

Multifont Arabic Characters Recognition Using Hough Transform and HMM/ANN Classification

Nadia Ben Amor

National Engineering School of Tunis, Tunisia
n.benamor@ttnet.tn , nadia.benamor@topnet.tn

Najoua Essoukri Ben Amara

National Engineering School of Sousse, Tunisia
Najoua.benamara@enim.rnu.tn

Abstract— Optical Characters Recognition (OCR) has been an active subject of research since the early days of computers. Despite the age of the subject, it remains one of the most challenging and exciting areas of research in computer science. In recent years it has grown into a mature discipline, producing a huge body of work.

Arabic character recognition has been one of the last major languages to receive attention. This is due, in part, to the cursive nature of the task since even printed Arabic characters are in cursive form.

This paper describes the performance of combining Hough transform and Hidden Markov Models in a multifont Arabic OCR system. Experimental tests have been carried out on a set of 85.000 samples of characters corresponding to 5 different fonts from the most commonly used in Arabic writing.

Some promising experimental results are reported.

Index Terms— Arabic Optical Character Recognition, Hough Transforms, Hidden Markov Models, Artificial Neural Networks.

I. INTRODUCTION

Arabic belongs to the group of Semitic alphabetical scripts in which mainly the consonants are represented in writing, while the markings of vowels (using diacritics) is optional.

This language is spoken by almost 250 million people and is the official language of 19 countries[1]. There are two main types of written Arabic: classical Arabic the language of the Quran and classical literature and modern standard Arabic the universal language of the Arabic speaking world which is understood by all Arabic speakers. Each Arabic speaking country or region also has its own variety of colloquial spoken Arabic.

Due to the cursive nature of the script, there are several characteristics that make recognition of Arabic distinct from the recognition of Latin scripts or Chinese.

The work we present in this paper belongs to the general field of Arabic documents recognition exploring the use of multiple sources of information. In fact, several experimentation carried out in our laboratory, had proved the importance of the cooperation of different types of information at different levels (features extraction, classification...) in order to overcome the variability of Arabic and especially multifont characters [2].

In spite of the different researches realised in the field of Arabic OCR (AOOCR), we are not yet able to evaluate objectively the reached performances since the tests had not been carried out on the same data base. Thus, the idea is to develop several single and hybrid approaches and to make tests on the same data base of multifont Arabic characters so that we can deduce the most suitable combination or method for Arabic Character Recognition.

In this paper, we present an Arabic Optical multifont Character Recognition system based on Hough transform for features selection and Hidden Markov Models for classification [3].

In the next section, the whole OCR system will be presented. The different tests carried out and obtained results so far are developed in the fourth section.

II. CHARACTERS RECOGNITION SYSTEM

In this section, we develop the main steps of the OCR system.

A. Preprocessing

Pre-processing covers all those functions carried out prior to feature extraction to produce a cleaned up version of the original image so that it can be used directly and efficiently by the feature extraction components of the OCR. In our case, the goal of image preprocessing is to generate simple line-drawing image such as the one in Figure 2 which presents the edges detection of the character 'noun'.

Our implementation uses the Canny edge detector [4] for this extraction.

While the extracted edges are generally good, they include many short, incorrect, (noise) edges as well as the correct boundaries. Noise edges are removed through a two-step process: first, connected components are extracted from the thresholded edge image, and then the smallest components, those with the fewest edge pixels, were eliminated. After noise removal, the resulting edges are quite clean.



Figure 1. Edges extraction using canny edge detector

B. Features extraction

Features extraction is one of the two basic steps of pattern recognition. We quote from Lippman [5]: “Features should contain information required to distinguish between classes, be insensitive to irrelevant variability in the input, and also be limited in number to permit efficient computation of discriminant functions and to limit the amount of training data required.”

In fact, this step involves measuring those features of the input character that are relevant to classification. After feature extraction, the character is represented by the set of extracted features.

There is an infinite number of potential features that one can extract from a finite 2D pattern. However, only those features that are of possible relevance to classification need to be considered. This entails that during the design stage, the expert is focused on those features, which, given a certain classification technique, will produce the most and efficient classification results.

Obviously, the extraction of suitable features helps the system reach the best recognition rate [6]. In a previous work, we have used wavelet transform in order to extract features and we have obtained very promising results [7,12]. In this paper, we present a Hough Transform based method for features extraction.

Hough Transform

The Hough Transform (HT) is known as the popular and powerful technique for finding multiple lines in a binary image, and has been used in various applications. Though the principle of the Hough Transform is rather simple and seems easy to use, we cannot bring out precise results without paying enough attention to the arrangement of the parameter space used in the HT.

The HT gathers evidence for the parameters of the equation that defines a shape, by mapping image points into the space defined by the parameters of the curve . After gathering evidence, shapes are extracted by finding local maxima in the parameter space (i.e., local peaks).

The HT is a robust technique capable of handling significant levels of noise and occlusion.

The Hough technique is particularly useful for computing a global description of a feature (where the number of solution classes need not be known *a priori*), given local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (e.g. coordinate point) indicates its contribution to a globally consistent solution .

Hough transform is used to identify features of a particular shape within a character image such as straight lines, curves and circles . When using the H T to detect straight lines, we rely on the fact that a line can be expressed in parametric format by the formula:

$r = x \cos \theta + y \sin \theta$, where r is the length of a normal from the origin to the line and θ is the orientation of r with respect to the x -axis.

To find all the lines within the character image we need to build up the Hough parameter space H . This is a two dimensional array that contains accumulator cells. These cells should be initialised with zero values and will be filled with line lengths for a particular θ and r . For our study the range of θ is usually from 0° to 180° although often we only need to consider a subset of these angles as we are usually only interested in lines that lie in particular direction.

Without using information from neighbouring pixels (which the Hough transform doesn't), each black pixel $p(x,y)$ in the input image can possibly lie on a line of any angle . For each black pixel $p(x,y)$ in the image, we take each angle along which we wish to find lines, calculate the value r as defined above and increment the value held in accumulator cell $H(r, \theta)$ by 1. The values in the resultant matrix will hold values that indicate the number of pixels that lie on a particular line $r = x \cos \theta + y \sin \theta$. These values don't represent actual lines within the source picture, merely a pixel count of points that lie upon a line of infinite length through the image.

Lines passing through more pixels will have higher values than those lines passing through fewer pixels. The line can be plotted by substituting values for either x and y or r and θ and calculating the corresponding co-ordinates.

• **Line Extraction**

To extract collinear point sets, one must first extract significant straight lines from the image. These lines correspond to major linear features. The advantage of the Hough transform [8] is the fact that it operates globally on the image rather than locally. The Hough Transform works by allowing each edge point in the image to vote for all lines that pass through the point, and then selecting the lines with the most votes. After all edge points are considered, the peaks in the parameter space indicate which lines are supported by the most points from the image.

The first thing to understand about parameter space for line extraction is that there is no one-to-one relationship between pixels in the image and cells in the parameter space matrix. Rather, each cell in parameter space represents a line that spans across the entire image.

The transformation between feature space and parameter space is the following:

- Project a line through each edge pixel at every possible angle (you can also increment the angles at steps).
- For each line, calculate the minimum distance between the line and the origin.
- Increment the appropriate parameter space accumulator by one.

The resulting matrix: The x-axis of parameter space ranges from 1 to the square root of the sum of the squares of rows and columns from feature space. This number corresponds to the furthest possible minimum distance from the origin to a line passing through the image. The y-axis represents the angle of the line. Obviously the axes could be switched.

The larger the numbers in any given cell of the accumulator matrix, the larger the likelihood that a line exists at that angle and distance from the origin.

C. Hidden Markov Models Classification

Hidden Markov models or HMM's are widely used in many fields where temporal (or spatial) dependencies are present in the data [9].

During the last decade, hidden Markov models (HMMs), which can be thought of as a generalization of dynamic programming techniques, have become a very interesting approach in pattern recognition.

The power of the HMM lies in the fact that the parameters that are used to model the signal can be well optimized, and this results in lower computational complexity in the decoding procedure as well as improved recognition accuracy. Furthermore, other knowledge sources can also be represented with the same structure, which is one of the important advantages of the hidden Markov modeling.

The HMM we retained uses a left-to-right topology, in which each state has a transition to itself and the next state. HMM for each character have 4 to 7 states, but we have noticed that 5 is approximately the optimal number of states.

The standard approach is to assume a simple probabilistic model of characters production whereby a specified character C produces an observation sequence O with probability $P(C;O)$. The goal is then to decode the character, based on the observation sequence, so that the decoded character has the maximum *a posteriori* probability.

Considering the choices of initial values of observation and transition matrixes, all models are identical at the beginning of the learning.

The number of states varies from 4 to 7 and it's worth mentioning that a 0 state has been added to make the computing easier.

Since there is no observation in this state, there will be no influence on the models.

Since the models are labeled by the identity of the characters they represent, the task of recognition is to identify, among a set of L models λ_k , $k=1, \dots, L$ those (the character) which gives the best interpretation of the observation sequence to be decoded i.e:

$$\text{Car} = \arg \max_{1 \leq \text{car} \leq L} [P(O | \lambda)]$$

D. Artificial Neural Networks Classification

Artificial Neural Networks classifiers (ANN) have been used extensively in character recognition [13, 14]. These networks can be used as a feature extractor, as a "pure" classifier when the inputs are features already extracted or combined with another classifier which is the case in our system.

Two models of neural network were tested: multilayer perceptrons (MLP) and radial basis function (RBF) networks but the best results were achieved with the MLP network.

The architecture we used contains three layers of neurons: One layer of neurons for the input, one intermediate layer, and finally an output layer.

E. Hybrid HMM/MLP Classification

A hybrid HMM/Neural Network architecture was applied; in this architecture, a three-layer neural network is used as a state emission probability estimator and the conventional forward-backward algorithm is applied for estimating continuous targets for the Neural Network training patterns. The final network is trained to estimate the probability of 28 categories of Arabic characters.

The hybrid system we developed uses multilayer perceptrons (MLPs) to classify input features. At recognition time, MLP outputs are estimates of the posterior probabilities.

This hybrid approach implies the creation of twenty eight networks corresponding each to the HMM observation of an Arabic character in its isolated form. Every network is characterized by:

- an input layer
- one hidden layer where the number of neurons were experimentally adjusted
- an output layer formed by a neuron corresponding to the considered character

III. EXPERIMENTAL RESULTS

The different tests have been carried out on isolated Arabic characters.

Due to the absence in Arabic OCR of a data base, we have created our own corpus which is formed by 85000 samples in five different fonts among the most commonly used in Arabic writing which are: Arabic transparent, Badr, Alhada, Diwani, Koufi.

The following figure presents the shapes of some characters in the five different fonts.

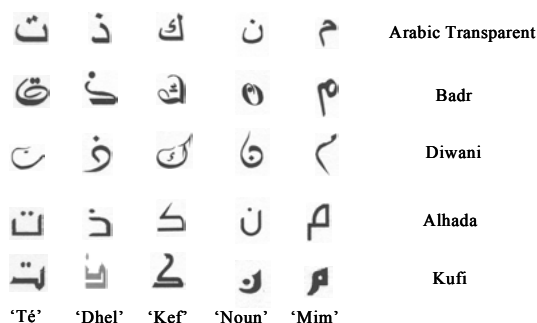


Figure2: Illustration of the five considered fonts

The achieved results for the HMM/MLP approach using Hough transform extracted features are shown in the following table. Overall recognition rate is of 97.36%

TABLE I. HOUGH TRANSFORM/HMM-MLP RECOGNITION RATE PER CHARACTER

Characters	Recognition rates
ا	97.64
ب	97.06
ت	98.29
ث	97.67
ج	92.39
ح	95.28
خ	94.67
د	95.08
ذ	96.87
ر	98.54
ز	97.39
س	98.76
ش	97.98
ص	96.23
ض	99.56
ط	97.39
ظ	93.56
ع	98.60
غ	98.54
ف	99.56
ق	95.78
ك	98.40
ل	97.65
م	96.79
ن	99.65
ه	99.81
و	97.39
ي	99.67
Average	97,36

IV. CONCLUSION

A wide variety of techniques are used to perform Arabic character recognition.

In this paper we presented a hybrid technique based on both neural networks and Hidden Markov Models for classification and we tested this hybrid approach with a method based on Hough transform for features extraction.

As results show, designing an appropriate set of features for the classifier is a vital part of the system and the achieved recognition rate is indebted to the selection of features especially when we deal with multifont characters.

We can notice also that thanks to the use of a hybrid approach at the classification level, the recognition rate has increased compared with the approach only base on HMM for the classification that we implemented in a previous work [11].

We are intending to carry out other hybrid approaches on the level of features extraction such as combining both features extracted from wavelet decomposition and Hough transform in order to take advantages of their both characteristics besides testing other hybrid classifiers such as the neuro-fuzzy one on which we are working and have already reached some good results [15].

REFERENCES

- [1] A. Amin. Arabic character recognition. In H. Bunke and P. Wang, editors, Handbook of Character Recognition and Document Image Analysis, pages 397–420. World Scientific Publishing Company, 1997.
- [2] N. Ben Amor, S. Gazeh, N. Essoukri Ben Amara: "Adaptation d'un système d'identification de fontes à la reconnaissance des caractères arabes multi-fontes" Quatrièmes Journées des Jeunes Chercheurs en Génie Électrique et Informatique, GEI'2004, Monastir, Tunisia , 2004.
- [3] N. Ben Amara, A. Belaïd and N. Ellouze: "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe : État de l'art" CIFED 2000
- [4] J.F Canny. "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-6, pp. 679-698, 1986.
- [5] R. Lippmann, "Pattern Classification using Neural Networks", IEEE Communications Magazine, p. 48, November 1989.
- [6] E. W. Brown, "Character Recognition by Feature Point Extraction", Northeastern University internal paper, 1992
- [7] N. Ben Amor , N. Essoukri Ben Amara: "Applying Neural Networks and Wavelet Transform to Multifont Arabic Character Recognition" International Conference on Computing, Communications and Control Technologies (CCCT 2004), Austin (Texas), USA, on August 14-17, 2004.
- [8] J. Illingworth and J. Kittler, "A Survey of the Hough Transform" Computer Vision, Graphics and Image Processing, vol. 44, pp. 87-116, 1988.
- [9] R.-D. Bippus and M. Lehning. "Cursive script recognition using Semi Continuous Hidden Markov Models in combination with simple features". In European workshop on handwriting analysis and recognition, Brussels, July 1994.

- [10] N. Ben Amor, N. Essoukri Ben Amara : “Hidden Markov Models and Wavelet Transform in Multifont Arabic Characters Recognition”, International Conference on Computing, Communications and Control Technologies (CCCT 2005), July 24-27, in Austin, Texas, USA (Silicon Hills) 2005.
- [11] N. Ben Amor, N.Essoukri Ben Amara “Multifont Arabic Character Recognition Using Hough Transform and Hidden Markov Models” ISPA2005 IEEE 4th International Symposium on Image and Signal Processing and Analysis September 15-17, 2005, Zagreb, Croatia.
- [12] N. Ben Amor, N. Essoukri Ben Amara :” Multifont Arabic Characters Recognition Using Hough Transform and Neural Networks” the Third International Symposium on Neural Networks (ISNN 2006) Chengdu China, May 28-31, 2006.
- [13] N. Ben Amor, N. Essoukri Ben Amara : “A hybrid approach for Multifont Arabic Characters Recognition”, 5th WSEAS Int.Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'06) Madrid, Spain, February 15-17, 2006.
- [14] M Altuwajri , M.A Bayoumi , "Arabic Text Recognition Using Neural Network" ISCAS 94. IEEE International Symposium on Circuits and systems, Volume 6, 30 May-2 June 1994.
- [15] N. Ben Amor, M. Zarai and N. Essoukri Ben Amara :” Neuro-Fuzzy approach in the recognition of Arabic Characters” IEEE - 2nd International Conference on Information & Communication Technologies: From Theory to Applications (ICTTA'06) Damascus-Syria (April 24-28, 2006).

Nadia Ben Amor is born in Sousse, Tunisia on January, 11th 1977. Her educational background is as follow:

June 2000: National Engineering School of Monastir (Tunisia): National Engineering Diploma in Electrical Engineering

June 2002: Faculty of Sciences of Monastir (Tunisia): Diploma of Intensive Studies (Master's Degree) in Electronics

Since December 2003: National Engineering School of Tunis (Tunisia): PhD Candidate (Laboratory of Systems and Signal Processing (LSTS))

She is recruited as a Chief Engineer Since February, 19th 2001 in the “National Society of Telecommunications, Tunisia”.