Gaussian Mixture Models for Arabic Font Recognition

Fouad SLIMANE[1,2] - Slim KANOUN[2] - Adel M. ALIMI[2] - Rolf INGOLD[1] - Jean HENNEBERT[1,3]
[1]*DIVA: Document, Image and Voice Analysis research group , Department of Informatics*
University of Fribourg (unifr), Bd de Pérolles 90, CH-1700 Fribourg, Switzerland
[2]*REGIM: REsearch Group on Intelligent Machines*
University of Sfax, National School of Engineers (ENIS), BP 1173, Sfax, 3038, Tunisia
[3]*Business Information System Institute, HES-SO // Wallis, Sierre,Switzerland*
Fouad.Slimane@unifr.ch, Slim.Kanoun@yahoo.fr, Adel.Alimi@ieee.org,
Rolf.Ingold@unifr.ch, Jean.Hennebert@hevs.ch

Abstract

We present in this paper a new approach for Arabic font recognition. Our proposal is to use a fixed-length sliding window for the feature extraction and to model feature distributions with Gaussian Mixture Models (GMMs). This approach presents a double advantage. First, we do not need to perform a priori segmentation into characters, which is a difficult task for arabic text. Second, we use versatile and powerful GMMs able to model finely distributions of features in large multi-dimensional input spaces. We report on the evaluation of our system on the APTI (Arabic Printed Text Image) database using 10 different fonts and 10 font sizes. Considering the variability of the different font shapes and the fact that our system is independent of the font size, the obtained results are convincing and compare well with competing systems.

1. Introduction

There has been relatively few works dedicated to the identification of Arabic fonts. It is although an important component for any robust and reliable multi-fonts OCRs (Optical Character Recognition). Font recognition can be combined with OCRs using either a priori or a posteriori approaches [8]. A priori approaches recognize fonts as a preliminary step of the text recognition. This procedure has the advantage of allowing the use of mono-font text recognition systems where the task is made easier thanks to a reduced variability of shapes. This is especially true for Arabic, where characters change shape depending to the used font and where there is a large intrinsic variability of character shapes according to their position in a word (beginning, middle, end or isolated). With a posteriori approaches, the information about the font is injected after the text recognition step, as a post-processing allowing to filter out incorrect OCR hypothesis. In both cases, font recognition improves OCRs' performance.

While not numerous, different approaches for Arabic font identification have been proposed in the past. In [2], Arabic font identification is based on some fine tuning of wavelet parameterization combined with Radial Basis Function neural networks for classification. Various tests were performed using pseudo-words, nine fonts and five different font sizes ranging from 12 to 36. The reported recognition rate is quite good reaching 99%, however we have to take into consideration the limited set of font sizes from 12 points up. In [7], font recognition is performed using fractal dimensions computed on blocks of images. This system was tested on 10 fonts and five different sizes. The best recognition rate obtained reached 96.5%. In [3], a system for Farsi font identification is presented, based on Sobel and Roberts gradients computed in 16 directions. The reported recognition rate is 94.2% using 5000 samples of 10 popular Farsi fonts.

Globally, there exists more than 450 different Arabic fonts. Developing a system that takes into account all these variations would be difficult and probably useless as the mostly used fonts in real-life applications are not so numerous. We therefore focus in this work on 10 widely used fonts that are representative of the different forms one can encounter in Arabic documents. Our system is also multi-size and has been evaluated on 10 different sizes. A novelty of our work is in the treatment of small sized fonts down to 6 points and where anti-aliasing filtering is applied. Such conditions covers

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.532

IEEE computer society

inputs taken from screen-based capture tools.

Regarding the proposed system, the novelty of our approach is in the use of a fixed-length sliding window for the feature extraction and in the use of Gaussian Mixture Models (GMMs) for the computation of likelihood estimates of font categories. This approach presents a double advantage. First, no a priori segmentation into characters is needed, which is an important feature for Arabic text where characters are tied to each other and difficult to separate. Second, we use versatile and powerful GMMs able to model finely the distributions of our features that are quite wide-scoped and largely multi-dimensional. Interestingly, we use the same features as for our previous work on Arabic text recognition [4].

The paper is organized as follows. In Section 2, we introduce the specificities of calligraphic Arabic script. In section 3, we present the database used for the evaluation of our system. Section 4 gives more details on our font recognition system. Results are discussed in Section 5 and are followed by our conclusions.

2. Characteristics of calligraphic Arabic script

The Arabic language is spoken by more than 300 million people. Arabic script is also very important in the Arabic culture and its style changes from one region to another, from one extreme formal simplicity to the full complexity of the arabesque. Arabic script is semi cursive both in printed and handwritten forms. It is written from right to left. Some Arabic letters change their shapes according to their position (beginning, middle, end, isolated) in the word. More details are in [4].

Figure 1. Examples of Arabic fonts

To our knowledge, there are over 450 Arabic fonts, all of which are used somewhere in the Muslim world. Figure 1 shows some of the mostly used fonts in Arabic: (A) Andalus, (B) Arabic Transparent, (C) AdvertisingBold, (D) Diwani Letter, (E) DecoType Thuluth, (F) Simplified Arabic, (G) Tahoma, (H) Traditional Arabic, (I) DecoType Naskh, (J) M Unicode Sara. They are the one we decided to use in our experiment. These fonts cover varied complexity of shapes of Arabic printed characters, from simple fonts with no or few overlaps and ligatures (AdvertisingBold) to more complex fonts rich in overlaps and ligatures (Diwani Letter).

3. APTI database

To evaluate our system, we used some parts of the large APTI (Arabic Printed Text Image) database [6]. APTI is freely available to the scientific community[1]. The APTI database was created in low-resolution "72 dot/inch" with a lexicon of 113,284 different Arabic words, 10 fonts, 4 styles and 10 different sizes. It contains more than 45 million Arabic word images representing more than 250 million different character shapes. Each word image in the APTI database is fully described using an XML file containing ground truth information about the sequence of characters as well as information about its generation. All Arabic letters have been adequately represented in the database. 120 labels were used in APTI to describe characters, taking into account their positions (beginning, middle, end, isolated). APTI is divided into 6 sets, 5 of which are freely available to the scientific community. The sets have been designed so that the number of words and representations of letters are very close from set to set (for more details about data dispersion, see [5]). In our tests, we use 1000 word images for each font and size. With 10 fonts and 10 font sizes (6, 7, 8, 9, 10, 12, 14, 16, 18 and 24), 100,000 word images are used in the training phase and an additional 100,000 different word images are used for the test phase. We choose to use APTI as we plan to develop a robust Screen based OCR for Arabic printed text on low resolution and open vocabulary.

4. System Description

Our system includes two parts. The first part is a front-end for the pre-processing of the images and for the feature extraction. The second one computes likelihood estimators of each font categories. In our approach, we make the assumption that the system receives as input an image where an Arabic word or a sequence of Arabic words is available. The segmentation into words or lines is assumed to be performed.

[1] http://diuf.unifr.ch/diva/APTI/

4.1. Pre-processing and Feature Extraction

Each word image is kept in gray-level and normalized into a fixed height size H of 45 pixels. The input image is then transformed into a sequence X of N feature vectors $\{x_1, \ldots, x_N\}$. Each feature vector x_n is computed from a $H \times W$ narrow analysis window sliding from right to left over the word image. In our settings, the analysis window has a constant width W of 8 pixels and is shifted by 1 pixel. The normalization size of 45 pixels and the 8 pixels window widths have been found to give optimal results.

The sliding window procedure do not require any segmentation into letters. The feature extraction itself is divided into three parts. The first part extracts, for each window:

- the number $N1$ of connected black components; the number $N2$ of connected white components; the ratio $N1/N2$;
- the relative vertical position of the smallest connected black component;
- the sum of perimeter P_c of all components c divided by the perimeter of the analysis window P_w;
- the compactness $(4\pi A)P^2$ where P is the shape perimeter in the window and A the area;
- the gravity centre of the window, of its right and left half parts, and of the first third, the second and the last part of the window: $\sum_{i=1}^{n} \frac{x_i}{nW}$; $\sum_{i=1}^{n} \frac{y_i}{nH}$ where W is the with and H the height of the window;
- the $\log$ of the estimated baseline position; the relative vertical position of baseline;
- the number of extrema in vertical projection; the number of extrema in horizontal projection.

The second part of the feature extraction consists in resizing the window into a normalized size of 20 pixels height and computing the horizontal and vertical projection values. The feature extraction, overall, results in a vector $x_n = \left[x_n^1, \ldots, x_n^{51}\right]$ of 51 coefficients.

The third part consists in the computation of so-called delta coefficients between two adjacent vectors using the following formula:

$$\Delta x_n^j = x_{n+1}^j - x_{n-1}^j, \quad \forall 1 < j < 51$$
$$\Delta x_n^j = x_n \quad where \quad n = 0 \quad or \quad n = N$$

Using the delta to complete the feature vector, we have 102 coefficients computed for each analysis window.

4.2. Modeling Likelihoods with GMMs

GMMs are used to model the likelihoods of the features extracted from the image. GMMs are well-known versatile and flexible modeling tools able to approximate any probability density function. With GMMs, the probability density function $p(x_n|M_f)$ or *likelihood* of a D-dimensional feature vector x_n given the model of a font category M_f, is estimated as a weighted sum of multivariate Gaussian densities

$$p(x_n|M_f) \cong \sum_{i=1}^{I} w_i \mathcal{N}(x_n, \mu_i, \Sigma_i)$$

in which I is the number of mixtures and w_i is the weight for mixture i. The Gaussian densities $\mathcal{N}$ are parameterized by a mean $D \times 1$ vector μ_i, and a $D \times D$ covariance matrix, Σ_i. In our case, we make the hypothesis that the features are uncorrelated and we use diagonal covariance matrices. By making the hypothesis of feature vector independence, the global *likelihood* score for the sequence of feature vectors, $X = \{x_1, \ldots, x_N\}$ is computed with

$$S_f = p(X|M_f) = \prod_{n=1}^{N} p(x_n|M_f)$$

Assuming equal a priori probabilities $P(M_f)$ of each font f, the font recognition is performed by electing the font f^* leading to the highest value of S_f. As the local likelihood values $p(x_n|M_f)$ are usually very small, the global likelihood S_f is usually expressed in the log domain to avoid running below machine representation limits.

During training time, the Expectation-Maximization (EM) algorithm is used to iteratively refines the component weights, means and variances to monotonically increase the likelihood of the training feature vectors [1]. In our experiments we used the EM algorithm to build the models by applying a simple binary splitting procedure to increase the number of Gaussian mixtures through the training procedure up to 2048 mixtures.

From a practical point of view, GMMs can be seen as one-state Hidden Markov Models. We therefore used the HTK toolkit to implement our modeling scheme. At recognition time, an ergodic HMM including all font models is built and the best path in this model simply determines the winner font using the standard Viterbi decoding procedures available in HTK. Performances are evaluated in terms of font recognition rates using an unseen set of word images.

5. Experimental results

All results are obtained with ten font sizes (6, 7, 8, 9, 10, 12, 14, 16, 18 and 24) and the ten fonts illustrated in Figure 1. Arabic transparent and Simplified Arabic Fonts show a strong morphological resemblance, the only difference is in the inter-character horizontal elongation. These two fonts allow us to evaluate the performance of our approach in the case of similar fonts.

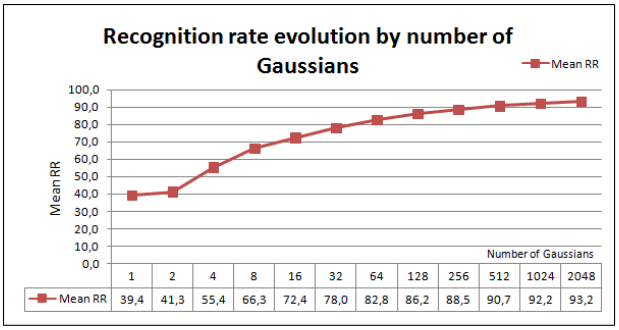

Figure 2. Evolution of recognition rate

One of the main advantage of GMMs is in their capacity to model complex forms of probability density functions. Increasing the number of mixtures I allows to model more finely the probability density functions. On the other hand, increasing I means that more parameters have to be estimated, which usually requires larger training sets. It also means more computations. Our first set of results explore the impact of the number of mixtures I on the recognition rate. As illustrated on Figure 2, we observe the mean rate of our Arabic font recognition increasing from 39.4% with 1 Gaussian mixture to 93.2% (when "Arabic transparent" and "Simplified Arabic" are separately considered) with 2048 Gaussian mixtures. As shown on the curve evolution, going above 2048 mixtures will not lead to significantly better results. We therefore used 2048 for the results presented below.

Table 1. Font recognition rate

Font	***RR***	***Font***	***RR***
Advertising Bold	**99.7**	*Andalus*	**99.8**
Arabic Transparent	**98.7**	*M Unicode Sara*	**99.8**
Tahoma	**99.7**	*Simplified Arabic*	**98.2**
Traditional Arabic	**99.1**	*DecoType Naskh*	**97.3**
DecoType Thuluth	**99.3**	*Diwani Letter*	**99.6**
Mean RR		**99.1**	

On average, we measure a global performance of 93.2%. Due to the morphological similarity between "Arabic transparent" and "Simplified Arabic", we observed that the recognition rates of theses fonts are pretty low. We performed a confusion matrix analysis, clearly showing that most of errors are between these similar fonts. We have extended the test by considering "Arabic transparent" and "Simplified Arabic" as a single font in training and recognition step. A significant improvement in recognition rate was recorded, leading to a global recognition rate of 99,1%, see table 1.

The overall system can be considered as quite robust in terms of performances considering that the inputs of the APTI database are images of single words. Better results could potentially be obtained in the case of larger length inputs such as lines or block of texts.

6. Conclusion and Future Work

We presented in this paper a simple but efficient system for the identification of Arabic fonts from multi-size, multi-font image inputs. The main novelty of our proposal is in the use of GMMs for the estimation of font category likelihoods themselves computed from local feature likelihoods. The feature extraction is using a fixed-length window sliding from right to left on the word image. The main advantage of this approach is that an a priori segmentation into characters is not needed. Conceptually, the models are capturing the feature distributions independently to any characters and dependently to the font categories. The evaluation of the system on a large database shows that performances above 99% can be reached on a set of 9 different fonts and 10 different sizes. Another advantage lies in the fact that a similar system architecture can be used for text recognition [4], sharing the same feature extraction front-end. In future work, we will evaluate a full multi-font, multi-size recognition system, using the font identification system presented here as preliminary step.

References

[1] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Royal Statistical Society*, 39(1):1–38, 1977.

[2] N. Essoukri Ben Amara and S. Gazzah. Une approche d'identification des fontes arabes. *CIFED*, (8):21–25, 2004.

[3] H. Khosravi and E. Kabira. Farsi font recognition based on sobel-roberts features. *PRL*, 31(1):75–82, 2010.

[4] F. Slimane, R. Ingold, A. M. Alimi, and J. Hennebert. Duration models for arabic text recognition using hidden markov models. *CIMCA*, pages 838–843, 2008.

[5] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert. Database and evaluation protocols for arabic printed text recognition. *DIUF-University of Fribourg - Switzerland*, 2009.

[6] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert. A new arabic printed text image database and evaluation protocols. *ICDAR*, pages 946–950, 2009.

[7] N. Zaghden, S. BenMoussa, and A. M. Alimi. Reconnaissance des fontes arabes par l'utilisation des dimensions fractales et des ondelettes. *CIFED*, pages 277–282, 2006.

[8] A. Zramdini. *Study of optical font recognition based on global typographical features*. University of Fribourg, Theses n°1106, 1995.