Recognition of Printed Amharic Documents

Million Meshesha, C. V. Jawahar
Center for Visual Information Technology,
International Institute of Information Technology - Hyderabad,
jawahar@iiit.net

Abstract

In Africa, there are a number of languages with their own indigenous scripts. This paper presents an OCR for Amharic scripts. Amharic is the official and working language of Ethiopia. This is possibly the first attempt towards the development of an OCR system for Amharic. Research in the recognition of Amharic script faces major challenges due to (i) the use of more than 300 characters in writing and (ii) existence of a large set of visually similar characters. In this paper, we propose a two-stage feature extraction scheme using PCA and LDA, followed by a decision DAG classifier with SVMs as the nodes. Recognition results are presented to demonstrate the performance on the various printing variations (fonts, styles and sizes) and real-life degraded documents such as books, magazines and newspapers.

1. African scripts

While the use and applications of OCRs are well developed for most languages in the world that use both Roman and non-Roman scripts [1, 2], there is limited research effort in this direction for the indigoenous scripts of African languages [3]. Africa is the second largest continent in the world, next to Asia. Document analysis research has not yet addressed the indigenous African scripts, as much they deserve. There are more than 55 independent countries in Africa with approximately 800 million people and over 800 ethnic groups. Its many languages testify to the vast diversity of the African people. In all, more than 2,500 languages (including regional dialects) are spoken in Africa. Some are indigenous languages, while others are installed by conquerors of the past. English, French, Portuguese, Spanish and Arabic are official languages of many of the African countries. Most African languages with a writing system use a modification of the Latin and Arabic scripts. However, there are also many languages in Africa with their own indigenous scripts that vary considerably in shapes. Some of these scripts include Amharic script (Ethiopia), Bassa script (Liberia), Mende script (Sierra Leone), Vai script (West Africa) and Meroitic script (Sudan) [4]. Among these, Amharic is the only language that is used as an official and working language since the 14th century [5]. It is the most commonly learned language next to English throughout the country. Amharic script is used for writing in the various languages in Ethiopia and Eritrea, including Amharic, Tigre and Tigrigna.

There is a bulk of printed documents (such as correspondence letters, newspapers, magazines, and books) available in government and private offices, libraries and museums. Digitization of these documents enables to harness already available information technologies to local information needs and developments. Those African languages that use modified scripts of Latin and Arabic language can be integrated to the existing Latin and Arabic OCRs with the same additional language processing modules. Therefore, we give more emphasis to indigenous African scripts. This is the motivation behind the present work. To the best of our know-ledge, this is the first work that reports the challenges towards the recognition of indigenous African scripts and a possible solution for Amharic script.

Table 1. Total number of symbols in Amharic writing system (FIDEL)

No.	Type of Amharic Characters	Number of Characters
1	Core characters	231
2	Labialized characters	51
3	Punctuation marks	8
4	Numerals	20
	Total	310

In this paper we report Amharic OCR for printed documents that vary in fonts, sizes, styles and degradations. Amharic script has 33 core characters each of which occurs in seven orders (one basic and six non-

basic forms) (see Figure 1), which represent syllable combinations consisting of a consonant and following vowel [5]. Other symbols representing labialization, numerals, and punctuation marks are also available. These bring the total number of scripts to 310. Table 1 shows the number of characters in each group. Existence of such large number of characters in the writing system is a great challenge in the development of OCR for the language.

1.1. Features of the Amharic scripts

Amharic scripts have certain notable features. As pointed out by Bender [5], the shape of many Amharic characters shows similarities with few distinctions among them, for example, ደ and ጀ, ተ and ቸ. Many basic characters are also clearly related in structure, for instance, ነ and ከ, ረ and ፉ. There are also remarkable differences in shapes among the basic characters. Consider ሀ and በ (both are open in one side but in opposite direction), መ and ወ (both are formed from two loops but differ in the connection of the loops), ሠ and ጠ (both have three legs which end in different direction), etc.

An interesting peculiarity of the Amharic writing system is the way vowels are formed. Vowels are derived from consonants in two ways. Some vowels (such as the fourth and seventh orders) take a modified shape of the base character by shortening/ lengthening one of its main strokes. On the other hand, adding small appendages, such as strokes, loops to the right, left, top or bottom of each base character forms the remaining vowels (like second, third and fifth orders). As shown in Figure 1, the second, third, and fifth orders are formed (with few exceptions) according to patterns of great regularity, while the fourth, sixth and seventh orders are highly irregular. For instance, the second order is mostly constructed by adding a horizontal stroke at the middle of the right side of the base character; where as, the sixth order is formed by adding a stroke, loop or other forms in either side of the base character.

Amharic characters can differ in size. There are very short characters (such as ፉ, ሠ, መ) and there are very long characters (such as ኝ, ኹ, ኽ). There is also noticeable variance in width, for instance between ነ, ሚ, and ጪ. As compared to Latin scripts, the concepts of upper case and lower-case letters are absent in Amharic writing system.

2. Recognition of Amharic characters

Document images are initially preprocessed, i.e. binarized, noise removed and skew corrected before individual components are extracted. Gaussian filtering and projection profiles have been used for noise removal and skew correction, respectively. The page segmentation algorithm follows a top-down approach by identifying text blocks in the pages. Then, lines and words are segmented using projection profiles. Next, characters are detected using projections and they are split into their constituent components using connected component analysis (see [6] for details about preprocessing and segmentation). These are used as an input for feature extraction, training and testing.

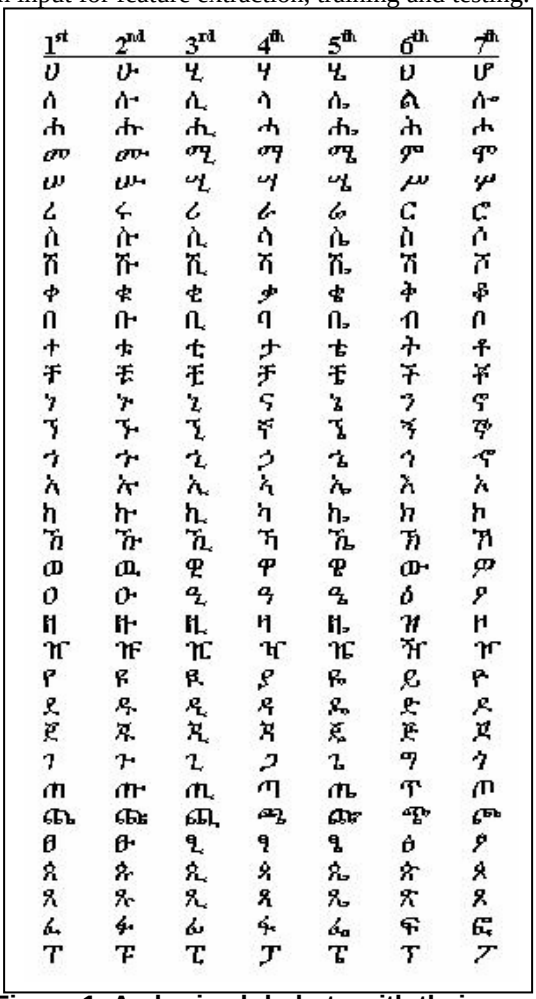

Figure 1. Amharic alphabets with their seven orders. The first order shows basic characters and others are non-basic vowels

2.1. Feature extraction

Feature extraction is the problem of identifying relevant information from raw data that characterize the component images distinctly [7]. There are methods that extract features like profiles, structural descriptors and transform domain representations. Alternates one could consider the entire image as the

feature. The former methods are highly language specific and become very complex to represent all the characters in the script. As a result, we extract features from the entire image by concatenating all the rows to form a single contiguous vector. With such a representation, memory requirement is very high for language like Amharic with large number of characters in the writing. Therefore we need to transform the features to obtain a lower dimensional representation.

$$
\begin{aligned}
&d_1 = \alpha_1 A^{-1}\Delta \\
&S_1^{-1} = 1/S_{11} \\
&for \quad n = 2 \quad to \quad K \\
&\quad d_n = \alpha_n A^{-1}\left\{\Delta - [d_1 \cdots d_{n-1}] S_{n-1}^{-1} [1/\alpha_1 0 \cdots 0]^t\right\} \\
&\quad \omega_n = (d_n^t \Delta)^2 / d_n^t A d_n \\
&\quad S_n^{-1} = \frac{1}{c_n}\left[\begin{array}{c|c} c_n S_{n-1}^{-1} + S_{n-1}^{-1} y_n y_n^t S_{n-1}^{-1} & -S_{n-1}^{-1} y_n \\ \hline -y_n^t S_{n-1}^{-1} & 1 \end{array}\right]
\end{aligned}
$$

Figure 2. Algorithm for computing L best discriminant feature vectors.

Principal Component Analysis (PCA) yields projection directions that maximize the total scatter across all classes. In choosing the projection which maximizes total scatter, PCA retains not only between-class scatter, that is useful for classification, but also within-class scatter, that is unnecessary for classification purposes. It is observed that much of the variation among document images is due to printing variations and degradations. Thus if PCA is presented with such images, the transformation matrix will contain principal components which retain these variations. Consequently, the points in the lower dimensional space will not be well separated and the classes may be smeared together.

Hence, in this work, we propose a two-stage feature extraction scheme. First we apply PCA for a gross dimensionality reduction. Then we use Linear Discriminant Analysis (LDA) to extract useful features for the classification. The objective of LDA is to find a projection, $y = Dx$ (where x is the input and D is the transformation), that maximizes the ratio of the between-class scatter and the within-class scatter [8]. For these, we apply the algorithm originally proposed by Foley and Sammon [9]. The algorithm extracts a set of optimal discriminant features for a two-class problem which suits the Support Vector Machine (SVM) classifier.

Consider the i^{th} image sample represented as an M dimensional (column) vector x_i, where M is the reduced dimension using PCA. For the sets of training samples, $x_1, \cdots, x_N$, we compute with-in class scatter (W) matrix and between-class difference (Δ) as:

$$W_i = \sum_{j=1}^{N_i} (x_{ij} - \mu_i)(x_{ij} - \mu_i)^t$$

$$\Delta = \mu_1 - \mu_2$$

where x_{ij} is the j^{th} sample in i^{th} class.

Sum of the with-in class scatter is also determined by, $A = cW_1 + (1-c)W_2$, where $0 \le c \le 1$, and the scatter space using $S_{ij} = d_i A^{-1} d_j$. The algorithm presented in Figure 2 is used for extracting an optimal set of discriminant vectors (d_n) that corresponds to the first L highest discriminant values such that ($\omega_1 \ge \omega_2 \ge \cdots \ge \omega_L \ge 0$) [8]. Here, K is the number of iterations for computing discriminant vectors and discriminant values, $c_n = s_{nn} - y_n^t S_{n-1}^{-1} y_n$, $y_n = [S_{in} \cdots S_{(n-1)(n)}]$, and α_n is chosen such that $d_n^t d_n = 1$. Interested readers may refer [8, 9] for details about LDA and Foley and Sammon algorithm.

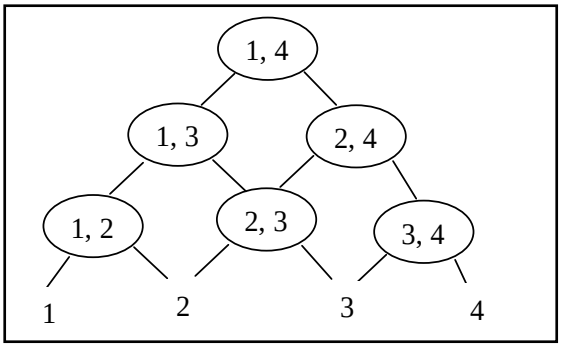

Figure 3. A rooted binary DDAG multi-class classifier for 4 class problem

2.2. Classification

SVM classifier has the ability to identify the decision boundary with maximal margin, which results in better generalization; a highly desirable property for a classifier to perform well on a novel data set [10, 11]. SVM is also suitable for OCR problems with high dimensional input data due to its effective training and testing algorithms and natural extension to the kernel methods.

SVMs are a pair-wise discriminating classifiers. Multi-class SVMs are usually implemented as combinations of two-class solution. We construct a directed acyclic graph (DAG), where each node in the graph

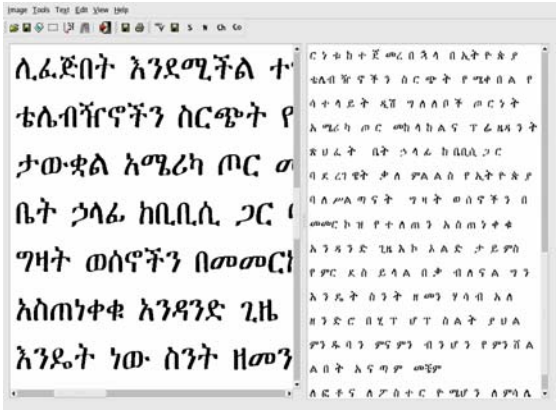

Figure 4. Screenshot of the user interface. The left side displays scanned image, while the right side shows its equivalent textual form after recognition.

corresponds to a two-class classifier for a pair of classes [10]. The multi-class classifier built using decision directed acyclic graph (DDAG) for a 4-class classification problem is shown in figure 3. It can be observed that the number of binary classifiers built for a N class classification problem is $N(N-1)/2$ .The input vector is presented at the root node of the DAG and moves through the DAG until it reaches the leaf node where the output (class label) is obtained.

SVM is trained pair-wise with the discriminant features extracted from the given two-class characters using the two-stage scheme. The classifier then creates two-class models that are used for classification.

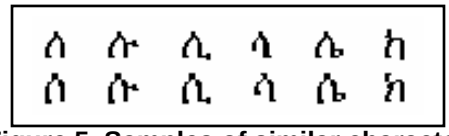

Figure 5. Samples of similar characters

3. Results and discussions

We have a system for the recognition of a given document images into equivalent textual format. A screenshot of an interface is shown in Figure 4. The system accepts either already scanned documents or scans document pages from a flat-bed scanner. Scanned pages are preprocessed and segmented into character components. Then discriminant features are extracted for classification using a two stage dimensionality reduction scheme based on 99% PCA and 15% LDA. Both methods perform well in feature dimensionality reductions. However, as presented in Figure 5, there are very similar characters in Amharic script that are even difficult for humans to identify them easily. The use of a two-stage feature extraction scheme solves the similarity problem encountered during the application of PCA alone. Based on experiment conducted on small datasets, the two-stage feature extraction scheme improves the accuracy rate by at least 8.06 %. This is because the new scheme extracts optimal features that discriminate between a pair of characters, which are used for training and classification. Finally characters are recognized and a minimal post-processor is used to correct miss-classified once. Results of the OCR are converted texts that can be easily manipulated.

Table 2. Recognition rate on the various fonts

Fonts	Test data size	Accuracy (%)
Power Geez	7850	99.08
Visual Geez	7850	96.24
Agafari	7850	95.53
Alpas	7850	95.16

We conducted extensive experiments to evaluate the performance of the recognition process on the various datasets of Amharic scripts. The experiments are organized in a systematic manner considering the various situations encountered in real-life printed documents. Our test datasets are, in general composed of printing variations (such as fonts, styles and sizes) and degraded Amharic documents (such as newspapers, magazines and books). We report performance of the Amharic OCR in all these datasets and the result is promising to extend it for the recognition of other African scripts.

Table 3. Performance report on different point sizes

Point Size	Test data size	Accuracy (%)
10	7680	98.64
12	7680	99.08
14	7680	98.06
16	7680	98.21

Performance evaluation is done in a step-wise manner as follows. In the first experiment, we considered the font-specific performance. There are a number of distinct fonts used in printed documents that are designed in an unstructured manner. We tested on four of these fonts: PowerGeez, VisualGeez, Agafari, Alpas. These are the most popular fonts frequently used for printing. We consider an average of more than 7500 samples for the experiment. Results are presented in Table 2. The recognition rate is high for all fonts. Misclassifications occurred because of two reasons. The first problem is related to the similarity in vowel formation between third and fifth orders of the same base characters. The degree of complexity of characters shape formation by individual font is also another factor for the reduction in the accuracy rate.

In the second experiment we dealt with samples printed at various point sizes of 10, 12, 14 and 16. Results are shown in Table 3. High recognition rates were obtained for each case. The results are almost uniform through out all font sizes as we scale them to a standard 20×20 size. The system is invariant to point size variations.

Next we experimented with samples printed at various font styles such as normal, bold and italics. Results are reported in Table 4. The system registers good performance for most of the font styles. Since we trained the OCR deliberately with normal font style, the recognition rate for italics is reduced. For better result we need to train the classifier with added samples of italics style.

Table 4. Accuracy rate on different font styles

Style	Test data size	Accuracy (%)
Normal	7680	99.08
Bold	7680	98.21
Italic	7680	89.67

In general, high performance has been registered on the above datasets; on the average 96.95% accuracy is obtained. This is because paper and printing qualities were reasonably good; rather the challenge here is printing variations. In real-life situations, however we also encounter the problem of degradations. We tested documents, such as books, magazines and newspapers. The documents are of poor quality as shown in Figure 6. We applied Guassian filtering to reduce the effect of degradations. Recognition results are shown in Table 5. Misclassification of characters is basically occurs because of artifacts such as large ink-blobs joining disjoint characters or components, and cuts of characters at arbitrary direction due to paper quality or foreign material.

Table 5. Performance of the system on degraded real-life documents

Document	Test data size	Accuracy (%)
Books	6240	91.45
Newspaper	5430	88.23
Magazine	5560	90.37

4. Conclusions

This paper presents the challenges towards the recognition of indigenous African scripts. We also highlight features of Amharic characters and problems related to the scripts that have bearings on Amharic OCR development, especially availability of large number of characters and similarity among symbols. We employed a two-stage feature extraction procedure using PCA and LDA for selecting optimal discriminant vectors for classification. The system is now being extensively tested on both printing variations and degraded documents. The use of SVM classifier is advantageous because of their generalization capability. We are currently working on other classifiers with smaller footprints. Future work will extend the OCR technology towards the recognition of other African scripts.

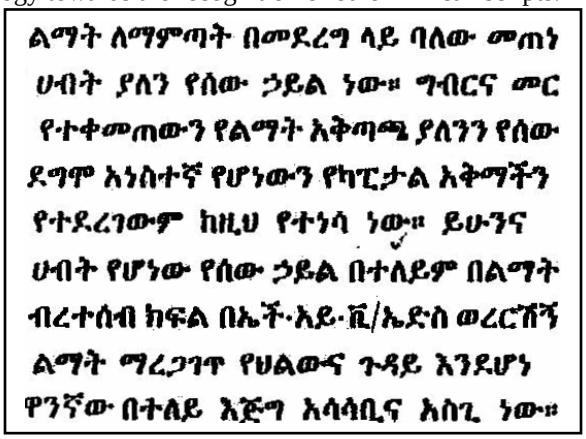

Figure 6. Sample from real-life documents

5. References

[1] G. Nagy. “At the Frontiers of OCR,” Proceedings of the IEEE, vol. 80, no. 7, 1992, pp. 1093-1099.

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

[3] Worku Alemu and Siegfried Fuchs, “Handwritten Amharic Bank Check Recognition Using Hidden Markov Random Field”, Document Image Analysis and Retrieval Workshop (DIAR'03), 2003, p. 28.

[4] S. Mafundikwa, “African Alphabets”, Zimbabwe, November 2000. http://www.ziva.org..zw/afrikan.htm

[5] Bender, M.L., “Language in Ethiopia”, Oxford University Press, London, 1976.

[6] C. V. Jawahar, M. N. S. S. K. Pavan Kumar and S. S. Ravi Kiran, “A Bilingual OCR for Hindi-Telugu Documents and its Applications”, International Conference on Document Analysis and Recognition (ICDAR), 2003, pp. 408-412.

[7] O. D. Trier, A. K. Jain and T. Taxt, “Feature Extraction Methods for Character Recognition: A Survey”, Pattern Recognition, vol. 29, no. 4, 1996, pp. 641-662.

[8] Duda, R.O., P.E. Hart, and D.G. Stork, Pattern Classification. John Wiley & Sons, Inc., New York, 2001.

[9] D.H. Foley and J.W. Sammon. “An Optimal Set of Discriminant Vectors”. IEEE Transactions on Computing, vol. 24, 1975, pp. 271-278.

[10] J.C. Platt, N. Cristianini and J. Shawe-Taylor, “Large Margin DAGs for Multi-class Classification”, Advances in Neural Information Processing Systems 12, 2000, pp. 547-553.

[11] C. JC. Burges. “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, 1998.