21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

New Features for Complex Arabic Fonts in Cascading Recognition System

Fouad Slimane[1,2] - Oussama Zayene[1] - Slim Kanoun[4] - Adel M. Alimi[2]
Jean Hennebert[1,3] - Rolf Ingold[1]
[1]*DIVA group , Department of Informatics, University of Fribourg (unifr), Switzerland*
[2]*REGIM lab., University of Sfax, National School of Engineers (ENIS), Tunisia*
[3]*ICT Institute, University of Applied Science, HES-SO // Fribourg, Switzerland*
[4]*University of Sfax, National School of Engineers (ENIS), BP 1173, Sfax, 3038, Tunisia*
{*fouad.slimane, oussama.zayene, jean.hennebert, rolf.ingold*}*@unifr.ch*
slim.kanoun@gmail.com, adel.alimi@ieee.org

Abstract

We propose in this work an approach for automatic recognition of printed Arabic text in open vocabulary mode and ultra low resolution (72 dpi). This system is based on Hidden Markov Models using the HTK toolkit. The novelty of our work is in the analysis of three complex fonts presenting strong ligatures: DiwaniLetter, DecoTypeNaskh and DecoTypeThuluth. We propose a feature extraction based on statistical and structural primitives allowing a robust description of the different morphological variability of the considered fonts. The system is benchmarked on the Arabic Printed Text Image (APTI) database.

1. Introduction

Text images included in web pages are usually compressed and generated at very low resolution (72-100 dpi) to speed up downloading. Images from screen captures share also these characteristics. Despite their relative effectiveness in high resolution (> 150 dpi), classical Arabic Optical Character Recognition (AOCR) systems usually show poor performance on low-resolution text. This was highlighted during the Arabic Recognition Competition held in ICDAR'11 [10].

Several systems based on Hidden Markov Models (HMMs) have been developed for the recognition of Arabic text [3, 6, 4, 7, 8, 1]. The success of HMMs is essentially in their good capacity to integrate context and to model variabilities. They also have the advantage to provide a character implicit segmentation when decoding the sequence of observations.

In this work, we focus on open vocabulary, offline Arabic text recognition at ultra low resolution, taking into account anti-aliasing effects. The recognition is based on HMMs using the HTK toolkit [13]. The novelty of our work is in the analysis of such systems for three Arabic fonts that present complexities with strong ligatures and overlaps between characters: *Diwani Letter*, *DecoType Thuluth* and *DecoType Naskh*.

This paper is organized as follows. The characteristics of Arabic fonts are presented in Section 2. In Section 3, we describe the proposed approach. The evaluation of our system with the APTI database is described in the fourth section. Results are presented and discussed in Section 4.

2. Characteristics of Arabic fonts

The Arabic script varies between communities and regions. There are scripts with extreme formal simplicity, for example the *Simplified Arabic* font which has no ligatures. Other scripts present complex arabesques such as, the strongly ligatured *Diwani Letter* font. There are more than 450 Arabic fonts of which only a few are commonly used in the Arab-Muslim world [12].

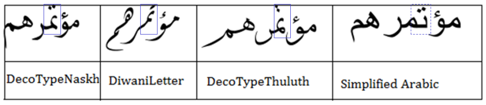

Figure 1. Same Arabic word in four different fonts

In this work, we focus on three fonts that present complexities and still are often used: *Diwani Letter*, *DecoType Thuluth* and *DecoType Naskh*. Figure 1

978-4-9906441-0-9 ©2012 ICPR

shows an example of the same word generated in four different fonts. The word image presented in *Simplified Arabic* font has neither ligation nor overlap between the characters while in the three other fonts, they present different types of ligatures and overlaps.

3. Proposed approach

The proposed word recognition system is based on HMMs. It was developed using the HTK toolkit[1]. As shown in Figure 2, it works in two phases: training and recognition. For both phases, we perform the same preprocessing and we extract the same features.

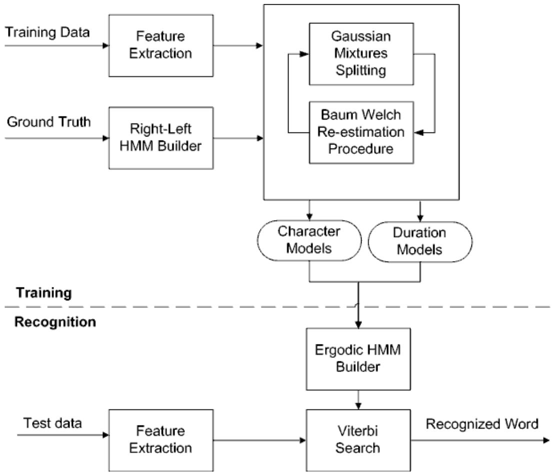

Figure 2. Word recognition system using Hidden Markov Models

3.1. Preprocessing

Preprocessing aims at preparing the word images for the feature extraction phase. As explained later, we will use multiple types of features, either based directly on the images in gray level, either based on the outline or skeleton of the image such as Fourier descriptors and Freeman directions. We used binarization and skeletonization as preprocessing for some of the features.

3.2. Feature extraction

The feature extraction is based on a sliding window technique used by many researchers [7, 1, 8, 5, 2]. The configuration of the sliding window is different from one researcher to another. In our case, we used 6 pixels width and an overlap between two successive windows of 1 pixel. Each window is further divided vertically into N cells with, for example, $N = 7$ for the Diwani Letter font. As presented in more details in our previous work, a set of common features for the three fonts is extracted in each window [12]. Specific features are further extracted and selected for each font as explained later in Section 4.2. Each word image is then transformed into a matrix of values where the number of lines corresponds to the number of analysis windows, and the number of columns M is equal to the number of feature coefficients in each feature vector. Each feature vector includes $M/2$ features concatenated with $M/2$ so-called delta coefficients computed as in [12].

3.3. Training and recognition

The idea is to represent each word by a HMM and each character by a sub-model with fixed number of states. Figure 3 shows an example of a word formed of three characters. Each character is represented by a HMM with five states. Special non-emitting states are used for start (S) and end (E) of characters and word.

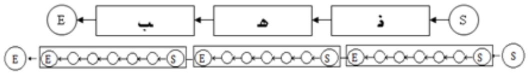

Figure 3. Example of HMM for a 3 characters Arabic word

- **Training phase**. Right-left HMMs corresponding to the known sequence of characters are prepared for each word, together with the features file. Once the HMM parameters are initialized, an embedded iterative training of character models is performed using a Baum-Welch procedure.
- **Recognition phase**. The recognition is done using the Viterbi algorithm applied to an ergodic HMM defined by the set of all character models. In the ergodic model, all transitions between character submodels are possible, allowing it to recognize any Arabic word.

4. System evaluation

4.1. The APTI database

To evaluate the performance of the proposed system, experiments were performed on a subset of the Arabic Printed Text Image (APTI) database. APTI[2] is a freely available database based on a large vocabulary and with

[1] http://htk.eng.cam.ac.uk/

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

images at ultra low resolution (72 dpi) for multi-font, multi-size and multi-style Arabic text recognition. This database was developed in 2009 by Slimane et al. [9] in a collaboration between the DIVA group from University of Fribourg, Switzerland and the REGIM lab. from University of Sfax, Tunisia. The database is synthetic, generated using a lexicon of 113,284 different Arabic words, 10 fonts, 10 sizes and 4 font-styles.

4.2. Tests and experimental results

Baseline system. Our baseline system is similar to the one presented in [11], using the set of features presented in [12]. The system is evaluated on font size 24 using font Andalus and the complex three fonts presented earlier: *Diwani Letter*, *DecoType Thuluth* and *DecoType Naskh*. Our system is a cascading system working in two steps: font recognition using the system presented in [12] and word recognition using font-dependent models as presented in Figure 2. For each font, 75,557 word images (set 1,2,3 and 4) were used in the training phase and an additional 18,868 different images (set 5) were used for the test phase according to the protocols described with APTI. The results for the baseline system are shown in column A of Table 1. Results are reported as recognition rates at the word level (WRR) and character level (CRR). We note that, while the results are very good for the font Andalus, the observed results are very low for the three other fonts. An explanation is clearly to be found in the complex morphology of these fonts presenting ligatures, overlaps and collisions between characters and diacritical marks. Horizontal lengthening is also present in the case of *DecoType Naskh*. To deal with these difficulties, we performed an analysis and design of relevant features able to cope with the specificities of each font morphologies.

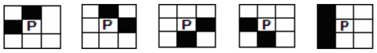

Figure 4. Five types of concavity configurations for a background pixel P

Selection of features. We tested a large set of features including the histogram of the Freeman directions, Zernike moments, Fourier descriptors, etc. The best features choice for each font has been implemented using a systematic selection procedure to maximize the recognition rate. Some features are based on a partitioning of the analysis windows into three parts according to upper and lower baselines automatically detected using a classical horizontal pixel projection method (see Figure 5).

Table 1. Results of the baseline system with common features (A); with selected feature for each font (B); with selected feature for each font and additional ligatured character models (C)

System	A		B		C	
Font	WRR	CRR	WRR	CRR	WRR	CRR
Andalus	94.2	99.2	–	–	–	–
Diwani Letter	49.4	86.6	93.7	98.1	95.1	98.6
DecoType Naskh	52.3	90.8	82.2	97.0	84.5	97.5
DecoType Thuluth	58.2	91.7	90.3	97.5	92.0	98.1
Mean cascading recognition system results			***88.7***	***97.5***	***90.5***	***98.1***

Common features for Diwani Letter, DecoType Naskh and DecoType Thuluth

- the number $N1$ of connected black components; the number $N2$ of connected white components; the ratio $N1/N2$; position difference between the gravity centers g of the black pixels in two consecutive windows: $f = g(t) - g(t-1)$
- the vertical position of the gravity center of the black pixels divided by the low baseline position
- the vertical position of the smallest connected black component divided by the window height
- 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
- density of black pixels in the window; density of black pixels below the low baseline; density of black pixels above the low baseline; density of black pixels in each column of the window
- gravity center of the window, of the right and left half and of the first third, the second and the last part of the window divided vertically
- number of black/white transitions between cells: $f = \sum_{i=2}^{n_c} |b(i) - b(i-1)|$ with n_c the number of cells in the window
- number of black/white transitions between cells located above the low baseline
- index of the area where the gravity center is in the window (up area = 1, medium area = 2, down area = 3), the areas are determined by the baseline positions
- number of sub-windows that belong to one of the five configurations shown in Figure 4, normalized by the number of sub-windows
- number of background pixels in the five configurations mentioned above but only for pixels located in the middle area of writing, between the

two baselines

Specific features for Diwani Letter

- the histogram of the 8 Freeman directions
- the Zernike moments (12 moments)
- the sum of the gradient norms
- the size of the smallest connected black component divided by the window height

Specific features for DecoType Naskh

- the moment invariants (7 moments)
- number of background pixels in the five configurations mentioned above (1) only for pixels located in the lower area of writing, below the lower baseline; (2) only for pixels located in the upper area of writing, over the upper baseline

Specific features for DecoType Thuluth

- the histogram of the 8 Freeman directions
- the Zernike moments (12 moments)
- the Fourier descriptors (9 descriptors)

As shown in Table 1, column (B), a significant improvement in comparison to the baseline system can be obtained using the common and specific features for each font.

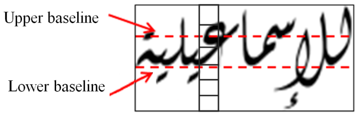

Figure 5. Upper and lower baselines on sample data

Ligatured character models. An analysis of the errors done by the system showed that the majority of deletion and insertion errors comes from ligatures between characters. For this reason, we have created new sub-models representing ligatured characters. As shown in Table 1, column (C), these new models improve significantly the system results with a global average of 90.5 % for word recognition and 98.1 % for character recognition in all three fonts.

5. Conclusion

In this paper, we showed that the performance of a baseline recognition system for off-line Arabic printed text can be significantly degraded when used on complex fonts where character overlaps and ligatures are numerous. We also showed that good performances can be regained by introducing dedicated feature extractions on these fonts. If we consider multi-font systems, the results reported here clearly encourage so-called cascade systems where a first sub-system recognizes the font before feeding a second sub-system for the text recognition with dedicated feature extraction and modeling.

References

[1] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji. Recognition of off-line printed arabic text using hidden markov models. volume 88, pages 2902–2912. 2008.

[2] J. H. AlKhateeb, O. Pauplin, J. Ren, and J. Jiang. Performance of hidden markov model and dynamic bayesian network classifiers on handwritten arabic word recognition. volume 24, pages 680–688. 2011.

[3] N. Ben Amara, A. Belaïd, and N. Ellouze. Utilisation des modèles markoviens en reconnaissance de l'écriture arabe : état de l'art. In *CIFEd*, Lyon, France, 2000.

[4] N. Ben Amor and N. Ben Amara. Combining a hybrid approach for features selection and hidden markov models in multifont arabic characters recognition. In *DIAL*, pages 5 pp. –107, 2006.

[5] H. Cao, R. Prasad, and P. Natarajan. Handwritten and typewritten text identification and recognition using hidden markov models. In *ICDAR*, pages 744–748, 2011.

[6] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel. Arabic handwriting recognition using baseline dependant features and hidden markov modeling. In *ICDAR*, pages 893–897, 2005.

[7] M. S. Khorsheed. Offline recognition of omnifont arabic text using the hmm toolkit (htk). volume 28, pages 1563–1571. 2007.

[8] 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.

[9] 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.

[10] F. Slimane, S. Kanoun, H. Abed, A. Alimi, R. Ingold, and J. Hennebert. Icdar 2011 - arabic recognition competition: Multi-font multi-size digitally represented text. In *ICDAR*, pages 1449 –1453, sept. 2011.

[11] F. Slimane, S. Kanoun, A. M. Alimi, J. Hennebert, and R. Ingold. Comparison of global and cascading recognition systems applied to multi-font arabic text. In *DocEng*, pages 161–164, 2010.

[12] F. Slimane, S. Kanoun, A. M. Alimi, R. Ingold, and J. Hennebert. Gaussian mixture models for arabic font recognition. *ICPR*, pages 2174–2177, 2010.

[13] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.