Recognising Cursive Arabic Text using a Speech Recognition System

M S Khorsheed

Abstract— This paper presents a system to recognise cursive Arabic typewritten text. The system is built using the Hidden Markov Model Toolkit (HTK) which is a portable toolkit for speech recognition system. The proposed system decomposes the page into its text lines and then extracts a set of simple statistical features from small overlapped windows running through each text line. The feature vector sequence is injected to the global model for training and recognition purposes. A data corpus which includes Arabic text from two computer-generated fonts is used to assess the performance of the proposed system.

Keywords— Document Analysis, Pattern Analysis and Recognition, Machine Vision, Arabic OCR, HTK

I. INTRODUCTION

AMONG the branches of pattern recognition is the automatic reading of a text, namely, text recognition. The objective is to imitate the human ability to read printed text with human accuracy, but at a higher speed.

Most optical character recognition methods assume that individual characters can be isolated, and such techniques, although successful when presented with Latin typewritten or typeset text, cannot be applied reliably to cursive script, such as Arabic. Previous research on Arabic script recognition has confirmed the difficulties in attempting to segment Arabic words into individual characters [1].

Arabic language provides a rich source of technical challenges for recognition algorithms. The most obvious characteristics of the Arabic language is that Arabic scripts are inherently cursive; writing isolated characters in 'block letters' is an unacceptable and unused writing style. The shape of the letter is context sensitive, depending on its location within a word. For example a letter as 'ه' has four different shapes: isolated 'ه' as in 'عبده', beginning 'هـ' as in 'هند', middle 'ـهـ' as in 'بهر' and end 'ـه' as in 'منه'. Since not all letters connect, word boundary location becomes an interesting problem, as spacing may separate not only words but also certain characters within a word.

Hidden Markov Models (HMMs) [2] are among other classification systems that are used to recognise character, word or text. They are statistical models which have been found extremely efficient for a wide spectrum of applications, especially speech processing. This success has motivated recent attempts to implement HMMs in character recognition whether on-line [3] or off-line [4]. The HMM provides an explicit representation for time-varying patterns and probabilistic interpretations that can tolerate variations in these patterns.

Computer & Electronics Research Institute, King AbdulAziz City for Science and Technology (KACST), P O Box 6086, Riyadh 11442, Saudi Arabia. Email: mkhorshd@kacst.edu.sa

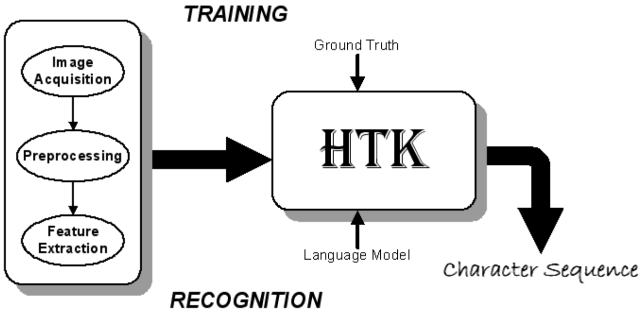


Fig. 1 A block diagram of the HTK-based Arabic recognition system.

In off-line recognition systems, the general idea is to transform the word image into a sequence of observations. The observations produced by the training samples are used to tune the model parameters whereas those produced by the testing samples are used to investigate the system performance.

This paper presents a HMM–based system to recognise cursive Arabic script offline. Statistical features are extracted from the text line image and fed to the recogniser. The system is built on the Hidden Markov Models Toolkit (HTK) [5]. This is primarily designed for building HMM-based speech processing tools in particular recognisers. The proposed system is lexicon free and it depends on the technique of character models and grammar from training samples.

II. SYSTEM OVERVIEW

Fig 1 illustrates the block diagram of the proposed system. The system is divided into stages. The first stage is performed prior to HTK, and includes: image acquisition, preprocessing and feature extraction. The objective is to acquire the document image, preprocess it and then decompose it into text line images. Each line image is transfered into a sequence of feature vectors. Those features are extracted from overlapping vertical windows along the line image, then clustered into discrete symbols.

Stage two is performed within HTK. It couples the feature vectors with the corresponding ground truth to estimate the character model parameters. The final output of this stage is a lexicon–free system to recognize cursive Arabic script. During recognition, an input pattern of

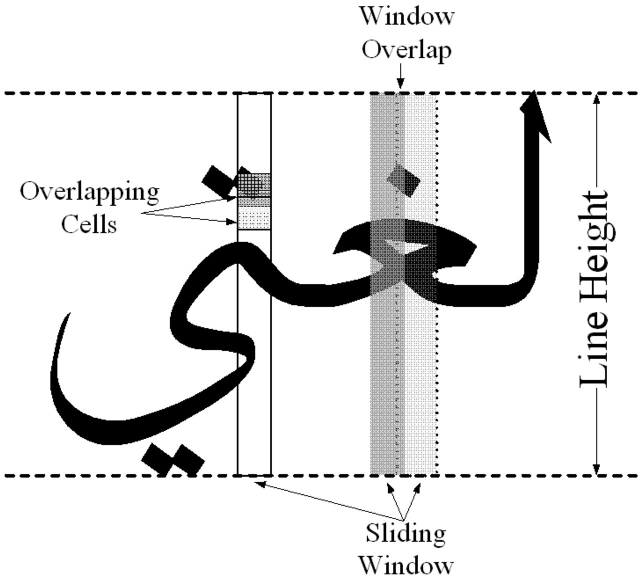
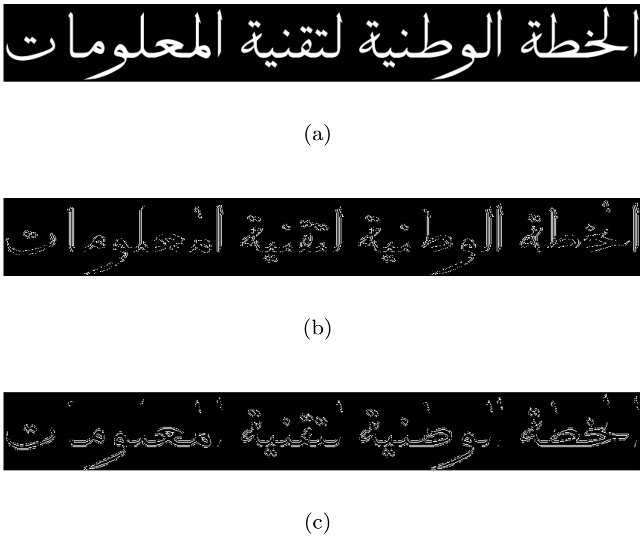


Fig. 2 Dividing the line into windows and cells.



(a)

(b)

(c)

Fig. 3 Feature extraction: a)original line image. b)vertical derivative of (a). c)horizontal derivative of (a).

discrete symbols representing the line image is injected to the global model which outputs a stream of characters matching the text line.

A. Feature Extraction

This research implements HMMs to recognise the input pattern. This implies that the feature vector, extracted from the text, is computed as a function of independent variable. In speech, the cepstral features are extracted from the speech signal with respect to time. Similarly, in on–line handwritten recognition a feature vector is computed as a function of time also. In off–line recognition system the case is different; there is no independent variable. Moreover, the whole page image needs to be recognised. In this research, the text line has been chosen as the unit for training and recognition purposes.

Now, assuming that the horizontal position along the text line is the independent variable, a sliding window is scanning the line from right to left. A set of simple features is extracted from pixels falling within that window. The resulting feature vector is mapped against predefined codebook vectors, and replaced with the symbol representing the nearest codebook vector. This step transfers the text line image into a sequence of discrete symbols.

Each line image is divided into overlapped narrow windows, see Fig 2. These windows are then vertically divided into cells where each cell includes a predefined number of pixels. Those cells are used for feature extraction.

Features extracted from the text could be structural [6], spectral [7] or as per here statistical. Statistical features are easy to compute and script independent. They avoid any segmentation at word or character level. These features are: intensity, intensity of horizontal derivative and intensity of vertical derivative, see Fig 3.

The intensity feature represents the number of ones in each cell. The intensity of horizontal derivative detects the edge through X–axis, then computes the number of ones in the resulting cell. The intensity of vertical derivative detects the edge through Y–axis, then computes the number of ones in the resulting cell. The feature vector of one narrow window is built by stacking features extracted from each cell in that window.

B. HTK Inference Engine

The hidden Markov model toolkit (HTK) [5] is a portable toolkit for building and manipulating hidden Markov models. It is primarily designed for building HMM–based speech recognition systems. HTK was originally developed at the Speech Vision and Robitics Group of the Cambridge University Engineering Department (CUED).

Much of the functionality of HTK is built into the library modules available in C source code. These modules are designed to run with the traditional command line style interface, so it is simple to write scripts to control HTK tools execution. The HTK tools are categorized into four phases: data preparation, training, testing and result analysis tools.

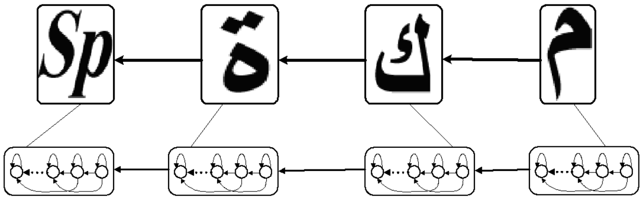
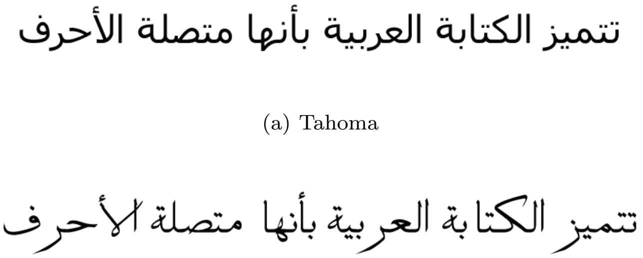


Fig. 4 HMM structure of the word Makkah “مكة”, sp denotes space character.


تتميز الكتابة العربية بأنها متصلة الأحرف
تتميز الكتابة العربية بأنها متصلة الأحرف

(a) Tahoma

(b) Thuluth

Fig. 5 Text sample written in Tahoma and Thuluth fonts.

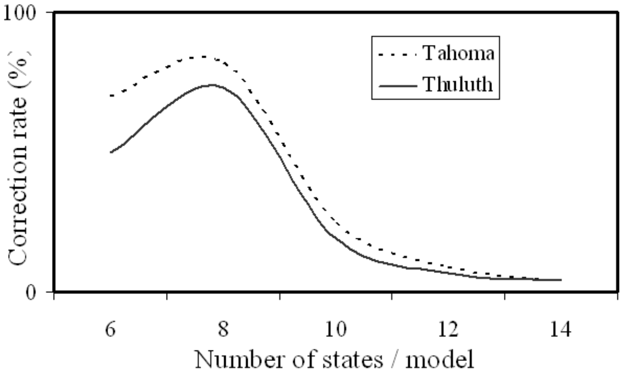


Fig. 6 The impact of the number of states per model on system performance.

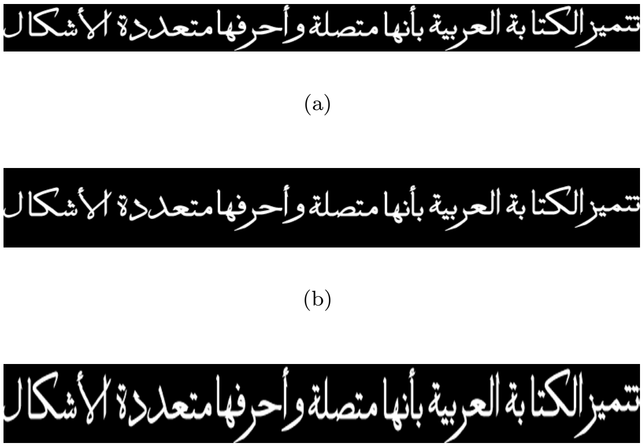
C. Network Hierarchy

The global model is a network of interconnected character models. Each character-model represents different shapes of a single letter in the alphabet. This scheme is refered to as mono-model. Mono-models are context-independant where each character in the alphabet is represented by a distinct HMM. Characters are separated from their preceding and succeeding neighbors within the word, see Fig 4.

III. EXPERIMENTAL RESULTS

The performance of the proposed system was assessed using a corpus which includes more than 100 pages of Arabic text in Tahoma font and another 100 pages of Arabic text in Thuluth font, see Fig 5. Tahoma is a simple font with no overlap or ligature. In contrast, Thuluth is a very rich font with challenges; overlaps, ligatures and decorative curves. The data corpus includes 37162 words and 201396 letters, not including spaces.

After noise elimination and deskewing, the 200 pages were decomposed into more than 5000 line images. A set of experiments were performed on each font separately and the performance of the two fonts were then compared against each other. All experiments were performed using 1500 line images for training and 1000 line images for testing. All character models had the same number of states; 8 states. There is no mathematical method to calculate the optimal number of states per model. Alternatively, various values were examined to select the best number of states per model, see Fig 6


تتميز الكتابة العربية بأنها متصلة وأحرفها متعددة الأشكال
تتميز الكتابة العربية بأنها متصلة وأحرفها متعددة الأشكال
تتميز الكتابة العربية بأنها متصلة وأحرفها متعددة الأشكال

(a)

(b)

(c)

Fig. 7 Text lines with different line heights: (a) 43 pixels, (b) 60 pixels and (c) line image in (a) resized to 60 pixel size.

TABLE I

Cell size categories.

Category	Cell Size	Horizontal Overlap	Vertical Overlap
CS_1	3×3	-	-
CS_2	3×3	1 pixel	-
CS_3	3×3	2 pixels	-
CS_4	5×5	2 pixels	-
CS_5	5×5	2 pixels	2 pixels

Text line height is proportional to the font size. The line image is measured in pixels. For the corpus under consideration the line image height varies from 35 pixels to 95 pixels depending on the font type and size. To eleminate this dependency, all line images were resized to a single height value; 60 pixels. This value equals the mean of image heights of all text lines in the data corpus, see Fig 7.

A. Cell Size

At any horizontal position, the sliding window is divided into a number of cells, as shown in Fig 2. These cells may or may not overlap. The overlap can be vertical or horizontal and it increases the amount of features generated from a single line and hence increases the processing time.

The first set of experiments studied the cell size parameter. Variuos combinations of cell sizes and overlaps were tested, see Table I.

Fig 8 shows the correction rates of the five categories for the Tahoma and Thuluth fonts. The codebook here includes 64 clusters. The results illustrates that HTK was more efficiently tuned for Tahoma font rather than for Thuluth font. This is sensible due to decorative curves

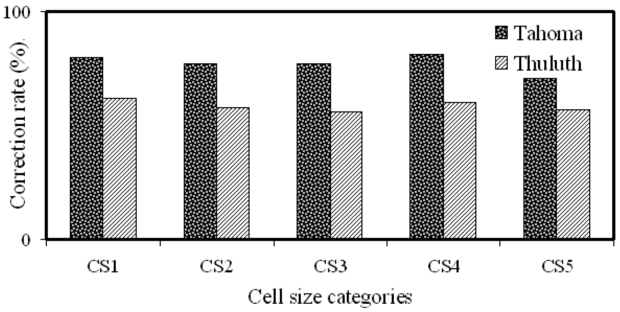


Fig. 8 System performance for the Tahoma and Thuluth fonts.

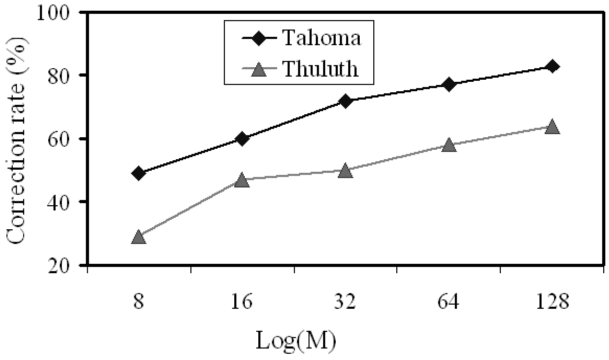


Fig. 9 Correction rates CR of CS_2 for Tahoma and Thuluth fonts.

found at the end of some letters in Thuluth font which overlap with succeeding letters.

B. Codebook Size

After extracting features from a line image, the feature vector dimensionality is reduced from 2D to 1D using K-means clustering algorithm [2]. Mapping the 2D feature vector to the nearest cluster is based on the minimum Euclidean distance measure.

This set of expriments studied the codebook size parameter. Five different codebook size values were examined for each font for each category: $8, 16, 32, 64$ and 128. A total number of 50 codebooks were created for this purpose: two fonts × five cell size categories × five codebook sizes. Fig 9 shows the correction rates of CS_2 for the two fonts. Again Tahoma font outperformed Thuluth font. The figure also illustrates the improvement in the system performance as the codebook size increases. This conclusion is also applicable to other cell size categories.

IV. CONCLUSION

A new system to recognise cursive Arabic text has been presented. The proposed system is based on HMM Toolkit basically designed for speech recognition purpose. Various model parameters have been studied using a corpus that includes data typewritten in a computer–generated font; Tahoma. The system was capable to learn complex ligatures and ovelaps. The system performance has been improved when implementing the tri–model scheme. Future work will concentrate on enlarging the data corpus to include more fonts.

References

[1] M. S. Khorsheed, “Off-line Arabic character recognition – A review,” Pattern Analysis and Applications, vol. 5, no. 1, pp. 31–45, 2002.

[2] L. Rabiner and B. Juang, Fundamentals Of Speech Recognition. Prentice Hall, 1993.

[3] H. Kim, K. Kim, S. Kim, and J. Lee, “On-line recognition of handwritten chinese characters based on hmms,” Pattern Recognition, vol. 30, no. 9, pp. 1489–1500, 1997.

[4] W. Kim and R. Park, “Off-line recognition of handwritten korean and alphanumeric characters using hmms,” Pattern Recognition, vol. 29, no. 5, pp. 845–858, 1996.

[5] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, The HTK Book. Cambridge University Engineering Dept., 2001.

[6] M. S. Khorsheed, “Recognising handwritten Arabic manuscripts using a single hidden markov model,” Pattern Recognition Letters, vol. 24, no. 14, pp. 2235–2242, 2003.

[7] M. S. Khorsheed, “A lexicon based system with multiple hmms to recognise typewritten and handwritten Arabic words,” in The 17th National Computer Conference, (Madinah, Saudi Arabia), pp. 613–621, 5-8 April 2004.