

# HANDWRITTEN GE'EZ CHARACTER RECOGNITION USING ARTIFICIAL NEURAL NETWORK

Achamie Aynalem<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer engineering, Debre Tabor University, Debre Tabor, Ethiopia

## Abstract

*In this paper, I've used artificial Neural Networks to recognize handwritten Ge'ez characters. In the paper, I've used one or two hidden layer feed forward Neural Network architecture with different learning algorithms to study the best performance parameters. I've made MATLAB user interface with adjustable network parameters to study the best performance by varying network parameters. I've used MATLAB image processing techniques for basic image preprocessing operations like noise removal, normalization, binarization, adaptive background removal and image border enhancement. Characters are segmented from the processed image and encoded to binary vector and fed to the multilayer ANN for training. After training the network with handwritten Ge'ez characters, a sample test data is fed to the network and tested for the performance of the network.*

**Keywords** - handwritten character recognition, neural network, multilayer neural network, Ge'ez character recognition, image processing.

\*\*\*

## 1. INTRODUCTION

Character recognition is a process that allows computers to recognize handwritten or printed characters available mostly in unreadable image format and to change them into a format that the computer can use. In this paper, character recognition involves recognition of characters which are printed, scanned or captured with a digital camera; and convert them into a text format.

Artificial Neural Network (ANN) is an information processing model which is inspired by the information processing mechanism of biological nervous systems, such as the brain. It is composed of a large number of interconnected processing units (artificial neurons) working together in parallel to perform a task [10]. Like people learn by example, Artificial Neural Networks learn by example. In living organisms' nervous system, learning involves adjustment of synaptic connections available between neurons. Likewise, learning in ANN is held by adjusting weights of synaptic connections between artificial neurons [1].

Neural networks are particularly applicable in areas with imprecise data or applications which are too complex for humans to understand or to describe using some mathematical model. Pattern recognition or data classification, through a learning process, is one of the major applications of ANN.

In recent years, Neural Networks have been used for character recognition though; these recognition techniques have been mostly confined to English characters

There are efforts on character recognition for English [2], Chinese, Japanese and also for Farsi, languages. However, character recognition system is not attempted for Ge'ez language, which is the second largest written Semitic language next to Arabic; and which have a special importance for religion, science, human history and medicine [3].

One of the great applications of neural networks in pattern recognition is to recognize handwritten characters. Character recognition is a type of pattern recognition in artificial neural networks that identify characters in an image and convert them to a text format.

In character recognition, the initial input image should be processed in order to remove noise and unnecessary background images, to adjust color and contrasting. Then the individual characters are segmented followed by encoding those segmented characters to a vector of 0's and 1's. The neural network can train from these vectors and a target input given for them for each encoded character.

## 2. GE'EZ ALPHABET

Ge'ez is one of the most ancient human languages which were spoken in Northern Ethiopia and Eritrea. Reflections

of those human civilizations in the region were written in Ge'ez; and many of those ancient scientific, historical and religious books are found on different parts of the world written in Ge'ez language mainly in Ethiopia, Eritrea and in many other countries including USA, Germany, Italy and England [3].

And in history, different lost sacred books of Christianity and Judaism are recovered from Ethiopian Jewish and Christians from their Ge'ez version. Nowadays, Ge'ez language has no speaker as a mother tongue language but used only as a liturgical language in Ethiopian and Eritrean Orthodox and Catholic churches. However, these days several languages such as Ethiopian Amharic, which is the second widely used Semitic language next to Arabic, Tigrinya in Northern Ethiopia and Eritrea use Ge'ez characters for writing [4].

Therefore, besides to the today's spoken languages of Amharic and Tigrinya, devising character recognition of Ge'ez characters will help to retrieve ancient historical, scientific, secret and sacred books written in Ge'ez all around the world.

The Ethiopian Amharic language has taken the basic Ge'ez alphabet and numeric symbols.

There are (32\*7=224) regular alphabetic symbols in Amharic.

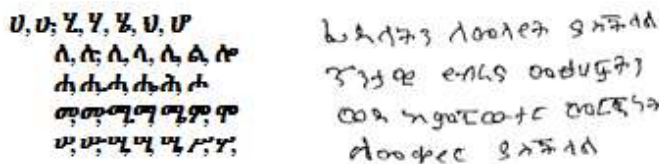


Fig 1: Sample Amharic characters Fig. 2 sample handwritten text in Amharic

In Ge'ez alphabet, there are no vowels; rather, vowel combination of each 32 regular characters is represented by different symbols. Based on that, each base character has 7 variants/symbols.

For example, the first Amharic base alphabet “ሀ” has the following 7 variants.

ሀ=hä, ሁ=hu, ሂ=hi, ሃ=ha ሄ=hie, ህ=h, ሆ=ho which results in 7 variants for only the first Amharic alphabet “ሀ”.

Therefore, there will be a total of 32\*7=224 regular symbols in Amharic character set; and there are around 21 numeric characters in Ge'ez, resulting in a total of 245 characters. Since the number of neurons at the input and output layers

of the neural network used for training these characters is proportional to the number of symbols in the language, the neural network will be complex and slow for training.

### 3. CHARACTER RECOGNITION

Character recognition is the conversion of images of handwritten or printed texts into machine-readable text formats. In the past years, recognition of characters from a scanned image using neural networks has been an active research area. The process of character recognition using artificial neural networks involves extracting of each image fragments that resemble characters from a scanned image and approximating those image fragments to the nearest character based on previously trained “knowledge” of image pattern-to-character mapping

For example, from a human conscious, if you see a randomly deformed or partially incomplete handwritten character like “a”, you could easily identify to the correct alphabetic character, by comparing the observed character with the available set of characters acquired previously through a long learning process. Likewise, neural networks have a little “intelligence” in associating some sort of pattern with the associated information or data through a learning process. This pattern recognition has got a special attention as a field of study in artificial intelligence and machine vision.

The character recognition process involves differentiating image fragments that represent characters from a scanned image and determine/approximate those letters which are represented in the light and dark areas of those fragments based on the learned associations.

Recognizing the right character especially from handwritten fonts is always a problematic area since different individuals have different handwriting styles; with some write/lay characters in a different shape and size, some write characters with slanted angles, others may write partially incomplete characters or leaving some minor details. This creates a new experience for neural network based character recognition algorithms resulting wrong interpretation of those handwritten characters. Many researches have been made to improve the performance of such algorithms. As mentioned above, a widely accepted approach which can improve the performance of handwriting recognition is with the use of neural networks.

Neural networks should be trained first using a sufficient dataset that associates the input image pattern with the associated character. If a neural network is trained with a sufficient dataset, it can recognize the pattern with good end results. Therefore, neural networks are used in situations where there is sufficient dataset for the training process.

As shown above, in this paper, Ge'ez characters in the Amharic language have many variants some of which look quite similar.

For example, if you look Amharic characters “ሀ”(hä) and “ሁ”(hu), they have similar general shape except for the tiny front extension at the second character. Of course, all the 32 regular alphabets have seven resembling variant symbols which have different sounds in the language.

Therefore, we need somewhat complex neural network architecture with a relatively larger number of input and hidden layer nodes to capture these minor details in Ge'ez alphabet.

In character recognition using neural networks, first, we have to find the image that contains the desired characters using a scanner or photo camera. After the image is accessed, we define processing algorithm, which will extract/identify important fragments that represent characters from the image, usually represented with a unique numerical array stored as a matrix and map those numerical arrays into a dataset of characters. With these sufficient amounts of dataset pairs, a neural network can be trained first; and later, it can identify the right character for a random input image vector.

Character recognition involves several stages which involve image pre-processing, segmentation, feature extraction, post-processing, and classification.

## 4. IMAGE PREPROCESSING

In character recognition, the image captured through a camera or scanner may not have the required quality for extracting features that represent characters. Therefore, first the acquired image should pass through a series of image pre-processing algorithms in order to;

- ✓ Remove noise incorporated during capturing through camera or scanning, due to old aged paper document or digital noise.
- ✓ Remove unnecessary background image from the image part that represents linguistic characters
- ✓ Improve edges of images that represent characters
- ✓ Replace colored or grayscale images with binary (black and white scale) images since color have no significant importance to recognize handwritten characters, and black and white scale images have smaller data size which reduces processing time and size of a neural network.

### 4.1 Image Representation with Gray Scale

In Intensity/grayscale, an image is represented by a single 2-dimensional matrix where each value in the matrix

represents intensity scale (Gray level) of a pixel available in the image.

### 4.2 Noise Removal

Images acquired through scanner or camera are always prone to noise due to different factors like capturing through camera or scanning, or capturing from old aged paper documents.

I've used an adaptive linear filter that removes noise based on the local variance in the image using Wiener 2 filter [6]. This filter has a special feature of removing white Gaussian noise without smoothing or blurring effect on the image.

### 4.3 Contrast Adjustment and Background Removal

Grayscale images found from a scanned document or taken with a camera may be associated with background. Sometimes, those background shades may have different intensity in different regions of the image. This creates a problem while converting a grayscale image to binary image. Image regions that represent characters should be made distinct from the remaining part of the image by removing the background and smoothing cluttered border. In this paper, I have used a special image processing procedure in order to remove unnecessary background images and smoothen corrugated borders. The procedure is given as follows;

### 4.4 Adaptive Background Removing

- ✓ In grayscale image of handwritten text, since texts are written in black (low intensity), the standard deviation in the intensity of pixels in regions of the image which contain texts will be relatively large.
- ✓ Pixels that represent characters have pixels below or equal to the mean, and background pixels will have intensities above the mean.

Therefore, by using the standard deviation and mean of pixel intensities of a region in an image, it is possible to classify the intensities of pixels to the right intensity level (to the background pixel intensity or text pixel intensity) using intensity adjustment operations. However, the standard deviation and mean values will be different for different regions in an image depending on the background variation and pattern of the textual image available in the region. Therefore, the algorithm is implemented starting from the first pixel in the image, calculate the approximate intensity of the pixel from a small number of its neighborhood using those stated sequences of operations, and repeating the algorithm for each pixel in the image using sliding neighborhood operation.

Further adjustment of border pixels of each segment in sliding operation will produce a good result in removing the variable background in an image.

### 4.5 Convert To Binary Scale

The processed grayscale image is converted to binary scale image using adaptive binarization.

### 4.6 Skeletonization

This process is used to remove unnecessary pixels which do not belong to the backbone of the character which results in reducing the broad strokes to thin lines that represent the backbone of characters. This skeletonization process is used after the image is converted to a binary image.

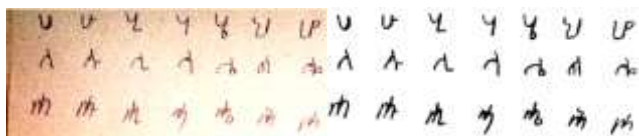


Fig 3: A) captured image before pre-processing B) image after pre-process

## 5. FEATURE EXTRACTION

The next phase of character recognition is feature extraction. At this stage, image regions which contain characters are segmented in order to separate characters from each other. Then segmented characters are normalized to a standard pixel size. Always characters appear in different size in different images. This creates a problem unless those characters are normalized to a standard pixel size. Therefore, after characters are segmented, the individual character images are normalized to 7 by 5 pixels. The number of pixels for normalization depends mainly on two factors;

- i. The number of symbols available in Ge'ez alphabet. We have 245 symbols in Ge'ez. Therefore, we should select a pixel size greater than 8 ( $2^8=256>245$ ).

Normalization with pixel size below 8 will result in different characters represented with the same bit sequence during encoding. However, for handwritten characters, some details may resemble due to some imperfections during image preprocessing stages. Therefore, I've selected larger pixel size ( $10*10=100$ ) in order to capture small variations among different characters in the language.

- ii. The number of neurons in the input layer of our neural network depends on the number of pixels in the input data. In order to reduce the training time, I've selected the minimum possible pixel size which is 100 pixels.

Line thickness of normalized characters is further adjusted by dilating.



Fig 4: extracted characters after preprocessed

*Encoding:* Since our image is a binary image, pixel intensities are either 0 or 1. Therefore, segmented characters are encoded to a sequence of 0's and 1's by converting matrices of those characters sub-images into a column vector. In such a way, each segmented character is represented uniquely by a sequence of 100 bits.

## 6. TRAINING NEURAL NETWORK

To train a neural network, we need to associate the target data for each input pattern of characters. This is done by associating the target character for each input character pattern. Therefore, the neural network can be trained by input matrix constructed from encoded input character patterns and target matrix constructed from encoded target characters given manually by the user for respective input patterns.

I've used one hidden layer feed forward neural network with tangent sigmoid activation function.

The number of input neurons depends on the size of the input vector. In this paper, I've represented each character with a vector of 100 bits. Therefore the number of input neurons is 100.

The number of neurons in the hidden layer will depend on the complexity of the decision boundaries needed for the pattern recognition task. This is something that cannot generally be known before training. I've started with 10 neurons in the hidden layer and then tested the network performance after training.

The number of neurons at the output layer depends on the number of bits used to represent characters.

In Amharic language, there are 245 basic symbols. Therefore, 8 bits are sufficient to represent each of these characters uniquely since  $2^8=256$  is greater than 245. The minimum possible number of neurons at the output layer, therefore, should be at least 8.

In this paper, I've used 9 neurons at the output layer. For example, Amharic character, "U" is encoded to a

hexadecimal value of 000h, “U” to 001h, “Z” to 002h value and so on.

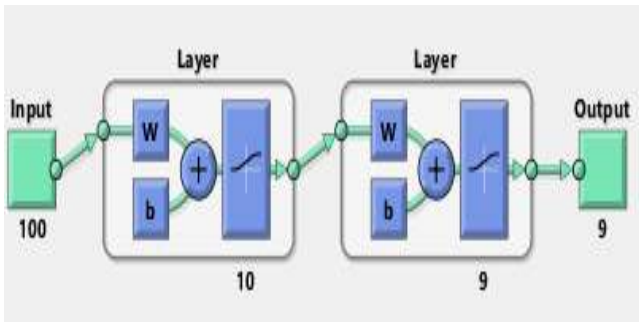


Fig 5: 2-layer feed forward network for training Ge’ez characters

Finally, a testing sample is given for the network and evaluated for the output matrix. Then the output matrix is decoded to Amharic characters and written to a file.

### 7. USER INTERFACE

For this research, I’ve created a graphical user interface using MATLAB GUI. In the interface, I’ve created input boxes so that the user could be able to vary different parameters of image processing and neural network parameters. This enabled me to study the performance of the system under different circumstances and network parameters to adjust to the best performance situation.

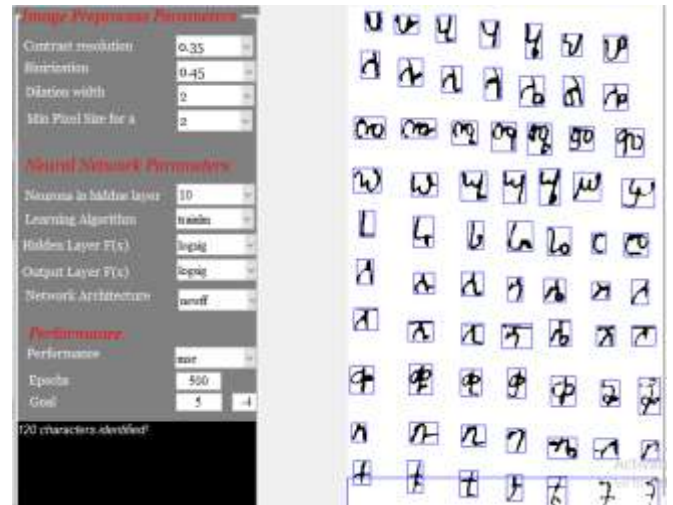
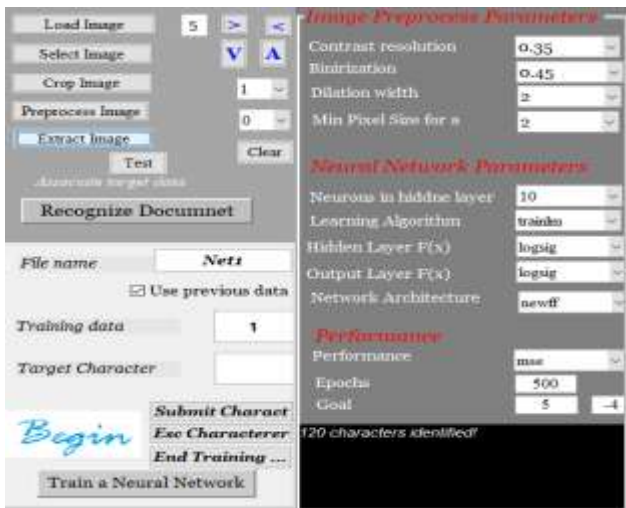


Fig 6: User interface for testing character recognition

### 8. RESULTS

A colored image is captured using a digital camera. Before feature extraction, the image is pre-processed through image processing algorithms to reduce irrelevant information or noise and to improve the image properties that will make feature measurement easier and reliable.

Then, features are extracted from the image, and some features which have less than 30 pixels and above the maximum pixel size are removed from the feature list since those may not represent characters. In addition, algorithms are implemented to identify characters in different lines and the space between words.

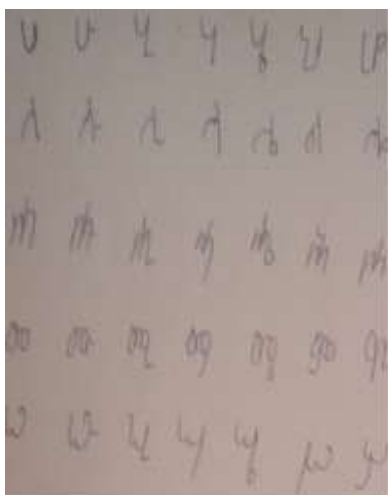
Each of the extracted features is then changed to a column vector, and a single matrix is constructed from column vectors of all the extracted features.

During constructing the training matrix, the user is requested to submit the target character for each of the extracted training features. From the set of submitted target characters, the target training matrix will be generated by a MATLAB routine.

Finally, the neural network parameters are adjusted through the indicated user interface boxes; the two matrices (training matrix of extracted features and the generated target matrix) are fed for the selected neural network, and trained by clicking the “train neural network” button shown above.

Once the network is trained, it can be used to test the performance of the network for another set of character images using the “Recognize” button.

Using this simulation prototype, I've tested for different character set and different parameter sets



A) Image before preprocessing

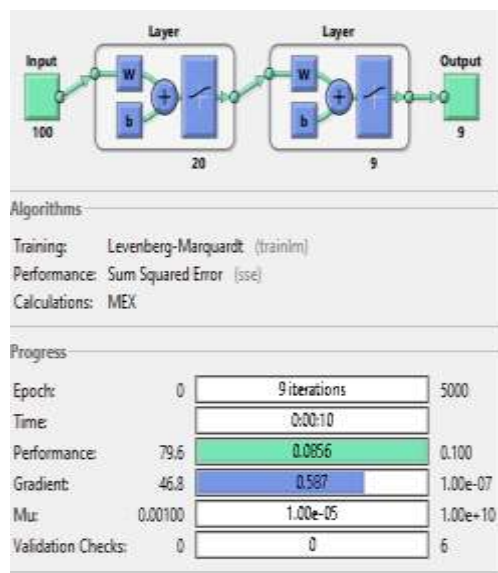


B) Image after preprocess

Fig 7



A) Segmented Ge'ez characters



B) training the network

Fig 8

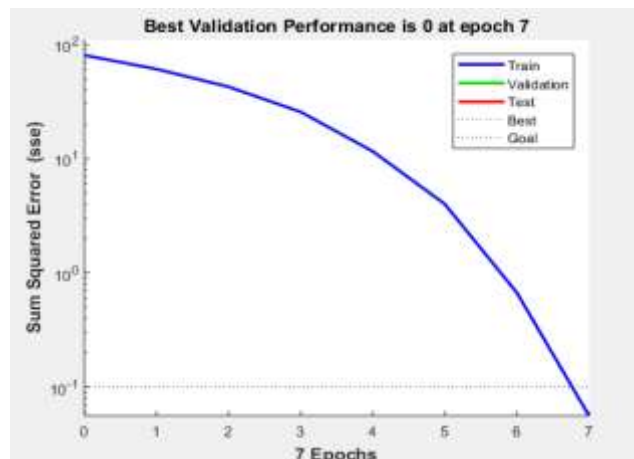


Fig 9: Performance measures of the training process

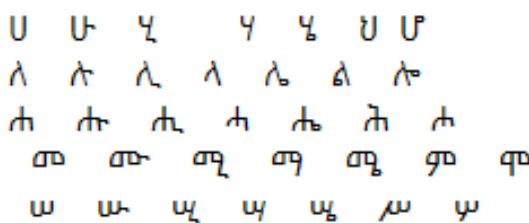


Fig 10: recognized characters saved as text format

It takes 30 seconds to train a total of 35 characters with 10 neurons at the hidden layer. The resulting accuracy is 81%

For 20 neurons at the hidden layer, it took below 5 seconds and 6 epochs with an accuracy of 100% as shown in figure 9 and 10 above.

For 30 neurons at the hidden layer, it took below 3 seconds and 5 epochs with an accuracy of 100%.

But above 35 neurons at the hidden layer, the network took large time for training but it saturated relatively at lower epochs.

For example, for the same input image with 60 neurons at the hidden layer, it took 42 seconds for training but stopped at the 4th epoch with an accuracy of 100%.

For 49 characters/features, with 60 neurons at the hidden layer, it took about 2 minutes saturated at the 8th epoch resulting an accuracy of 100%.

Therefore, for a given training image, there is an optimum number of neurons at the hidden layer of the feed-forward neural network. The greater the number of characters/features in the input training image, the larger number of neurons will be needed at the hidden layer. In fact, in order to capture the different variant characters or features of the input image, there should be sufficient amount of neurons at the hidden layer.

## 9. CONCLUSION

The paper gives a useful method for the recognition of handwritten Ge'ez characters to a great extent. The proposed method has been applied to different unknown handwritten characters. Neural network based method gives the accuracy well above 90%. However, in this process, the accuracy falls below 85% when the training character set and testing character sets have large difference in the size of input images.

This neural network based method of text recognition has accuracy almost 100% for printed texts. Therefore, this method could be employed for retrieving documents written in Ge'ez fonts for Amharic or Tigrinya languages or to retrieve the abundant ancient historical, scientific and sacred books written in Ge'ez language which are currently found in different parts of Ethiopia and Eritrea. Those ancient books found in Ethiopia and Eritrea are located in unfavorable remote areas which are vulnerable to loss and damage. Therefore, retrieving those materials by scanning and converting them to text format would be valuable for passing those materials for the next generation.

## REFERENCES

- [1] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale and Orlando De Jesús, *Neural Network Design*, 2nd Edition, eBook.
- [2] Chirag I Patel, Ripal Patel and Palak Patel, *Handwritten Character Recognition using Neural Network*, International Journal of Scientific and Engineering Research Volume 2, May-2011
- [3] Yemisrach Benalfew. (1999). *Ethiopian historical antiques*. retrieved April 29, 2018, <http://www.twn.my/title/1955.htm>
- [4] Alejandro Gutman and Beatriz Avanzati. (2013). *Semitic languages*. retrieved April 29, 2018, from <http://aboutworldlanguages.com/semitic-branch>
- [5] Heikki N. Koivo, *Neural Networks: Basics using MATLAB Neural Networks*, 2008
- [6] Alasdair McAndrew, *An introduction to digital image processing with MATLAB*,
- [7] *Image processing toolbox*, The MATHWORKS.inc
- [8] Shashank Araokar, *Visual Character Recognition using Artificial Neural Networks*, University of Mumbai, India
- [9] Reza gharoie ahangar and Mohammad Farajpoor Ahangar, *Handwritten Farsi Character Recognition using Artificial Neural Network*, International Journal of Computer Science and Information Security (IJCSIS), August 2009
- [10] Christos Stergiou and Dimitrios Siganos, *Neural Networks*, retrieved, April 1, 2018, [https://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.htmlpril](https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.htmlpril)