers (visible and hidden) are training using RBM, 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 [25]. Softmax function converts a raw value in to a posterior probability.

### Implementation Details

The proposed system is implemented in the following way:

**For Dataset Preparation Methods:** The programs are being implemented using MATLAB 2010 starting from image conversion to grey scale to the segmentation of individual characters from images.

**For DBNs:** We have used and customized deep learn tool box which is an open source MATLAB 2014 library for training the network and to make the tests.

**Experimental Setup:** A laptop computer with 3GB memory, Intel core i3 2.3GHz 3MB L3 Cache having windows 7 operating systems was used to conduct the experiment and a Sony digital camera with 16MP was used for image acquisition.

## TEST RESULTS OF THE DBN CLASSIFIER

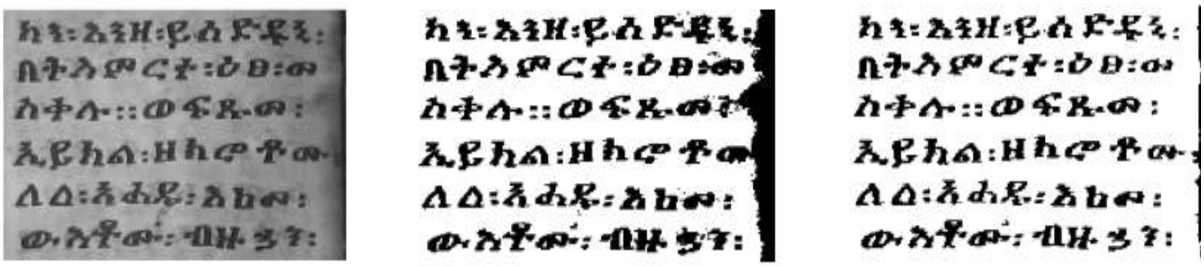
To the best of our knowledge, since there is no other research result in ancient Ethiopic manuscripts, it was very difficult to present the comparison of the selected system with other recognition algorithms. Discussion and analysis of results are mostly based on percentage accuracy.

### Results of Dataset Preparation Methods

These steps are only used for an automatic preparation of the dataset for our training and we haven't made performance evaluations on these methods. 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 3.

(a) (b)

Figure 3: Output of RGB to gray scale conversion: (a) color image – left (b) gray image– right



(a) (b) (c)

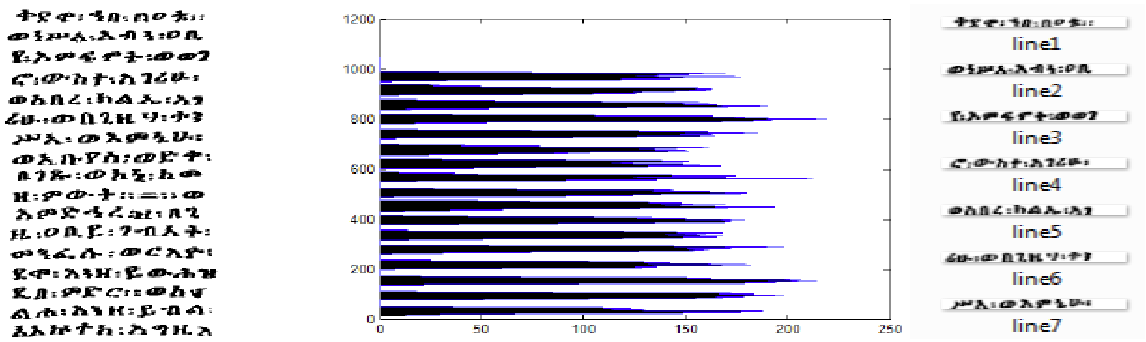
Figure 4: Output of binarization: (a) gray scale image (b) binarization using Otsu (c) binarization using hybrid

We have gotten better result using hybrid binarization technique. The comparison of the Otsu and the hybrid techniques are given in Figure 4. 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 or during scanning, which is global skew problem. In many document analysis problems, skew angles as high as 23 degrees are observed. We have applied skew detection and correction on each image document as shown in Figure 5.

(a) (b)

Figure 5: Output of skew detection: (a) skewed image (b) deskewed image

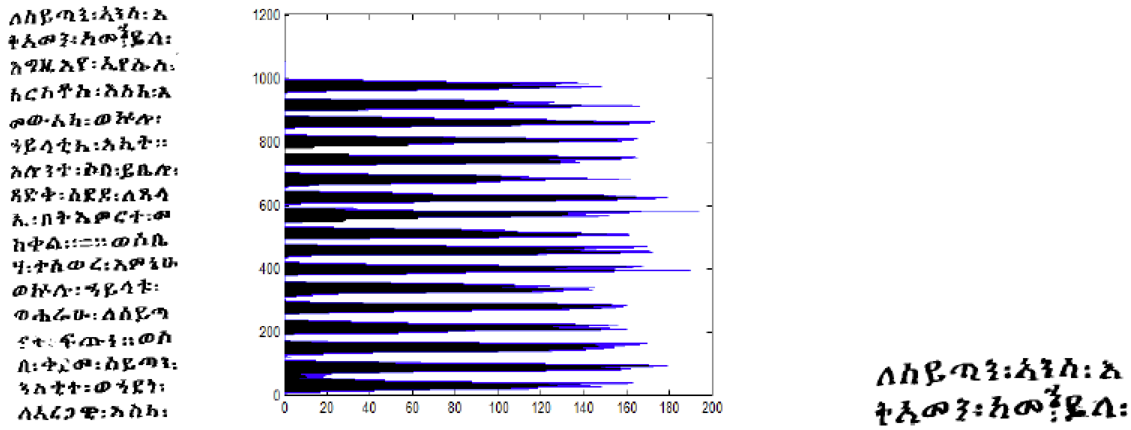
For segmenting lines and characters in images, vertical (Y histogram) and horizontal (X histogram) projections are implemented respectively. The result of vertical projection for a sample page is shown in Figure 6. As shown in the figure, the lines of the sample images are segmented accurately. The algorithm was tested with a number of document pages, and all tests produced perfect results; this shows that the selected algorithm works very well.




(a) (b) (c)

Figure 6: 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 7 (a) and (b) respectively In this case, the algorithm will segment the two lines incorrectly as a single line, as shown Figure 7(c).



(a) (b) (c)

Figure 7: Line segmentation error (a) input image (b) Y histogram (c) error segmented

From the given segmented line, each of the characters is extracted automatically using the X histogram. Figure 8(c) shows sample results of segmented characters from segmented lines (Figure 8(a)) using the horizontal histogram projection (Figure 8(b)). The result shows that each character is segmented accurately.

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 9(d).

After dataset preparation methods are done, we have gotten 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 “ጰ” and “ፐ” appeared four times and none respectively.

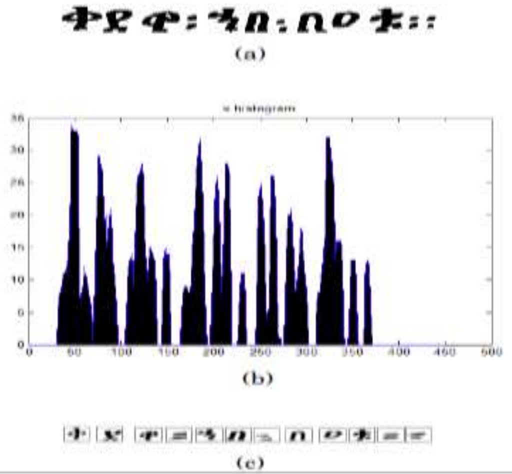


Figure 8: Output of character segmentation (a) input image (b) X histogram (c) segmented characters

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.

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.

Figure 9: Error in output segmentation: (a) input image (b) segmenting two characters as one (c) input image (d) segmenting one character by splitting in to two characters

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. Among the total data set 70% was used for training and 30% used for testing the system.

**Result of Training and Recognition Test**

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. 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 and 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 from 2 to 4 and different values were set and tested for each.

The pre-training algorithm for the network model used was RBM and final fine tuning used was soft max. Finally the proposed network was tested using 720 x 900 pixel mat file of 720 characters.

Various experiments were done by using different values of epoch, number of hidden layer, and number of hidden units. The recognition error is the ration of number of misclassified characters to total number of characters in the test dataset. Accuracy is also the ratio of number of correct classification to total number of characters.

Table 1: Recognition error with two hidden layers

| Epoch | Two 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 |

The results of recognition errors for different hidden layers are summarized in Tables 1, 2 and 3. Tables 1 to 3 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 epochs (See Table 2). It is also observed that in the four layers, at 150 epochs with 300 units the error is larger than the 2 and 3 hidden layers with the same number of epochs.

Table 2: Recognition error with three hidden layers

| Epoch | Three 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 | 0.06250 |

Table 3: Recognition error with four hidden layers

| Epoch | Four Hidden Layers | | |
|---|---|---|---|
| | 100 units | 200 units | 300 units |
| 50 | 0.218056 | 0.129167 | 0.105556 |
| 100 | 0.179167 | 0.10000 | 0.08750 |
| 150 | 0.172222 | 0.0875 | 0.077778 |

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 layers 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 with 93.75% accuracy.

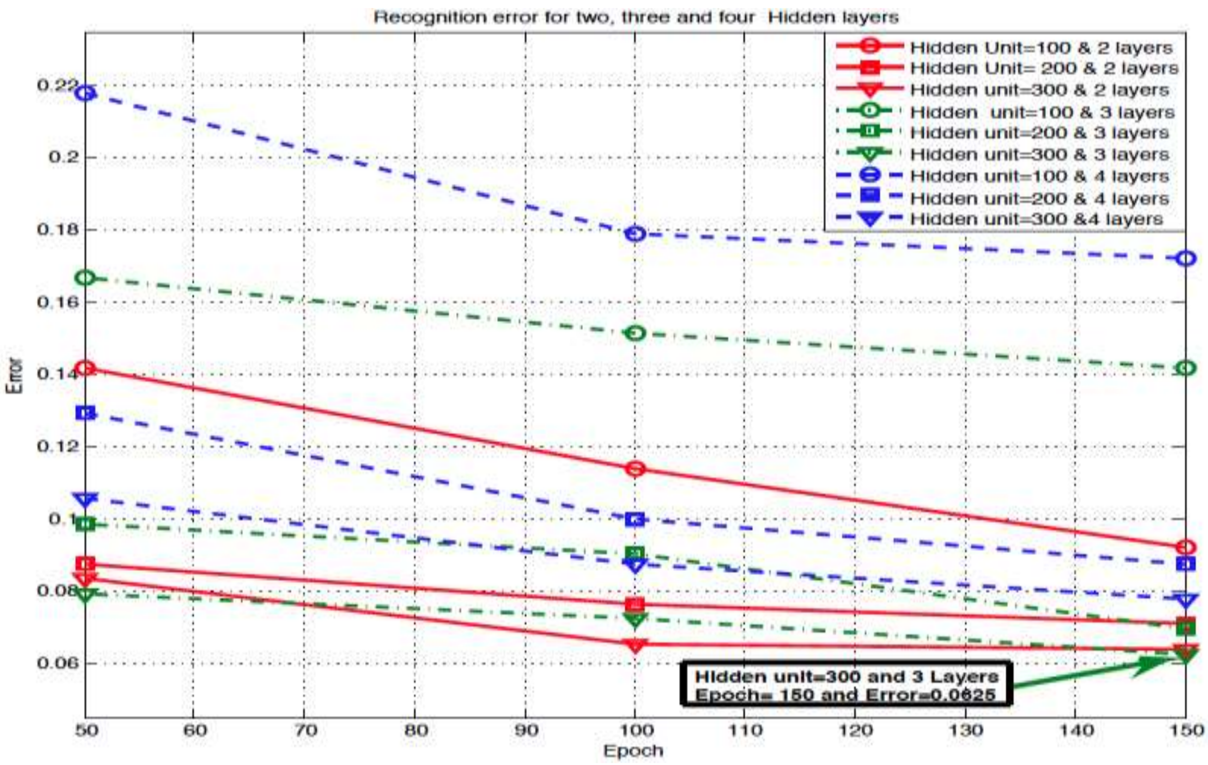



Figure 10: Graph of overall recognition error with two, three and four layers

**Comparison with other similar works**

To compare the results of our approach with others, three research results mentioned in [3, 10, and 11] were selected. The selection criteria was based on the fact that the researchers used handwritten Amharic text, otherwise since the documents are not very similar to ancient Ethiopic manuscripts we found it hard to make exact comparison. Table 4 shows rough comparison of the selected research results with that of ours.

As the comparison table shows the recognition rate of our system produced better results. The result obtained by the HMM system is comparable to ours because it was obtained with considerably smaller number of training words of good quality documents.

Table 4 Performance comparison of different approaches

| Research | Document type | No. of pages or characters used in dataset | Recognition Rate |
|---|---|---|---|
| HMM based [3] | Modern Handwritten | 100 words | 93% (for good quality document) |
| | | 10,932 words | 76%(for good quality document) |
| Writer Independent [26] | Church document | 114 pages | 87% |
| Lexicon Based [10] | Modern handwritten | 307 pages | 87% (top -5 choices) |
| **Our approach** | **Ancient manuscript** | **200 pages** | **93.75%** |

## CONCLUSION AND FUTURE WORKS

This research work aimed to propose a system for recognition of Ancient Ethiopic manuscript CR using DBNs. The recognition system consists of data acquisition, preprocessing, segmentation, and classification and testing the recognition. We have prepared dataset consisting of 24 base characters of 100 frequencies; totally 2400 characters are prepared in mat file. Among the total characters 1680 used for training and 720 characters used for testing the system.

The classifier network (DBN) was trained using a data set of Geez characters using RBM greedy layer wise unsupervised training and Soft max supervised fine tuning for the final RBM layer.

In our experiment we have tested 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.

Even though the dataset frequency for each character was low to make very good training, we have seen that DBN is an excellent technique for ancient handwritten document CR with very good accuracy.

Even though we have shown that DBNs are performing very well in recognition accuracy for ancient manuscript characters recognition task, there are issues that can addressed in future research activities. Some of these issues are:

In order to design a complete recognition system for ancient Ethiopic manuscript characters, a complete dataset should be prepared that consists of the entire Geez alphabets by collecting several manuscripts from different sources. The network model parameters of the DBNs can be varied and experimented for optimal performance in the accuracy of the recognition.

Due to computational resource limitation, we have only trained the system to a maximum epoch of 150. By including regularization methods into the training to avoid over fitting, the network can be trained for more number of epochs and its performance can be improved with larger and computationally capable environment.

The recognition is highly dependent on the accuracy of pre-processing and segmentation for automatic manuscript recognition task. Additionally, we haven't included automatic text region detection to exclude non-text areas and segmenting multi-column text areas. Therefore, the research can be extended to include text area detections and improvements of preprocessing including image quality enhancing along with high performance segmentations.

## REFERENCES

[1]. Graves, Alex, et al, "Unconstrained on-line handwriting recognition with recurrent neural networks," *Advances in neural information processing systems*. 2008.

[2]. Prasad, Kauleshwar, et al, "Character recognition using matlab's neural network toolbox," *International Journal of u-and e-Service, Science and Technology* 6.1 (2013): 13-20.

[3]. Assabie, Yaregal, and Josef Bigun, "Offline handwritten Amharic word recognition," *Pattern Recognition Letters* 32.8 (2011): 1089-1099.

[4]. Gupta, Anshul, Manisha Srivastava, and Chitralekha Mahanta, "Offline handwritten character recognition using neural network," *2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*. IEEE, 2011.

[5]. Scelta, Gabriella F., and Pilar Quezzaire-Belle, "The Comparative Origin and Usage of the Ge'ez writing system of Ethiopia," *Unpublished manuscript, Boston University, Boston. Retrieved July* 25 (2001): 2009

[6]. Assabie, Yaregal, "*Optical Character Recognition of Amharic text: an Integrated Approach*," Diss. Addis Ababa University, 2002.

[7]. Meshesha, Million, and C. V. Jawahar, "Optical character recognition of Amharic documents," *African Journal of Information & Communication Technology* 3.2 (2007).

[8]. Cowell, John, and Fiaz Hussain, "Amharic character recognition using a fast signature based algorithm," *Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003.* IEEE, 2003.

[9]. Yuan, Aiquan, et al, "Offline handwritten English character recognition based on convolutional neural network," *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 2012.

[10]. Assabie, Yaregal, and Josef Bigun, "Lexicon-based offline recognition of Amharic words in unconstrained handwritten text," *2008 19th International Conference on Pattern Recognition*. IEEE, 2008.

[11]. Assabie, Yaregal, and Josef Bigun, "Writer-independent offline recognition of handwritten Ethiopic characters," *Proc. 11th ICFHR* (2008): 652-656.

[12]. Diem, Markus, and Robert Sablatnig, "Recognizing characters of ancient manuscripts," *Computer Vision and Image Analysis of Art*. Vol. 7531. International Society for Optics and Photonics, 2010.

[13]. Shah, Mansi, and Gordhan B. Jethava, " A literature review on hand written character recognition," (2013).

[14]. Bengio, Yoshua, et al, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*. 2007.

[15]. Bengio, Yoshua, "Practical recommendations for gradient-based training of deep architectures," *Neural networks: Tricks of the trade*. Springer, Berlin, Heidelberg, 2012. 437-478.

[16]. Ramappa, Mamatha Hosalli, and Srikantamurthy Krishnamurthy, "Skew detection, correction and segmentation of handwritten Kannada document," *International Journal of Advanced Science and Technology* 48 (2012).

[17]. Sahu, Vijay Laxmi, and Babita Kubde, "Offline handwritten character recognition techniques using neural network: A review," *International journal of science and Research (IJSR)* 2.1 (2013): 87-94.

[18]. Dave, Namrata, "Segmentation methods for hand written character recognition," *International journal of signal processing, image processing and pattern recognition* 8.4 (2015): 155-164.

[19]. Dongre, Vikas J., and Vijay H. Mankar, "Devnagari document segmentation using histogram approach," *arXiv preprint arXiv: 1109.1247* (2011).

[20]. Anupama, N., Ch Rupa, and E. Sreenivasa Reddy, "Character segmentation for Telugu image document using multiple histogram projections," *Global Journal of Computer Science and Technology* (2013).

[21]. Arica, Nafiz, and Fatos T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *IEEE Transactions on*

*Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 31.2 (2001): 216-233.

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

[23]. Larochelle, Hugo, et al, "Exploring strategies for training deep neural networks," *Journal of machine learning research* 10.Jan (2009): 1-40.

[24]. Hinton, Geoffrey E, "A practical guide to training restricted Boltzmann machines," *Neural networks: Tricks of the trade*. Springer, Berlin, Heidelberg, 2012. 599-619.

[25]. Wikibooks, "Artificial neural network." wikibooks.org, 2014. (Accessed on March 2016).

[26]. Getu, Siranesh, "*Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Network,"* Diss. Addis Ababa University, 2016.