Offline Handwritten Arabic Word Recognition Using HMM - a Character Based Approach without Explicit Segmentation

Volker Märgner, Haikal El Abed, Mario Pechwitz

Offline Handwritten Arabic Word Recognition Using HMM - a Character Based Approach without Explicit Segmentation

Volker Märgner – Haikal El Abed – Mario Pechwitz

Technical University of Braunschweig
Institut for Communications Technology (IfN)
Department of Signal Processing for Mobile Information Systems
Schleinitzstrasse 22, 38106 Braunschweig, Germany

{v.maergner,elabed}@tu-bs.de and mp@ifnenit.com

Abstract : *This paper presents the IfN's Offline Handwritten Arabic Word Recognition System. The system uses Hidden Markov Models (HMM) for word recognition, and is based on character recognition without explicit segmentation. The first part of this paper deals with databases for word recognition systems, and in particular, the IFN/ENIT - database. The second part gives a short description of the pre-processing, normalisation, and feature extraction methods needed for this system. The final part gives a practical approach to the HMM-Recogniser used in our system and some results are presented.*

Index Terms : Offline Handwritten Recognition, Arabic Word Recognition, Segmentation, Hidden Markov Models, Competition, Database.

Résumé : *Cet article présente le système de reconnaissance des mots manuscrits arabes développé au sein du Laboratoire IfN. Le système utilise les techniques des Modèles Cachés de Markov (MMC ou Hidden Markov Model - HMM) pour la reconnaissance des textes, et est basé sur la reconnaissance de caractères sans segmentation explicite. La première partie de cet article traite l'importance des bases de données pour les systèmes de reconnaissances. La deuxième partie contient une courte description du prétraitement, de la normalisation, et des méthodes d'extraction des caractéristiques images requises pour ce système. La partie principale présente une approche pratique du HMM utilisé dans notre système.*

Mots-clés : Reconnaissance off-line des mots manuscrits, Manuscrits arabes, Segmentation, Modèles Cachés de Markov, Competition, Bases de données.

1 Introduction

Automatic recognition of handwritten words remains a challenging task despite the promising improvements in the latest recognition methods and systems. Especially concerning the automatic recognition of Arabic handwritten words, a lot of work must still be done. The most important requirement for the development and comparison of recognition systems is a large database combined with ground truth (GT) information. Compared to English text, where handwritten words and numbers have been publicly available for a long time, (e.g. CEDAR, NIST) the situation for Arabic today is quite different. Since in the case of Arabic handwritten words many papers use a specific, more or less small data set of their own (see e.g. [ALB 95] and [AMI 98]), or they present results on large databases that are not available to the public (see e.g. [KHA 99] and [YOU 03]), it follows that it is impossible to compare different results which would be important for improving existent methods. The IFN/ENIT - database, published at the CIFED 02 conference, is a first attempt to overcome this situation [10]. This database is available for free, for non-commercial use (www.ifnenit.com). Presently, more than 30 research groups all over the world use this database, with a first competition based on the IFN/ENIT-database organised at ICDAR 2005 [MäR 05]and a second competition announced for ICDAR 2007.

Methods to recognize handwritten words are well known and widely used for many different languages. In opposition to printed text in most languages, the characters in cursive handwritten words are connected. In recent years, methods based on Hidden Markov Models (HMM) particularly, have been very successfully used for recognising cursively handwritten words. It is quite obvious that in the case of a limited lexicon a recognition system using HMM methods should give good results. Yet it is also clear that the great difference in the shape of handwritten characters between Latin and Arabic requires not only a modification and adaptation of the pre-processing and feature extraction process to the characteristics of the Arabic writing, but also that the HMM must be adopted to Arabic handwriting style, along with post-processing that uses language dependent syntax and semantics.

This paper gives a detailed description of the recognition system of Arabic handwritten words, developed at IFN in the last years. Section 2 provides an overview of the IFN/ENIT database as a basic tool for the development of recognition systems, section 3 describes the pre-processing, normalisation, and feature extraction in more detail, and section 4 presents the HMM used for recognition. Training and recognition processes are described, and in section 5, the results obtained with the IFN/ENIT database are presented. The paper ends with some concluding remarks.

2 Databases and Text Recognition

Most recognition systems in use today are developed for applications with a restricted lexicon of words. These systems are focused on certain applications such as the reading of cheque amounts or postal addresses, which are proven to be realistic and profitable. The further development of recognition systems, however, needs a large amount of data to train and test the system, especially if statistical methods like HMM are used. The implementation of a system requires real world data, but data from the bank or the postal system are often confidential and inaccessible for non-commercial research. As the amount of data is crucial for a reliable training of recognition systems, we decided to use similar artificial data collected on special forms instead of scarce real world data. Despite the disadvantage of using artificial data, the data labelling process is made simpler due to the fact that the forms can be adopted to the automatic labelling process. For a test environment, we chose the names of 946 Tunisian towns/villages together with their postcodes. 411 people, most of them selected from the narrower range of the Ecole Nationale d'Ingnieurs de Tunis (ENIT), contributed to the IFN/ENIT database. Each writer was asked to fill a form with pre-selected names of Tunisian towns/villages and the corresponding postcode. Town names and numbers were extracted automatically, and GT and baseline information were added automatically as well. Finally, GT and baseline information were verified manually. The version of IFN/ENIT database used consists of 26459 handwritten Tunisian town/village names, with these names made up of about 115000 pieces of Arabic words (PAWs) and about 212000 characters. Each handwritten town name comes with a binary image bitmap and additional Ground Truth (GT) information. Table 1 gives an example of a data set entry of the IFN/ENIT database.

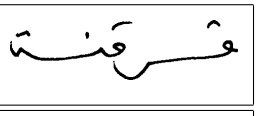
Image	
Ground truth:	
Postcode	3070
Global word	قرقنة
Character shape sequence	قB|رE|قB|نM|تE|
Baseline y1,y2	77,83
Baseline quality	B1
Quantity of words	1
Quantity of PAWs	2
Quantity of characters	5
Writing quality	W1

Table 1: A data set entry of the IFN/ENIT-database. The symbols M, B, A, E stand for the used character shapes (middle, begin, alone, end position in a word).

3 Pre-processing, Normalisation, and Feature Extraction

The first step in an offline word recognition system is the conversion of the paper document via scanning into the digital form.

3.1 Pre-processing

In the following steps, basic pre-processing tasks such as: noise filtering, text block segmentation, image binarisation, and word segmentation must be performed. These tasks are strongly dependent on the quality and nature of the documents used, but they are independent from the subsequent processing steps. Therefore, contained in the IFN/ENIT database are already pre-processed binary images of single words to make the work on recognition methods independent from the nature of special documents. These fundamental pre-processing tasks have already been done during the database development. Using the resulting binary image of a cropped town name from the database as a first step, a contour representation of the image and a noise reduction filtering are applied. The contour coding algorithm used is a fast sequential algorithm. Noise is simply and quickly reduced on the contour list by deleting contours that are too short to be part of a character. Finally, a vectorisation of the contour code is performed with a given approximation threshold. This results in a further data reduction and an additional contour noise reduction.

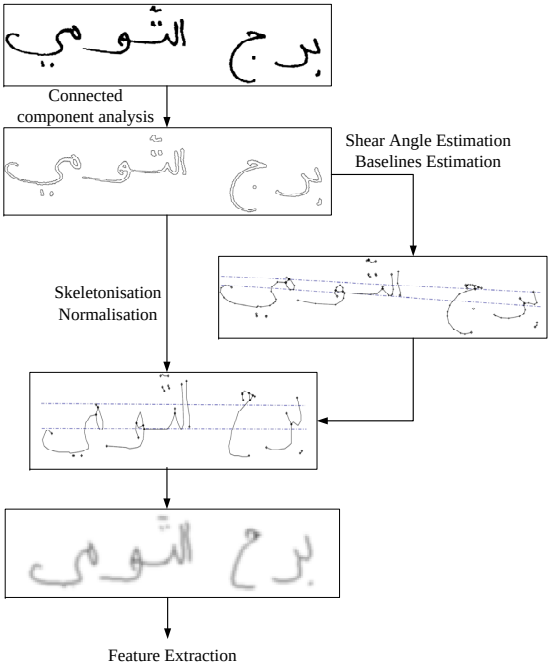


Figure 1: Example of the pre-processing and normalisation steps

3.2 Normalisation

For estimating normalisation parameters as well as for performing normalisation, a skeletonisation of the word image is used. The skeletonisation itself is also performed on the list of the contour representation resulting directly in a graph representation of the skeleton [FER 94] very similar to the contour representation. The advantages of this method are the following: The data structure is easily accessible, the pen-dependant line width is removed, and the connected components are easily obtained.

The task of the normalisation step is to reduce the writing style difference between writers to make the recognition process possible and more effective.

The most important features for normalisation are the baselines of a word, with a differentiation between the upper and the lower baseline (writing line). An example is given in figure 1. It is clear that many characters can not be recognised without the information about their relative vertical position within a word. This information is contained in the upper and lower baselines of a word, which varies strongly between different writers. If the baselines are detected, the normalisation of the skew angle and the height of the word can be found.

Often the horizontal projection method is used to detect the writing line. This method is robust and easy to implement but it needs straight lines and long words, which is often not the case for single handwritten words. Experiments have shown that a feature based method performs better. The implemented method is completely based on the processing of the polygonally approximated skeleton [PEC 02b]. The goals of this method are to calculate robust structural features and use these features for classifying the obtained connected components into writing-line relevant and irrelevant ones. When this is done, a regression analysis of the relevant points can estimate the final writing line position. An example is given in figure 2 and more details can be found in [PEC 02b]. The estimation of the upper baseline can not be done by the same method. Experiments have demonstrated that the estimation of the upper baseline is very inconsistent using projection or feature methods. The upper baseline, therefore, was instead estimated by a very simple method based on following assumptions: The upper baseline is parallel to the writing line and is located above this line in a fixed distance of 40% of the distance between writing line and the top of the word.

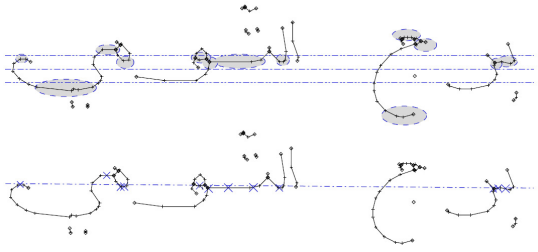


Figure 2: Baseline estimation

The length of a word can be normalized if the number of characters in a word is known. The number of characters is estimated from the number of black/white transitions along three lines parallel to the estimated baseline at predefined fixed positions.

Based on the estimated parameters all geometrical normalisation steps of a word can be performed, - again on the skeleton graph (see figure 1). First, a rotation is performed resulting in a horizontal baseline. Then a vertical height normalisation is done with a nonlinear characteristic within the ascender and descender region, which results in constant heights for ascender and descender regions and thus in a fixed total height of the resulting skeleton graph. Next, horizontal width normalisation with a linear characteristic is calculated, yielding a word with constant average character width. (The line thickness was already normalised during the generation of the skeleton to a thickness of one.) Finally a normalised image with re-thickened lines is made by Gaussian low-pass filtering of the image with normalised skeleton lines. This results in a normalised grey level word image. Figure 1. shows an example of a normalised word image.

3.3 Feature Extraction

The extraction of the features, which are used for the recognition process, is a difficult task with two aims: first, to identify which features are relevant, and secondly, to find all of them. After testing several different features we finally chose the grey valued pixels of the normalised word image as features.

3.3.1 Sliding Window

To collect the features of a word a rectangular window is shifted from right to left (in accordance with the Arabic writing direction) across the normalised grey level script image to generate a feature vector (frame) at each shift position. This results in a vast amount of features for each frame. Figure 3 gives an example of the sliding window feature extraction method. In this example the window consists of three columns, which are concatenated to build one feature vector.

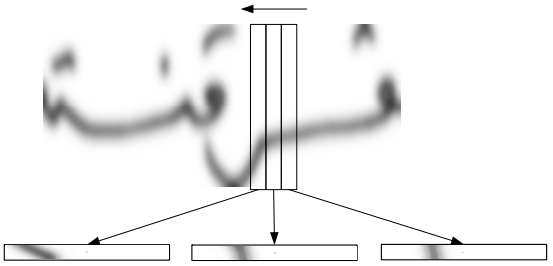


Figure 3: Feature extraction with sliding window

3.3.2 Karhunen-Loève Transform

In order to reduce the feature vector dimension, the Karhunen-Loève Transform (KLT) is applied. KLT is a standard statistical method to reduce a feature set to only the most relevant features.

The sequence of these transformed feature vectors are the input for the HMM recogniser.

4 HMM Recogniser

The problem of recognising a handwritten word as a whole can now be considered as a sequence of decisions in which feature vectors are grouped into smaller “decision units“ and sequentially recognised. The sequence of these “decision units“ represents the unknown word. To solve such a recognition problem, Hidden Markov Models are widely used, in the beginning to recognise speech and later to recognise cursive written words.

Details of HMM will not be discussed further in this paper. For more information about HMM, refer to e.g.

[RAB 90] and [HUA 90]. In the following section the special solutions used in this system will be discussed. The first step is to define the HMM model that will be employed.

4.1 General Definition

Hidden Markov Models can be described with the parameter set $\lambda = (A, B, \Pi)$:

- Matrix A: Transitions probabilities from one state to another, with $A = \{a_{ij}\}$ and $a_{ij} = p(X_t = j | X_{t-1} = i)$
- Matrix B: Distributes probabilities of observations, with $B = \{b_j(o)\}$
- Matrix Π: probabilities to reach a state from the initial state, with $\Pi = \{\pi_i\}$

The goal is to determin the probability of an unknown sequence of observations $P(o_1, ..., o_T|\lambda)$ and maximise the likelihood $\widehat{\omega_i} = arg\ max\ p(\underline{o}|\lambda_j)$.

4.2 Principal Structure

The following tasks must be completed to develop a cursive word recogniser:

1. Choose the states and the corresponding observations
2. Choose a topology of the states
3. Choose a strategy to segment the word into observations (manually - automatically)
4. Select training and testing data
5. Run the training of the HMM parameters
6. Test the system on the test data

In the following section, the solutions of tasks 1. - 6. will be briefly described as they were realised in our HMM recogniser.

4.3 Initialisation and Pre-Configuration of the Recogniser

4.3.1 Observations

Handwritten words are interpreted as a sequence of character shapes, which are concatenated to build the appearance of an individual handwritten word. Each character shape is interpreted as observation output of a state of the HMM. Especially in the case of Arabic handwriting, the character shapes differ from its position in a word. It follows then that the number of states is more than triple the number of characters in the Arabic alphabet.

4.3.2 HMM Topology

Many different model topologies were discussed using HMM systems. The simplest and most used topology is the left-right Bakis topology (figure4). Each state has three different paths, a recursive self transition, a transition to the next state, and a transition that skips the next state. For the topology shown in figure 4 we have the following transition matrix:

$$A = \begin{pmatrix} 0 & \pi_0 & \pi_1 & 0 & 0 & 0 & 0 \\ 0 & a_{00} & a_{01} & a_{02} & 0 & 0 & 0 \\ 0 & 0 & a_{11} & a_{12} & a_{13} & 0 & 0 \\ 0 & 0 & 0 & a_{22} & a_{23} & a_{24} & 0 \\ 0 & 0 & 0 & 0 & a_{33} & a_{34} & a_{3e} \\ 0 & 0 & 0 & 0 & 0 & a_{44} & a_{4e} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{ee} \end{pmatrix} \quad (1)$$

Using this simple model for the observation of a character we chose a model with 7 states for each character. This number, of course, is a parameter to optimise, which depends on the size and the quality of the data used.

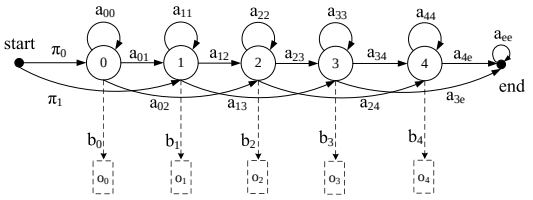


Figure 4: The Bakis model: A simple Left-Right-Model with start and end state. (The Bakis model allows transitions to the same state, the next and the one after the next)

4.3.3 Initialisation

Using character models requires training data that are segmented into character shapes. As this is a very time-consuming difficult and error-prone job, we implemented an algorithm to segment a word into characters automatically. The initialisation of a segmentation into n x 7 segments of a word with n characters is done by a Dynamic Programming clustering procedure. This procedure minimises an appropriate cost function, which ensures a maximum uniformity of feature vectors belonging to one state of a model. Recursively the minimisation of the mean square error of the feature vectors belonging to the same segment in respect to its mean value is done. Finally the initial segments are obtained by applying the back propagation method.

As second step the states of the multi modal distributions have to be initialised, which is called initialisation of the codebook estimation. This initialisation is done in two steps: the first step is the so called LBG-algorithm [GRA 84]. This algorithm uses the Euclidian distances only and in a second step the EM-algorithm is used to optimise the codebook initialisation [HUA 90].

These initialisation steps are the basis of the subsequent training of the HMM parameters.

4.3.4 Data

The selection of training and testing data is also a very important task. The data must be relevant to the task and sufficient to train all parameters of the HMM, and also - with another set - test the quality of the realised system. For this case the IFN/ENIT-database was used for training and testing our system. Each word in this database is labelled not only with the Arabic word but also with a string, which describes the sequence of character shapes of this word (see

table 1) this enables the automatic initialisation of the segmentation as described in 4.3.3. The words in the database are not equally distributed, but the words are chosen in a way that each character shape appears more than 30 times in the training data set. This ensures a minimum amount of data to make training on character shape level possible.

4.4 Training of the HMM

The training of the HMM-parameters is done by means of the Viterbi Algorithm using a segmental k-means Algorithm. The initial codebook is incorporated into the training procedure, that is, in each iteration only the state-vector assignment resulting from best path obtained from applying the Viterbi Algorithm is used to reestimate model parameters. As mentioned before the character shapes are modeled with HMM and for the recognition process concatenated to valid words of the lexicon used. Figure 5 shows an example of the concatenation of character models to build a word model. It can be seen that one character model is used twice in the word model. Figure 6a gives an example of the training, showing that each character shape of the same type, independent of the word where it was written, contributes to the statistical character shape model. This enables a statistical training with less training data than in the case of word based models.

4.5 Recognition

For recognition again basically a standard Viterbi Algorithm is used. The recognition process has to perform the task to assign to an unknown feature sequence a valid word from the lexicon. The basic way to do this is to calculate the probability that the observation was produced by a state sequence for each word of the lexicon. The sequence with the highest probability gives the correct word. As this procedure is too time consuming two actions were implemented:

1. A tree structured lexicon representing valid words is built. This leads to a significant search space reduction and unambiguous assignment of a word to each leaf.

2. A beam search strategy is used to reduce the search space. The number of hypotheses generated at each step is controlled by a constant score threshold relative to the currently best score and a maximum number of allowed hypotheses to be active at the same time.

Both procedures cause an acceleration of the recognition process but may also lead to a suboptimal solution only. If the parameters of these procedures are selected carefully a good result can be achieved.

5 Results

The recognition result we reached with the system presented in this paper is shown in table 2. This table also shows the results of the systems, which made a contribution to the first international competition on Arabic handwriting recognition systems at ICDAR 2005 [MäR 05]. The highest recognition rates are reached by both systems which are using HMM methods. Perhaps this fact is due to the robustness of the frame-based HMM strategy [MAK 98] additionally the

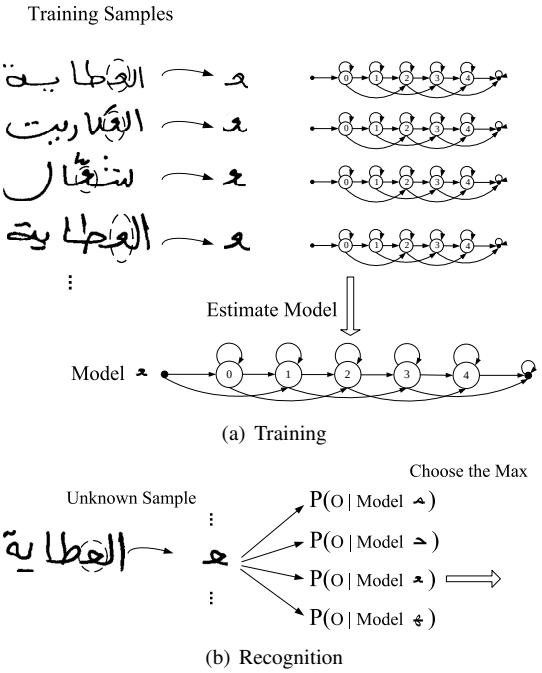


Figure 6: (a) a HMM is trained for each "mode" using a number of examples of that "mode" from the training set, and (b) to recognise some unknown sequence of "modes", the likelihood of each model generating that sequence is calculated and the most likely (maximum) model identifies the sequence.

baseline-dependent features give an improvement [ELH 05], which shows the importance of an efficient baseline detection algorithm.

System name	No.	1	1-5	1-10
ICRA	1	65.74	83.95	87.75
SHOCRAN	2	35.70	51.62	51.62
TH-OCR	3	29.62	43.96	50.14
UOB	4	75.93	87.99	90.88
REAM*	5	15.36	18.52	19.86
ARAB-IFN	6	74.69	87.07	89.77

Table 2: Recognition results in % with the new dataset e

6 Conclusion and Future Work

The results, which were achieved with our system using the frame-based HMM approach to recognise handwritten Arabic words are very promising. Nevertheless there is still a lot of work to do. To achieve an improvement a comparison of different methods and systems on the same data is one of the very important tasks. Therefore we are going to organise a second competition on Arabic handwritten word recognition systems at ICDAR 2007 and we are also continuing to develop further data sets.

The next steps in our work on Arabic handwriting recognition are the realisation of a system to recognise handwritten words without the explicit use of a lexicon and the integration of post-processing using linguistic methods into the system.

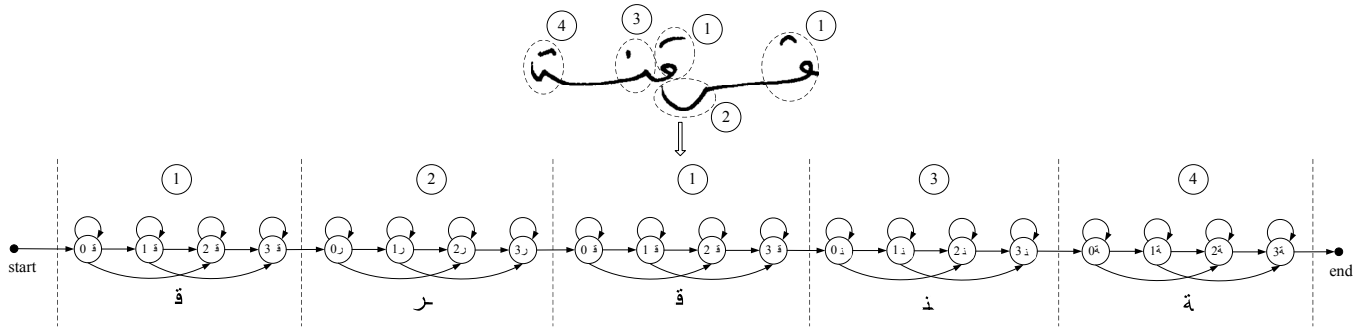


Figure 5: The HMM model for the word Kerkennah (قرقنة) with a duplication of the 1. character and 4 models each character.

References

[ABD 06] ABDULKADER A., A Two-Tier Approach for Arabic Offline Handwriting Recognition, *10th International Workshop on Frontiers in Handwriting Recognition (IWFHR'06)*, 2006.

[ALB 95] AL-BADR B., MOHMOND S. A., Survey and Bibliography of Arabic Optical Text Recognition, *Signal Processing*, vol. 41, 1995, pp. 49-77.

[ALM 02] ALMA'ADEED S., ELLIMAN D., , HIGGINS C., A Data Base for Arabic Handwritten Text Recognition Research, *Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 485-489.

[AMI 98] AMIN A., Off-line Arabic Character Recognition: The State of the Art, *Pattern Recognition*, vol. 31, n° 5, 1998, pp. 517-530.

[ELH 05] EL-HAJJ R., LIKFORMAN-SULEM L., , MOKBEL C., Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling., *ICDAR*, 2005, pp. 893-897.

[FER 94] FERREIRA A., UBEDA S., Ultra fast parallel countor tracking with application to thinning, *Pattern Recognition*, vol. 27, n° 7, 1994, pp. 867-878.

[GRA 84] GRAY R., Vector Quantization, *IEEE Acoustic Speech and Signal Proc. Magazine, April 1984*, vol. 1, n° 1, 1984, pp. 4-28.

[HUA 90] HUANG X. et al., *Hidden Markov Modells for Speech Recognition*, Edinburgh Universal Press, 1990.

[HUL 94] HULL J. J., A Database for Handwritten Text Recognition Research., *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, n° 5, 1994, pp. 550-554.

[JIA 05] JIANMING JIN HUA WANG X. D., PENG L., Printed Arabic document recognition system, *Proceedings of SPIE - Document Recognition and Retrieval XII*, vol. 5676, 2005, pp. 48-55.

[KHA 99] KHARMA N., AHMED M., , WARD R., A new comprehensive database of handwritten words, numbers and signatures used for OCR testing, *Proc of IEEE Canandian conference on electrical and computer engineering*, 1999, pp. 766-768.

[MAK 98] MAKHOUL J., SCHWARTZ R. M., LAPRE C., , BAZZI I., A Script-Independent Methodology For Optical Character Recognition., *Pattern Recognition*, vol. 31, n° 9, 1998, pp. 1285-1294.

[MäR 01a] MÄRGNER V., Automatic Generation of Databases for Arabic Text Recognition, *Proceedings of the Tunisian-German Conference on Smart Systems and Devices (SSD'01)*, 2001, pp. 292-297.

[MäR 01b] MÄRGNER V., Synthetic Data for Arabic OCR System Development, *Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, vol. 2, 2001, pp. 1159-1163.

[MäR 05] MÄRGNER V., PECHWITZ M., , EL-ABED H., ICDAR 2005 Arabic Handwriting Recognition Competition, *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, vol. 1, 2005, pp. 70-74.

[PEC 02a] PECHWITZ M., MADDOURI S. S., MÄRGNER V., ELLOUZE N., , AMIRI H., IFN/ENIT- Database of Handwritten Arabic Words, *Colloque International Francophone sur l'écrit et le Document (CIFED'02)*, 2002, pp. 127-136.

[PEC 02b] PECHWITZ M., MÄRGNER V., Baseline estimation for Arabic handwritten words, *Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 479-484.

[PEC 03] PECHWITZ M., MÄRGNER V., HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT- Database, *Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, vol. 2, 2003, pp. 890-894.

[RAB 90] RABINER L. R., A Tutorial on Hidden Markov Models and Selected Apllications in Speech Recognition, WAIBEL A., LEE K.-F., Eds., *Readings in Speech Recognition*, pp. 267-296, Kaufmann, San Mateo, CA, 1990.

[YOU 03] YOUSEF AL-OHALI A. M. C., SUEN C. Y., Databases for recognition of handwritten Arabic cheques., *Pattern Recognition*, vol. 36, n° 1, 2003, pp. 111-121.