Optical Character Recognition of Amharic Documents

Million Meshesha C. V. Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, India
million@research.iiit.ac.in jawahar@iiit.ac.in

***Abstract*— In Africa around 2,500 languages are spoken. Some of these languages have their own indigenous scripts. Accordingly, there is a bulk of printed documents available in libraries, information centers, museums and offices. Digitization of these documents enables to harness already available language technologies to local information needs and developments. This paper presents an Optical Character Recognition (OCR) system for converting digitized documents in local languages. An extensive literature survey reveals that this is the first attempt that reports the challenges towards the recognition of indigenous African scripts and a possible solution for Amharic script. Research in the recognition of African indigenous scripts faces major challenges due to (i) the use of large number characters in the writing and (ii) existence of large set of visually similar characters. In this paper, we employ a novel feature extraction scheme using principal component and linear discriminant analysis, followed by a decision directed acyclic graph based support vector machine classifier. Recognition results are presented on real-life degraded documents such as books, magazines and newspapers to demonstrate the performance of the recognizer.**

Index Terms— Optical Character Recognition; African Scripts; Feature Extraction; Classification; Amharic Documents.

I. INTRODUCTION

Nowadays, it is becoming increasingly important to have information available in digital format for increased efficiency in data storage and retrieval, and optical character recognition (OCR) is being known as one of valuable input devices in this respect [1], [2]. This is also supplemented by the evolution of large digital libraries and the advancement of information technology.

Digital libraries such as Universal Digital Library, Digital Library of India [3], etc. are established for digitizing paper documents and make them available to users via the Internet. The advancement of information technology supports faster, more powerful processors and high speed output devices (printers and others) to generate more information at a faster rate. These days the availability of relatively inexpensive document scanners and optical character recognition software has made OCR an attractively priced data entry methodology [1].

Optical character recognition technology was invented in the early 1800s, when it was patented as reading aids for the blind. In 1870, C. R. Carey patented an image transmission system using photocells, and in 1890 P.G. Nipkow invented sequential scanning OCR [4]. However, the practical OCR technology used for reading characters was introduced in the early 1950s as a replacement for keypunching system. A year later, D.H. Shephard developed the first commercial OCR for typewritten data. The 1980's saw the emergence of OCR systems intended for use with personal computers [4]. Nowadays,

it is common to find PC-based OCR systems that are commercially available. However, most of these systems are developed to work with Latin-based scripts [5].

Optical character recognition converts scanned images of printed, typewritten or handwritten documents into computer readable format (such as ASCII, Unicode, etc.) so as to enable electronic data processing. The potential of OCR for data entry application is obvious: it offers a faster, more automated, and presumably less expensive alternative to the manual data entry devices, thereby improving the accuracy and speed in transcribing data into the computer system. Consequently, it increases efficiency and effectiveness (by reducing cost, time and labor) in information storage and retrieval.

Major applications of OCR include: (i) Library and office automation, (ii) Form and bank check processing, (iii) Document reader systems for the visually impaired, (iv) Postal automation, and (v) Database and corpus development for language modeling, text-mining and information retrieval [2], [6].

While the use and application of OCR systems is well developed for most languages in the world that use both Latin and non-Latin scripts [7], an extensive literature survey reveals that few conference papers are available on the indigenous scripts of African languages. Lately some research reports have been published on Amharic OCR. Amharic character recognition is discussed in [8] with more emphasis to designing suitable feature extraction scheme for script representation. Recognition using direction field tensor as a tool for Amharic character segmentation is also reported [9]. Worku and Fuchs [10] present handwritten Amharic bank check recognition. On the contrary, there are no similar works being found for other indigenous African scripts.

Therefore, there is a need to exert much effort to come up with better and workable OCR technologies for African scripts in order to satisfy the need for digitized information processing in local languages.

II. AMHARIC SCRIPTS

In Africa more than 2,500 languages, including regional dialects are spoken. 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. As a result, most African languages with a writing system use a modification of the Latin and Arabic scripts. There are also many languages with their own indigenous scripts and writing systems. Some of these scripts include Amharic script (Ethiopia), Vai script (West Africa), Hieroglyphic script (Egypt), Bassa script (Liberia), Mende script (Sierra Leone), Nsibidi/Nsibiri script (Nigeria and Cameroon) and Meroitic script (Sudan) [11].

Amharic, which belongs to the Semitic language, became a dominant language in Ethiopia back in history. It is the official and working language of Ethiopia and the most commonly learnt language next to English throughout the country. Accordingly, there is a bulk of information available in printed form that needs to be converted into electronic form for easy searching and retrieval as per users' need. Suffice is to mention the huge amount of documents piled high in information centers, libraries, museums and government and private

offices in the form of correspondence letters, magazines, newspapers, pamphlets, books, etc. Converting these documents into electronic format is a must in order to (i) preserve historical documents, (ii) save storage space, (iii) enhance retrieval of relevant information via the Internet and other applications. This enables to harness existing information technologies to local information needs and developments.

	Ge'ez ä	*Ka'eb* u	*Salis* ï	*Rab'e* a	*Hamis* é	*Sadis* i	*Sab'e* o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ḥ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
q	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
t	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
h	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
n	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
a	አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
k	ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
w	ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
ạ	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
z	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
y	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
d	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
g	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ṭ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
p	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
ts	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ts	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
f	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
p	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

Fig. 1: Amharic alphabets (FIDEL) with their seven orders row-wise. The 2nd column shows list of basic characters and others are vowels each of which derived from the basic symbol.

Those African languages using a modified version of Latin and Arabic scripts can easily be integrated to the existing Latin and Arabic OCR technologies. It is worth to mention here some of the related works reported at home and in the Diaspora to preserve African languages digitally (predominantly in languages that use Latin scripts). Corpora projects in Swahili (East Africa), open source software for languages like Zulu, Sepedi and Afrikaans (South African), indigenous web browser in Luganda (Uganda), improvisation of keyboard characters for some African languages, the presence of African languages on the Internet, etc. [12].

Therefore, we need to give more emphasis to those indigenous African scripts. This is the motivation behind the present report. To the best of our knowledge, this is the first work that reports a robust Amharic character recognizer for conversion of printed document images of varying fonts, sizes, styles and quality.

Amharic is written in the unique and ancient Ethiopic script (inherited from Geez, a Semitic language), now effectively a syllabary requiring over 300 glyph shapes for representation. As shown in Fig. 1, Amharic script has 33 core characters each of which occurs in seven orders: one basic form and six non-basic forms consisting of a consonant and following vowel. Vowels are derived from basic characters with some modification (like by attaching strokes at the middle or end of the base character, by elongating one of the leg of the base character, etc.). Other symbols are also available that represent labialization, numerals, and punctuation marks. These bring the total number of characters in the script to 310 as shown in Table 1.

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

Table 1. Summary of the number of characters in the Amharic script (FIDEL).

2.1 Challenges in Building an OCR for African Scripts

Character recognition from document images that are printed in African scripts is a challenging task, including Amharic documents. To develop a successful character recognition system for these scripts, we need to address some of the following issues.

A. Degradation of documents

Document images from printed documents, such as books, magazines, newspapers, etc. are extremely poor in quality. Popular artifacts in printed document images include:

(a) Excessive dusty noise,
(b) Large ink-blobs joining disjoint characters or components,
(c) Vertical cuts due to folding of the paper,
(d) Cuts at arbitrary direction due to paper quality or foreign material,
(e) Degradation of printed text due to the poor quality of paper and ink,
(f) Floating ink from facing pages etc.

This is the main issue where most character recognition research even for Latin and non-Latin scripts also fails. We need to carefully design an appropriate representational scheme and classification method so as to accommodate the effect of degradation.

B. Printing variations

Printed documents vary in fonts, sizes and styles. Building character recognition system is challenging in this situation. For example, some of the commonly used fonts in Amharic printed documents include `PowerGeez', `VisualGeez', `Alphas', `Agafari', etc. Each of these fonts offers several stylistic variants, such as normal, bold, italic, etc. They are also written in different point sizes, including 10, 12, 14, etc. These fonts, styles and sizes produce texts that vary in their appearances (i.e. in size, shape, quality, etc.) within printed documents. We need to standardize the variation in size by applying normalization techniques and extract suitable features so that the representation is invariant to printing variations.

C. Large number of characters in the script

The total number of characters in Amharic script is more than three hundred. Existence of such a large number of Amharic characters in the writing system is a great challenge in the development of Amharic character recognizer. Memory and computational requirements are very intensive. We need to design a mechanism to compress the dimension of character representation so as to come up with computationally efficient recognizers.

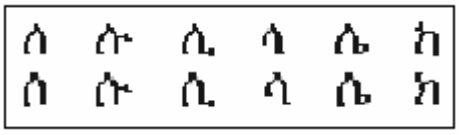

Fig. 2: Samples of visually similar characters in Amharic writing system.

D. Visual similarity of most characters in the script

There are a number of very similar characters in Amharic script that are even difficult for humans to identify them easily (examples are presented in Fig. 2). Robust discriminant features needs to be extracted for classification of each of the character into their proper category or class.

E. Language related issues

African indigenous languages pose many additional challenges. Some of these are: (i) lack of standard representation for the fonts and encoding, (ii) lack of support from operating systems, browsers and keyboard, and (iii) lack of language processing routines.

These issues add to the complexity of the design and implementation of an optical character recognition system.

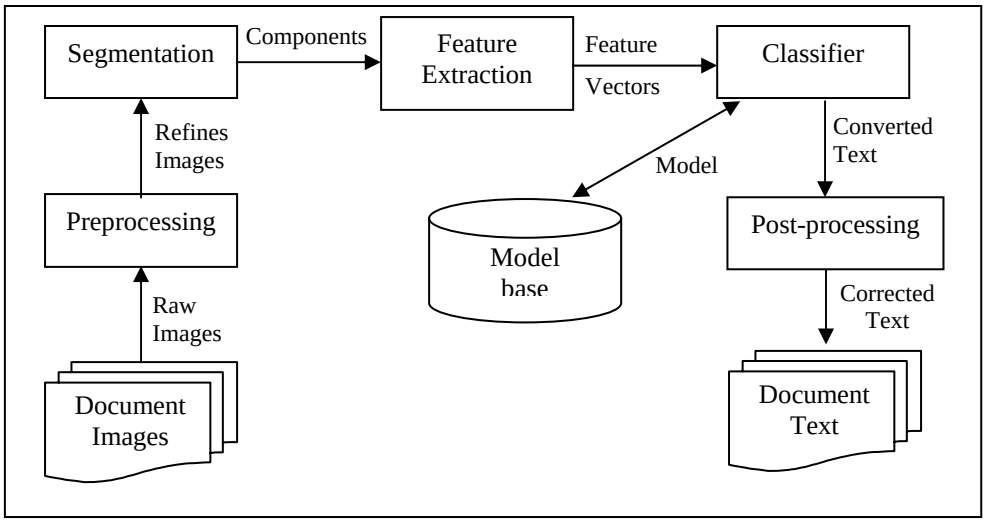

Fig. 3: Overview of the Amharic OCR design.

III. RECOGNITION OF AMHARIC SCRIPT

We built an Amharic OCR on top of an earlier work for Indian languages [13] and also reported a preliminary result of our work in [14]. Fig. 3 shows the general framework of Amharic OCR design.

The system accepts as input scanned document images. The resolution, measured in number of dots per inch (dpi), produced by the scanner determines how closely the reproduced image corresponds to the original. The higher the resolution, the more information is recorded and, therefore, the greater the file size and vice versa. For most uses, 200 to 300 dpi is an adequate resolution for producing sufficient image quality while keeping the size of the image files manageable. Most systems however recommend scanning documents at 300 dpi resolution for optimum OCR accuracy. We also use 300 dpi to digitize documents for our experiment.

Once images are captured, they are preprocessed before individual components

are extracted. Preprocessing is necessary for efficient recovery of the text information from scanned image. The preprocessing algorithms employed on the scanned image depend on paper quality, resolution of the scanned image, the amount of skew in the image, the format and layout of the images and text and so on [2], [6]. In this work we apply binarization, noise removal, skew correction and normalization techniques.

Scanned documents often contain noise that mainly arises due to printer, scanner, paper quality and age of the document. We use binarization to convert gray scale image (with 256 possible different shades of gray from black to white pixels) to binary colors (with just black and white pixels) to ease image processing. In due course binarization removes some of the noises in images. However, because of the level of degradations in digitized document images of magazines, books and newspaper, there is a need to apply filters so as to reduce the effect of degradation during the recognition process. We use Gaussian filtering that smoothes an image by calculating weighted averages in a filter box [2].

During the scanning operation, a certain amount of image skew is unavoidable. So, skew detection and correction is an important pre-processing step. In our work projection profile have been used for skew correction in the range of ± 20%. Further detail is available in [13].

Once pages are preprocessed, then they are segmented into individual components. Segmentation is an operation that seeks to decompose an image into sub-images of individual symbols [15]. The page segmentation algorithm follows a top-down approach by identifying text blocks in the document pages. This is followed by line, word and character segmentation.

In our implementation, line, word and characters in the text are segmented in the following manner. Lines are detected from text blocks using horizontal projection. A text line can be found between two consecutive boundary lines where the height of projection profile is least. Identified text lines are then segmented into its constituent words.

We employ vertical projection for word segmentation. This approach scans each line, the valleys of which show the boundaries of each word in the line. Next, characters are identified using vertical projection. To identify the boundaries of characters each word is scanned in the vertical direction. During scanning if there is no black pixel encountered till the base line is reached, then this scan marks a demarcation line for characters.

Once character boundaries have been detected, it is often useful to extract character components. We use connected component analysis to identify the connected segments of a character. The final result is a set of character components with their bounding box that are scaled to a standard size of 20×20 . This dataset is used as an input for feature extraction, training and testing the recognition engine.

IV. Feature Extraction

Feature extraction is the problem of identifying relevant information from raw data that characterize the component images distinctly. There are many popular methods to extract features. Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition systems. A survey of feature extraction schemes for character recognition is available in [16]. Different feature extraction methods are designed for different representations of the characters. Some of these features consider profiles,

structural descriptors and transform domain representations [2], [17]. 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. The later scheme provides excellent results for printed character recognition.

As a result, we extract features from the entire image by concatenating all the rows to form a single contiguous vector. This feature vector consists of zeros (0s) and ones (1s) representing background and foreground pixels in the image, respectively. With such a representation, memory and computational requirements are very intensive for languages like Amharic that have large number of characters in the writing.

Therefore we need to transform the features to obtain a lower dimensional representation. There are various methods employed in pattern recognition literatures for reducing the dimension of feature vectors [18]. In the present work we use Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

4.1 Principal Component Analysis

Principal component analysis (PCA) can help identify new features (in a lower dimensional subspace) that are most useful for representation [18]. This should be done without losing valuable information. Principal components can give superior performance for font-independent OCRs, easy adaptation across languages, and scope for extension to handwritten documents.

Consider the i^{th} image sample represented as an M dimensional (column) vector x_i, where M depends on the image size. From the large sets of training datasets, $x_1, \cdots, x_N$ we compute the covariance matrix as:

$$\sum = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T \quad (1)$$

Then we need to identify minimal dimension, say K such that:

$$\sum_{i-1}^{K} \lambda_i / Trace(\sum) \leq \alpha \quad (2)$$

where λ_i is the i^{th} largest eigenvalue of the covariance matrix $\sum$ and α is a limiting value in percent.

Eigenvectors corresponding to the largest K eigenvalues are the direction of greatest variance. The k^{th} eigenvector is the direction of greatest variation perpendicular to the first through $(k-1)^{st}$ eigenvectors. The eigenvectors are arranged as rows in matrix A and this transformation matrix is used to compute the new feature vector by projecting as $Y_i = Ax_i$. With this we get the best one dimensional representation of the component images with reduced feature size.

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 unwanted information for classification purposes. Much of the variation seen among document images is due to printing variations and degradations. If PCA is applied on such images, the transformation matrix will contain principal components that retain these variations in the projected

feature space. Consequently, the points in the projected space will not be well separated and the classes may be smeared together.

Thus, while the PCA projections are optimal for representation in a low dimensional space, they may not be optimal from a discrimination standpoint for classification. Hence, in our work, we further apply linear discriminant analysis to extract useful features from the reduced dimensional space using PCA.

4.2. Linear Discriminant Analysis

Linear discriminant analysis (LDA) selects features based entirely upon their discriminatory potential. Let the projection be $y = Dx$, where x is the input image and D is the transformation matrix. The objective of LDA is to find a projection that maximizes the ratio of the between-class scatter and the within-class scatter [16].

We apply the algorithm proposed by Foley and Sammon [19] for linear discriminant analysis. The algorithm extracts a set of optimal discriminant features for a two-class problem which suits the Support Vector Machine (SVM) classifier we used for classification task.

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

$$W_i = \sum_{i=1}^{N_i} (x_{ij} - \mu_i)(x_{ij} - \mu_i)^t \quad (3)$$

$$\Delta = \mu_1 - \mu_2 \quad (4)$$

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

Sum of the with-in class scatter is also determined by,

$$A = cW_1 + (1-c)W_2 \quad (5)$$

where $0 \le c \le 1$, and the scatter space using:

$$S_{ij} = d_i A^{-1} d_j \quad (6)$$

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

Fig. 4: Algorithm for computing L best discriminant feature vectors. The vertical and horizontal lines are drawn to partition the scatter matrix S_n^{-1} such that the result obtained is a 2×2 matrix.

The algorithm presented in Fig. 4 is called recursively for extracting an optimal set of discriminant vectors (d_n) that corresponds to the first L highest discriminant values such that ($\omega_1 \ge \omega_2 \ge \cdots \ge \omega_L \ge 0$) [19]. Here, K is the number of iterations for computing discriminant vectors as well as discriminant values, and, α_n is chosen such that:

$$d_n^t d_n = 1 \quad (7)$$

y_n and c_n are determined in the following manner.

$$y_n = [S_{in} \cdots S_{(n-1)(n)}] \quad (8)$$

$$c_n = s_{nn} - y_n^t S_{n-1}^{-1} y_n$$

Since the principal purpose of classification is discrimination, the discriminant vector subspace offers considerable potential as feature extraction transformation. Hence, the use of linear discriminant features for each pair of classes in each component classification is an attractive solution to design better classifiers.

However, the problem with this approach is that the classification becomes slower, because LDA extracts features for each pair-wise class. This requires stack space for storing $N(N-1)/2$ transformation matrices. For instance, for dataset with 200 classes LDA generates around 20 thousand transformation matrices for each pair and is very expensive for large class problems.

Hence, we propose a two-stage feature extraction scheme; PCA followed by LDA for optimal discriminant feature extraction. We first reduce the feature dimension using PCA and then run LDA on the reduced lower dimensional space to extract the most discriminant feature vector. This reduces to a great extent the storage and computational complexity, while enhancing the performance of SVM-based decision directed acyclic graph (DDAG) classifier.

V. CLASSIFICATION

Training and testing are the two basic phases of any pattern classification problem. During training phase, the classifier learns the association between samples and their labels from labeled samples. The testing phase involves analysis of errors in the classification of unlabelled samples in order to evaluate classifier's performance. In general it is desirable to have a classifier with minimal test error.

We use Support Vector Machine (SVM) for classification task [20]. SVMs are pair-wise discriminating classifiers with the ability to identify the decision boundary with maximal margin. Maximal margin results in better generalization [21] which is a highly desirable property for a classifier to perform well on a novel dataset. Support vector machines are less complex and perform better (lower actual error) with limited training data.

Identification of the optimal hyper-plane for separation involves maximization of an appropriate objective function. The result of the training phase is the identification of a set of labeled support vectors x_i and a set of coefficients α_i. Support vectors are the samples near the decision boundary. They have class labels y_i ranging ±1. The decision is made from:

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x)) \quad (9)$$

where K is the kernel function, which is defined by:

$$K(x, y) = \phi(x)\phi(y) \quad (10)$$

where $\phi : R^d \to H$ maps the data points in lower dimensions to a higher dimensional space H.

Binary classifiers like SVM are basically designed for two class classification problems. However, because of the existence of a number of characters in any script, optical character recognition problem is inherently multiclass in nature. The field of binary classification is mature, and provides a variety of approaches to solve the problem of multiclass classification [22]. Most of the existing multiclass algorithms address the problem by first dividing it into

smaller sets of a pair of classes and then combine the results of binary classifiers using a suitable voting methods such as majority or weighted majority approaches [22], [23].

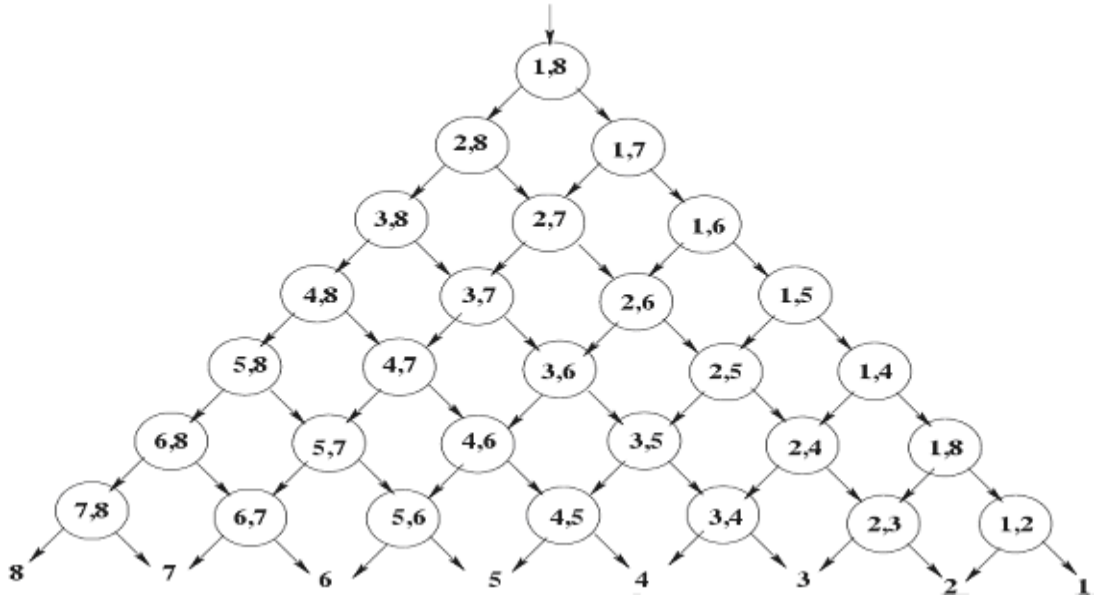

Fig. 5: A rooted binary decision directed acyclic graph (DDAG) multiclass classifier for eight-class classification problem.

5.1 Multiclass SVMs

Multiclass SVMs are usually implemented as combinations of two-class solution using majority voting methods. It has been shown that the integration of pairwise classifiers using decision directed acyclic graph (DDAG) results in better performance as compared to other popular techniques such as decision tree, Bayesian, etc. [20].

We construct a decision directed acyclic graph (DDAG), where each node in the graph corresponds to a two-class classifier for a pair of classes. The multiclass classifier built using decision directed acyclic graph (DDAG) for 8-class classification problem is shown in Fig. 5. 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 DDAG and moves through the DDAG until it reaches the leaf node where the output (class label) is obtained. At each node a decision is made concerning to which class the input belongs.

For example, at the root node of Fig. 5, a decision is made whether the input belongs to class 1 or class 8. If it does not belong to class 1, it moves to the left child; otherwise, if it does not belong to class 8, it moves to the right child. This process is repeated at each of the remaining nodes till it reaches the leaf node where the decision is reached to classify the input as one of the eight classes.

VI. Results and Discussions

We have a recognition system that converts a given document images into equivalent textual format. The system accepts either already scanned document

images or scan a given Amharic text document at a resolution of 300 dpi on a flat-bed scanner, HP7670 Scanjet scanner. The scanned document is binarized, noise removed, skew corrected and scaled before individual components are extracted. Preprocessed pages are then segmented into character components for feature extraction and classification. Following Amharic letters shape formation; we can first decompose lines in a text page into words and then words into appropriate character components for recognition and then recompose the recognized components to determine the order of characters, words and lines in a text page.

Once character components are identified, optimal discriminant features (in a lower dimensionality space) are extracted for classification. We use a two stage dimensionality reduction scheme based on 99 percent principal component analysis which is followed by 15 percent linear discriminant analysis.

The original dimension of feature vector of each character image is 400 (20×20) after normalization. Principal component analysis reduces the dimensionality from 400 to 295 and linear discriminant analysis further to 50. We are dealing with such reduced optimal discriminant feature vectors. Both methods perform well in feature dimensionality reductions. The use of a two-stage feature extraction scheme further solves the similarity problem encountered during the application of PCA alone. This is because of the fact that the new scheme extracts optimal features that discriminate between a pair of characters employed for classification using support vector machine.

We conduct 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 datasets are of two types: One set of datasets considers printing variations (such as fonts, styles and sizes). The other is degraded documents such as newspapers, magazines and books. We report the performance of the recognizer in all these datasets and the result obtained is promising to extend it for other indigenous African scripts.

In the first experiment, we consider the printing variation. We test on the most popular fonts such as PowerGeez, VisualGeez, Agafari and Alpas that are used for typing and printing purposes, four point sizes (10, 12, 14 and 16) and font styles (such as normal, bold and italics). Performance results are shown in Table 2.

Good recognition rates are obtained for multiple point size. The results are almost uniform through out all font sizes as we scale them to a standard size of 20×20 before feature extraction and dimensionality reduction. This makes the system invariant to point size variations.

Result	Fonts				Point Size				Style		
	Power Geez	Visual Geez	Agafari	Alpas	10	12	14	16	Normal	Bold	Italic
Datasets	7850	7850	7850	7850	7680	7680	7680	7680	7680	7680	7680
Accuracy	99.08	96.24	95.53	95.16	98.64	99.08	98.06	98.21	99.08	98.21	89.67

Table 2: Performance result of the Amharic OCR on pages that varies in fonts, sizes and styles.

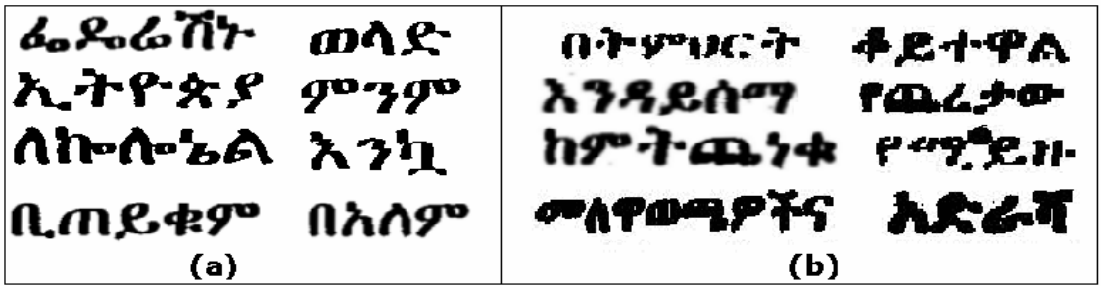

Fig. 6: Performance of the recognizer on Amharic word images. (a) A set of correctly recognized sample words, (b) A set of mis-recognized sample words.

The recognition rate for fonts is more than 95 percent. It works well for 'Power Geez' font as the system is trained on datasets prepared using this font. There is around three percent accuracy difference as compared to other fonts. Recognition accuracy is comparable for other fonts. Mis-classifications mainly 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.

The system works well for normal and bold styles with more than 98 percent accuracy. Since we trained the OCR deliberately with normal font style, the recognition rate for italics is reduced. This is because the shape of characters written using italics style is very complex unless skew is corrected. It is obvious that, as can be done else where, better result can be obtained by retraining the classifier with added training samples of italics style.

In general, better performance has been registered on the above synthetic datasets; on the average 96.95% accuracy is obtained. This is because paper and printing qualities were reasonably good. Fig. 6 (a) shows sample word images of this type that are correctly recognized. The challenge with these datasets is printing variations that happened due to fonts, sizes and styles used for document production

In real-life situations, however we also encounter the problem of degradations in document images. To see the effect of this artifacts, we evaluate the performance of the OCR on Amharic documents digitized from books, magazines and newspapers. The documents are of poor quality (both in printing and paper quality). Fig. 6 (b) depicts some of these degraded words. We apply Gaussian filtering algorithm to reduce the effect of degradations in the images during the recognition process. Recognition results are shown in Table 3.

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

Table 3: Performance of the system on degraded real-life documents.

On the average around 90 percent recognition is obtained. Characters are misrecognized because of artifacts such as cuts, merges, ink blobs, etc. that are commonly observed in printed document images scanned from books, magazines and newspapers. Degradations due to cuts

break character components into two or more, and ink-blobs join disjoint characters as one connected components. Due to these changes, for instance a cut at the bottom and top of the character '0' results in visually similar characters with '  ' and '  ', respectively. Further, merge of characters, say '  ' and '  ' produces character '  ', etc. The same thing also happens with blobs. This requires degradation modeling in order to simulate the effect of noise and accordingly design advanced noise removal algorithms that are effective in detecting degradations in document images before applying segmentation, feature extraction and classification techniques.

As the complexity and diversity of document images increases, the performance of the OCR is greatly affected since it is sensitive to degradation and unseen fonts. This is also true for those OCRs designed for Latin-based scripts. The problem is with the principle of existing OCR design, which guides to train the classifier offline and use it online for recognition. The approach therefore does not allow the engine to learn from its errors that happened during the recognition process. As a solution, most commercial systems suggest retraining the OCR engine again with the unseen samples for performance improvement. This is however inefficient and time consuming. Hence, there is a need to investigate alternate approaches so as to come up with an intelligent OCR that can learn from its mistake and improve its performance overtime. Currently we are investigating the feasibility of designing a data-driven OCR.

VII. Conclusion and Future Work

This paper presents the challenges towards the recognition of indigenous African scripts. We employed a two-stage feature extraction scheme using principal component analysis and linear discriminant analysis for selecting optimal discriminant feature 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 efficiency and generalization capability.

Developing a robust Amharic OCR system is one of the future work, besides extending the OCR technology towards the recognition of other indigenous African scripts. As an immediate solution we are also working in the area of indexing and retrieval from degraded document images using only image properties (without explicit recognition), in parallel with the development of large corpus of archived document images written in the various African languages.

References

[1] Nagy, G., "Twenty years of document image analysis in PAMI", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, January 2000, pp. 38-62.

[2] Mori, S., Nishida, H. and Yamada, H., Optical Character Recognition, New York: John Wiley & Sons, Inc., 1999.

[3] Ambat V., Balakrishnan N., Reddy R., Pratha L. and Jawahar C.V., "The Digital Library of India project: process, policies and architecture", International Conference on Digital Libraries (ICDL'06), 2006, pp. 5-8.

[4] Mori, S., Suen, C. Y. and Yamamoto, K., "Historical review of OCR research and development", *Proceedings of the IEEE*, vol. 80, no. 7, July 1992, pp. 1029-1058.

[5] Chaudhuri, B. B. and Pal, U., "A complete printed Bangla OCR system," *Pattern Recognition*, vol. 31, no. 5, May 1998, pp. 531–549.

[6] Pavlidis, T. and Mori, S., "Optical character recognition", *Proceedings of the IEEE*, vol. 80, no. 7, July 1992, pp. 1026-1028.

[7] Suen, C. Y., Mori, S., Kim, S. H. and Leung, C. H, "Analysis and recognition of Asian scripts - The state of the art", in International Conference

on Document Analysis and Recognition, 2005, pp. 866-878.

[8] Cowell, J. and Hussain, F., "Amharic Character Recognition using a Fast Signature Based Algorithm," in Proceedings of the Seventh International Conference on Information Visualization, 2003, pp. 384-389.

[9] Yaregal, A. and Bigun, J., "Ethiopic Character Recognition Using Direction Field Tensor," in 18th International Conference on Pattern Recognition (ICPR), 2006, pp. 284-287.

[10] Worku, A., and Fuchs, S., "Handwritten Amharic bank check recognition using hidden markov random field", in Document Image Analysis and Retrieval Workshop, 2003, p. 28.

[11] Mafundikwa, S., "African alphabets," www.ziva.org.zw, Link to a page, November 2000 [Online]. Available: http://www.ziva.org.zw/afrikan.htm, [Accessed April 23, 2007].

[12] Osborn, D., "African languages and information and communication technologies: literacy, access, and the future". in Proceedings of the 35th Annual Conference on African Linguistics, 2006, pp. 86 - 93.

[13] Jawahar, C. V., Pavan Kumar, M. N. S. S. K. and Ravi Kiran, S. S., "A bilingual OCR for Hindi-Telugu documents and its applications", in International Conference on Document Analysis and Recognition (ICDAR), 2003, pp. 408-412.

[14] Million Meshesha and Jawahar, C. V., "Recognition of printed Amharic documents," in Proceedings of Eighth International Conference on Document Analysis and Recognition (ICDAR), 2005, pp 784-788.

[15] Casey, R. G. and Lecoline, E., "A survey of methods and strategies in character segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, July 1996, pp. 690-706.

[16] Trier, O., Jain, A. and Taxt, T., "Feature extraction methods for character recognition - A survey", *Pattern Recognition*, vol 29, no. 4, 1996, pp. 641-662.

[17] Gonzalez, R. C. and Woods, R. E., Digital Image Processing, 2nd ed., India: Pearson Education, 2002.

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

[19] Foley, D.H. and Sammon, J. W., "An optimal set of discriminant vectors", *IEEE Transactions on Computing*, vol. 24, March 1975, pp. 271-278.

[20] Platt, J. C., Cristianini, N. and Shawe-Taylor, J., "Large margin DAGs for multi-class classification", in Advances in Neural Information Processing Systems 12, 2000, pp. 547-553.

[21] Burges, C. JC., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, June 1998, pp. 121-167.

[22] Allwein, E. L., Schapire, R. E. and Singer, Y., "Reducing multiclass to binary: A unifying approach for margin classification," In Proceeding of 17th International Conf. on Machine Learning, 2000, pp. 9-16.

[23] Dietterich, T. G., "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, 2000, pp. 1-15.

Million Meshesha received his M. Sc. from Addis Ababa University (AAU), Ethiopia, in 2000. Since then, he is with the Faculty of Informatics at AAU. Presently, he is a PhD candidate at the International Institute of Information Technology (IIIT) Hyderabad, India. His research interests include Document Image Analysis, Artificial Intelligence and Information Retrieval.

C. V. Jawahar received his PhD in 1998 from IIT Kharagpur, India. He worked with Center for Artificial Intelligence and Robotics till Dec 2000 and since then he is with IIIT Hyderabad, India. He is presently an Associate Professor. His areas of research are Document Image Analysis, Content Based Information Retrieval and Computer Vision.