A Hybrid System for Robust Recognition of Ethiopic Script

Yaregal Assabie and Josef Bigun
School of Information Science, Computer and Electrical Engineering
Halmstad University, SE-301 18 Halmstad, Sweden
{Yaregal.Assabie, Josef.Bigun}@ide.hh.se

Abstract

In real life, documents contain several font types, styles, and sizes. However, many character recognition systems show good results for specific type of documents and fail to produce satisfactory results for others. Over the past decades, various pattern recognition techniques have been applied with the aim to develop recognition systems insensitive to variations in the characteristics of documents. In this paper, we present a robust recognition system for Ethiopic script using a hybrid of classifiers. The complex structures of Ethiopic characters are structurally and syntactically analyzed, and represented as a pattern of simpler graphical units called primitives. The pattern is used for classification of characters using similarity-based matching and neural network classifier. The classification result is further refined by using template matching. A pair of directional filters is used for creating templates and extracting structural features. The recognition system is tested by real life documents and experimental results are reported.

1. Introduction

Ethiopic script is a writing system used mainly in Ethiopia by several languages like Geez, Amharic, Tigrigna, Guragegna, etc. Since the 19th century, the Ethiopic alphabet has been largely used by Amharic, which is the official language of Ethiopia and the second most spoken Semitic language in the world next to Arabic. The recently standardized alphabet has a total of 435 characters. However, the most commonly used alphabet, which has 34 base characters and seven orders, is conveniently written in a tabular format as shown in Table 1. The first order in the table represents the base character and other orders are modifications that represent vocalized sounds of the base character.

Research and development of automatic recognition of characters started in the 1950s. Since then, many studies have been done for recognition of various scripts like Latin, Arabic, Chinese, and Japanese scripts [10]. However, the technology for recognition of Ethiopic characters is far behind, with the first published work appearing only recently [7]. Moreover, the structural complexity and interclass similarity of Ethiopic characters pose an additional difficulty for classifiers.

Table 1. Part of the Ethiopic alphabet

	Base Sound	Orders						
		1st (ä)	2nd (u)	3rd (i)	4th (a)	5th (e)	6th (ə)	7th (o)
1	h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
2	l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
3	h	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
4	m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
5	s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
6	r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
7	s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
8	š	ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
9	q	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
10	b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
31	p'	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
32	f	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
33	p	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
34	v	ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ

A number of different approaches have been proposed to solve character recognition problems. However, most of them are grouped into one of the following four important recognition techniques: *template matching*, *structural and syntactic*, *statistical*, and *artificial neural networks*. Template matching is one of the simplest and earliest methods where the character to be recognized is matched against a database of stored templates of characters. Structural and syntactic methods analyze the interrelationships

Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)
0-7695-2822-8/07 $25.00 © 2007 **IEEE**

between simpler sub-patterns and recognition of characters is made based on a set of syntax rules. Statistical techniques extract numerical feature data from characters and the recognition can be regarded as a process of feature space partitioning [9]. Artificial neural networks share the properties of biological neural networks and they use mathematical or computational model for classification [8]. Each of the recognition techniques has its own strengths and limitations, and hybrid systems draw upon the synergy effect of two or more techniques [6].

In this paper, we present a hybrid recognition system where Ethiopic characters are structurally and syntactically analyzed to generate patterns of primitive structural features. Similarity-based pattern matching and artificial neural networks are used for classification of the unknown input. For confusing characters, the result is further refined by template matching technique. The hybrid recognition system efficiently recognizes documents with various font types, sizes, and styles.

2. Structural and syntactic analysis of Ethiopic characters

We use a structural and syntactic technique to model the spatial relationships of primitive structural features of Ethiopic characters. The model handles variation in the size of a character because we encode the relative size of structural features. The structural and syntactic model of Ethiopic characters is given below and further exposed in detail in [2].

Prominent structural features in Ethiopic characters form a set of *seven* primitive structures. The primitives differ from each other by their relative length, orientation, spatial position and structure. The classes of primitive structures (with example characters) are: *long vertical line* (ዘ), *medium vertical line* (ሰ), *short vertical line* (ሐ), *long forward slash* (ጊ), *medium forward slash* (ፊ), *backslash* (ል) and *appendages* (ቲ).

Horizontal lines connect two or more of these primitive structures to form the overall complex structure of characters. Connections between primitives occur at one or more of the following three connection regions: *top (t)*, *middle (m)*, and *bottom (b)*. The first connection detected as one goes from top to bottom is considered as *principal* and the other connections, if there exist, are *supplemental* connections. A total of 18 connection types are identified between primitives of the Ethiopic characters and a summary is given in Table 2. The principal connection is used to determine the spatial relationships between two primitives. Two connected primitives α and β are represented by the pattern α**z**β, where **z** is an ordered pair (x,y) of the connection regions *t*, *m*, or *b*. In this pattern, α is connected to β at region x of α, and β is connected to α at region y of β.

Table 2. Connection types between primitives

Principal Connection	Supplementary Connections					
	None	(m,b)	(b,m)	(b,b)	(m,m)+(b,m)	(m,m)+(b,b)
(t,t)						
(t,m)						
(t,b)						
(m,t)						
(m,m)						
(m,b)						
(b,t)						
(b,m)						
(b,b)						

The spatial relationships of primitives representing a character are modeled by a special tree structure as shown in Fig. 1. Each node in the tree stores data about the primitive and its connection type with its parent node. The child nodes correspond to the possible number of primitives connected to the parent primitive. Primitive tree for a character is built first by selecting the left top primitive of a character as the root primitive. Then, based on their spatial positions, other primitives are appended to the tree recursively in the order of {*left*{top, middle, bottom}, *parent*, *right* {bottom, middle1, middle2, top}}.

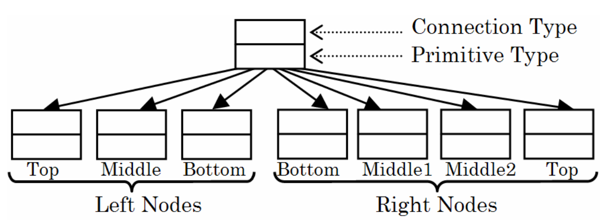

Figure 1. General tree structure of characters

We extract structural features by making use of a pair of directional filters both of which can be computed from direction field tensor. For a local neighborhood *f(x,y)* of an image *f*, the direction tensor **S**, is computed as a 2x2 symmetric matrix using derivative operators D_x and D_y [5].

$$S = \begin{pmatrix} \iint (D_x f)^2 dxdy & \iint (D_x f)(D_y f) dxdy \\ \iint (D_x f)(D_y f) dxdy & \iint (D_y f)^2 dxdy \end{pmatrix} \quad \textbf{(1)}$$

Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)
0-7695-2822-8/07 $25.00 © 2007 IEEE

From the direction field tensor **S**, we can compute the following images [4].

$$I_{10} = \iint ((D_x + iD_y) f) dxdy \quad \textbf{(2)}$$

$$I_{11} = \iint \left| (D_x + iD_y) f \right|^2 dxdy \quad \textbf{(3)}$$

$$I_{20} = \iint ((D_x + iD_y) f)^2 dxdy \quad \textbf{(4)}$$

I_{10} is equivalent to the ordinary gradient field in which the angle shows the direction of intensity differences in a local neighborhood of an image, and the magnitude shows the average change in intensity. I_{11} measures the optimal amount of gray value changes in a local neighborhood. I_{20} is complex valued where its argument is the optimal local direction of pixels in double angle representation and its magnitude is measure of the local LS strength. In Fig. 2, I_{10} and I_{20} images are displayed in color where the hue represents direction of pixels with the red color corresponding to the direction of zero degree.

Figure 2. (a) Ethiopic text, (b) I_{10} of *a*, (c) I_{20} of *a*, *(d)* I_{11} of *a*

We use I_{10} and I_{20} to extract structural features in characters. Since I_{20} provides optimal direction of pixels, it is used to effectively group pixels into parts of primitives and connectors based on the direction information. In Fig. 2, it can be seen that a primitive structure in the original text image results in two lines (at left and right edges) in the I_{10} and I_{20} images. Since the left and right edges have opposite directions in the I_{10} image, it provides more convenient information for extraction of structural features. Therefore, primitive structures are formed from two matching left and right lines in the I_{10} image. The basic algorithms for extracting structural features are discussed in [3].

Extraction of the structural features is done on segmented characters. The horizontal area that lacks LS in the I_{20} image segments text lines, and the vertical area that lacks LS segments individual characters within each text line. Fig. 3 shows primitive structures extracted from the document image in Fig. 2a.

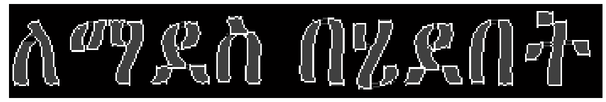

Figure 3. Extraction of primitives

3. The hybrid recognition system

In Section 2, we discussed about the structural and syntactic model that uses a special primitive tree to represent characters in terms of the spatial interrelationships of primitive structures. Data stored in the primitive tree of a character is converted into one-dimensional string by recursively traversing the tree in the order of {**left**{top, middle, bottom}, **parent**, **right**{bottom, middle1, middle2, top}}. This is similar to *in-order* traversal of binary search trees and produces a unique string pattern for each primitive tree. The pattern generated from character images is used for similarity-based matching and as an input for the neural network system. The outputs of similarity-based matching and neural network classification are compared and the best result is selected. Then, template matching is applied for confusing characters. The general framework of the hybrid recognition system is shown in Fig. 4 and each component of the system is further explained below.

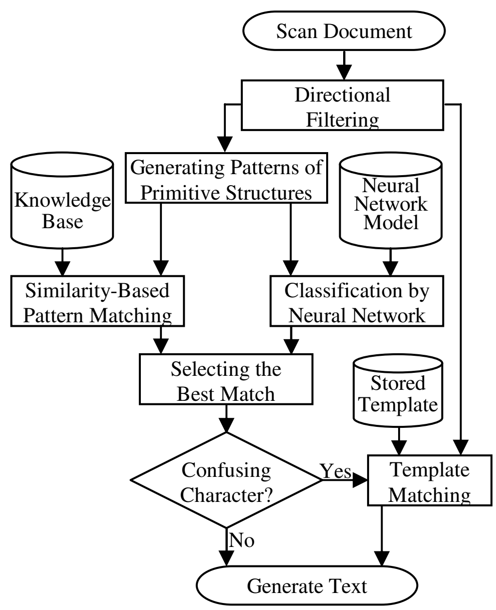

Figure 4. The hybrid recognition system

3.1 Similarity-based pattern matching

The cost of tree matching is more expensive in terms of time complexity than string matching [1]. Therefore, pattern matching is performed after the primitive tree is converted to its equivalent string pattern. For computation purpose, primitives and connectors are assigned unique two-digit numerical codes. The knowledge base of characters is also built from the possibly occurring string patterns of each character. We employ approximate pattern matching which finds the occurrence (similar pattern) of an input pattern within some threshold of error. Fig. 5 shows the a simplified approximate pattern matching algorithm used to find the most similar pattern in the knowledge base.

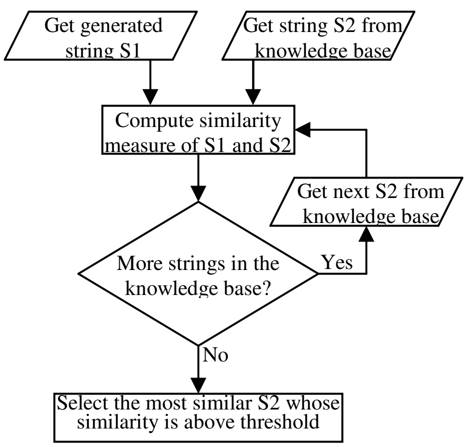

Figure 5. The pattern matching algorithm

3.2 The neural network classifier

Artificial neurons are typically organized into three layers: *input*, *hidden* and *output*. The input layer takes data of the unknown pattern whereas the output layer provides an interface for generating the recognition result. The hidden layer contains many of the neurons in various interconnected structures hidden from the outside view.

In this study, the inputs to the neural network system are patterns of primitives and their spatial relationships. Since neural networks take numerical data as an input, we assign binary numbers to primitives and their spatial relationships. We use 72 binary digits to encode patterns generated from the primitive tree. Each Ethiopic character in the alphabet is also encoded with a nine-digit binary number which is sufficient to represent all the 435 characters. Therefore, the neural network model has 72 input nodes and 9 output nodes. The hidden layer also has 72 nodes which is set optimally through experiments. The neural network model is further exposed in [4]. Fig. 6 illustrates the neural network model.

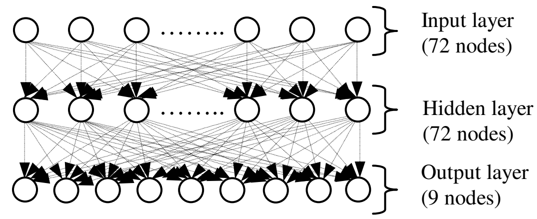

Figure 6. The neural network model

The training data of the neural network model is prepared from the possibly occurring patterns of primitives and their spatial relationships. Fig. 7 shows the first runs of the training progress using BrainMaker neural network tool. The graph depicts that the average error is declining over the course of training, which is a desired property of a good neural network model.

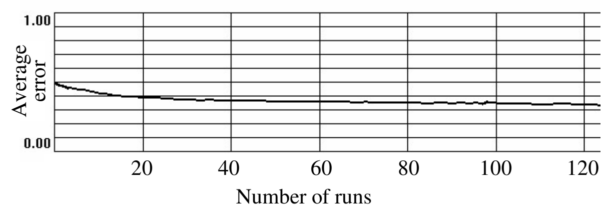

Figure 7. Progress of training

3.3. Template matching

There are a few set of characters like ሰ and ለ; ሸ and ኸ; ፉ and ፋ; ጥ, ፐ and ኘ; and ጎ and ጐ that confuse with each other during recognition. The confusion comes from the fact that they are structurally similar with each other. Their difference lies in the orientation or structure of their primitives. For such characters, the orientation and structure templates of the primitives that discriminate each other with in the groups are stored. After classification is made by the neural network, for such confusing characters, the result is further refined by matching the templates of specific discriminative primitives. Fig. 8 shows some examples of confusing characters and their primitive structures that are stored to discriminate the characters.

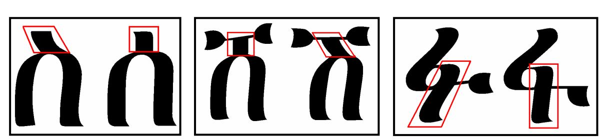

Figure 8. Confusing characters with their discriminative primitives

Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)
0-7695-2822-8/07 $25.00 © 2007 IEEE

4. Experiments

The recognition system works by setting the size of Gaussian window according to font sizes and document types. For clean documents, we used a window of 3x3 pixels for texts with font sizes of less than or equal to 12, a window of 5x5 pixels for font sizes of 16, and a window of 7x7 pixels for font sizes of 20. On the other hand, for most documents taken from books, newspapers, and magazines (font sizes are equivalent to about 12), a window of 5x5 pixels was used because of their higher level of noise.

We designed several neural network models by varying the input and output formats, the number of hidden nodes, and other training parameters. The model described in Section 3.2 is chosen based on the optimal results we obtained from training and testing procedures.

The robustness of the system is tested with about 68 images of documents taken from Ethiopic Document Image Database [4]. An average accuracy of 91% was obtained for noisy documents like books, newspapers and magazines. For clean printouts with Visual Geez Unicode font type and with 8, 10, 12, 16, and 20 font sizes, we obtained recognition rates of 94%, 95%, 95%, 96% and 96%, respectively. For printout documents with Visual Geez Unicode font type and 12 font size, recognition rates of 95%, 94% and 95% were achieved for normal, italic and bold font styles, respectively. The results for various font types, each with 12 font size are summarized in Table 3.

Table 3. Recognition results for different fonts

Font Type	Recognition (%)
Visual Geez Unicode	95
Visual Geez 2000 Main	95
Visual Geez 2000 Agazian	96
Power Geez	95
Geez Type	94

5. Discussion and Conclusion

The key requirement of recognition systems is to tolerate variations in the characteristics of documents. To this effect, we have designed a recognition system where various pattern recognition techniques are used at various stages of the recognition system. We use structural and syntactic analyzer to generate a pattern of primitive structures and their spatial relationships. A hybrid system of similarity-based matching and neural network classification play an important role in optimally recognizing the unknown input. The approximate pattern algorithm was efficient for structurally simple characters like ሀ. The neural networks tend to outperform similarity-based pattern matching for characters which are structurally complex and built from higher number of small-size primitive structures like . Template matching is finally applied to refine the result for confusing characters.

The recognition system tolerates documents with a skewness of up to 10^{o}. The common errors in the recognition process arise from extraction of primitive structures and segmentation of characters. We can further improve the recognition system by additional studies of these algorithms. Spell checker and parts of speech analyser can also help increase the recognition result.

References

[1] A. Sanfeliu, “Matching tree structures,” In H. Bunke and A. Sanfeliu (eds.), *Syntactic and Structural Pattern Recognition: Theory and Applications*, World Scientific, Singapore, 1990.

[2] Y. Assabie and J. Bigun, “Structural and syntactic techniques for recognition of Ethiopic characters,” In *Structural, Syntactic, and Statistical Pattern Recognition*, volume LNCS-4109, Springer, August 17-19, Hong Kong 2006, pp. 118-126.

[3] Y. Assabie and J. Bigun, “Ethiopic character recognition using direction field tensor,” In *Proc. of the ICPR2006*, IEEE Computer Society, August 20-24, Hong Kong 2006, pp. 284-287.

[4] Y. Assabie and J. Bigun, “A neural network approach for multifont and size-independent recognition of Ethiopic characters,” In *Proc. of the IWAPR2007*, Springer, July 21-23, Plymouth, UK, 2007.

[5] J. Bigun, *Vision with Direction: A Systematic Introduction to Image Processing and Vision*, Springer, Heidelberg, 2006.

[6] H. Bunke and A. Kandel (eds.), *Hybrid Methods in Pattern Recognition, World Scientific*, Singapore, 2002.

[7] J. Cowell and F. Hussain, “Amharic Character Recognition Using a Fast Signature Based Algorithm”, In *Proc. of the 7th Int’l Conf. Information Visualization*, 2003, pp. 384-389.

[8] G. Dreyfus, *Neural Networks: Methodology and Applications*, Springer, Heidelberg, 2005.

[9] A. Jain, R. Duin and J. Mao, “Statistical Pattern Recognition: A Review”, *IEEE-TPAMI*, vol. 22(1), 2000, pp. 4-37.

[10] C. Y. Suen, S. Mori, S. H. Kim, and C. H. Leung, “Analysis and Recognition of Asian Scripts - The State of the Art,” In *Proc. of ICDAR 2003*, vol. 2, Edinburgh, 2003, pp. 866-878.

Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)
0-7695-2822-8/07 $25.00 © 2007 IEEE