IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 2, No. 1, March 2013, pp. 20~26
ISSN: 2252-8938

Segmentation and Recognition of Arabic Printed Script

S. NOURI, M. FAKIR
Information processing and telecommunications Team, Faculty of Science and Technology,
PB 523, Beni Mellal, Morocco.

Article Info

Article history:

Received Aug 05, 2012
Revised Oct 27, 2012
Accepted Jan 03, 2013

Keyword:

Segmentation
Arabic characters
Preprocessing
Recognition
Density profile
Hu moments
Histogram

ABSTRACT

In this work we present a method for the recognition of Arabic printed script. The major problem of the automatic reading of cursive writing is a segmentation of script to isolate characters. The recognition process consists of four phases: Preprocessing, segmentation, feature extraction and the recognition.

In the preprocessing, the image is scanned and smoothed. The correction of skew lines is done by using *Hough* transform. In the second phase, the text is segmented into lines, words or parts of words and each word into characters based on the principle of projection of the histogram. Features such as: density, profile, Hu moments and histogram are used to classifier the characters.

Corresponding Author:

M. FAKIR,
Departement of Computer Sciences,
Faculty of Sciences and Technics, University Sultan Moulay Slimane, Morocco
Email: fakfad@yahoo.fr

1. INTRODUCTION

Segmentation and recognition of character Arabic printed are two research important topics since the 70th [1]. The subject is a part of the future of human-machine communication. This recognition system is used in several domains where the text is the base of work.

The recognition systems of Arabic character consist of four steps: preprocessing, segmentation feature extraction and classification. After the acquisition of the image, thresholding is applied on the acquired image. Then the step of segmentation begins by segmenting text lines using the horizontal histogram then each line is segmented into words or parts of word using the vertical projection of histogram, after that we remove the base line for each word then segmenting each word into character using the vertical histogram projection. Overlapping is the problem directly affects the quality of this step. In the end the data is

2. CHARACTERISTICS OF ARABIC CHARACTERS

The Arabic script is written from right to left. It is cursive, that is to say the letters are generally related to each other. Each character can take four different forms, depending on its position in the word (Table 1). A set of black pixels adjacent to each other is called a connected component. The latter, in Arabic script, does not necessary represent an entire word; it may be only part of word, as some characters should not be attached to their successor left in the word. Moreover, there are different letters have the same shape but differ in the position and number of points that belong to them. Vowels are not used systematically in the Arabic script; signs that match the vowels are used to avoid errors in pronunciation. There are two types of texts: texts with or without the vowel signs. Some Arabic texts (Qur'an and books of learning reading and writing for children) include vowel signs. The other is to say, books, journals; publications are texts without these signs.

3. PREPROCESSING

The text is scanned with a suitable resolution and is stored as a binary image. Before being analyzed image undergoes some preprocessing, including rehabilitation, smoothing and normalization.

The Preprocessing is carried out in order to improve the quality of image to be processed. The acquisition devices such as scanners and cameraman distort the image of the text. Some faults can occur, for example skew of the writing because of bad positioning of the sheet in the scanner, or black spots caused by dirt or dust.

To prevent these defects, two preprocessing are considered: a smoothing followed by a recovery. A third is planned to process entries of the same size is standardization.

3.1. Smoothing

It consists in two elementary operations. Cleaning which is to detect all the tasks that are not part of the text to process and eliminate it from the image of the text. The second operation is the capping which is internal to plug the holes in the shape of the character and to equalize the rounds of writing. These two operations are very delicate in the sense that some tasks representative's diacritics points can be confused with noise, holes intra characters can be confused with the character of deformation and consequently blocked by mistake.

Table 1. Arabic printed characters in four forms

Name	Isolated	First	Middle	Last
Alif	ا	ا	ا	ا
Baa	ب	ب	ب	ب
Taa	ت	ت	ت	ت
Thaa	ث	ث	ث	ث
Geem	ج	ج	ج	ج
Hha	ح	ح	ح	ح
Kha	خ	خ	خ	خ
Dal	د	د	د	د
Thal	ذ	ذ	ذ	ذ
Raa	ر	ر	ر	ر
Zain	ز	ز	ز	ز
Seen	س	س	س	س
Sheen	ش	ش	ش	ش
Saad	ص	ص	ص	ص
Dhad	ض	ض	ض	ض
Tta	ط	ط	ط	ط
Zha	ظ	ظ	ظ	ظ
Ain	ع	ع	ع	ع
Ghain	غ	غ	غ	غ
Faa	ف	ف	ف	ف
Gaf	ق	ق	ق	ق
Kaf	ك	ك	ك	ك
Lam	ل	ل	ل	ل
Meem	م	م	م	م
Noone	ن	ن	ن	ن
Haa	ه	ه	ه	ه
Waw	و	و	و	و
Yaa	ي	ي	ي	ي

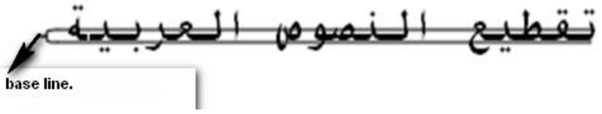

Figure 1. Example of writing showing the base line.

3.2. Text Line Skew Correction

It consists rectifying a skew of writing caused by deformation of the writing on the acquisition or processing of skew writing. The usual procedure is to detect the angle of recovery of skew, and to correct it in a rotate isometric - an angle equal to the value of the angle of skew found. In our case we use the method of Hough Transform [11].

3.3. Normalization

Normalization Consist to represent all characters to be process in a matrix of pixels of the same size (specifies the maximum height of characters), the number of columns is variable depending on the type of character.

4. SEGMENTATION

4.1. Ligne Segmentation

Lines of text are extracted using the histogram horizontal projection (Figure 2). A space between two dense parts in black pixels, corresponds to spacing. To solve the problem of false text lines we set the value of the space between two lines with at least two pixels. A space of less than two pixels is considered false text line and is directly connected to the next line where spacing may correspond to an area of a pixel.

To remedy this problem, some authors such as [12] first identify the various lines of writing, then group the text blocks based on their proximity to the lines of writing already localized.

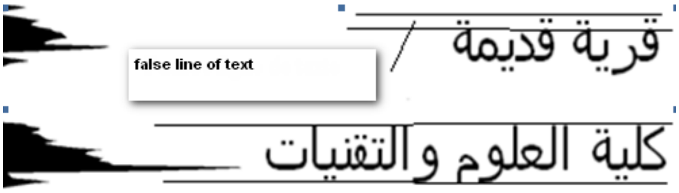

Figure 2. Example of horizontal histograms and a false line of text that results.

4.2. Thickness of the Script

The thickness of writing is calculated by looking at a line of text, column by column and calculating each time the number of black pixels in the column. The width of the writing is the number of pixels found most frequently. It will correspond subsequently to the width of the base line.

4.3. Base Line Detection

The base line is detected using the histogram of horizontal projections. The baseline represents the maximum amplitude of histogram and has width (Figure 4).

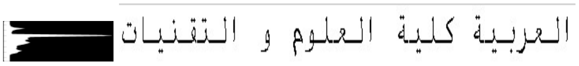

Figure 4. Sample text and associated horizontal histogram.

4.4. Segmentation of Lines into Words

This type of segmentation involves analyzing the current line of text for segmenting into part of the word or whole word (Figure 5). To achieve this segmentation, the line of text is examined from top to bottom for each line of pixels to determine whether a pair of lines, the black pixels is connected (Figure 5). They are differentiated by their size and their locations relative to the base line (above or below the base line). They are associated with the body of the word before another processing [13].

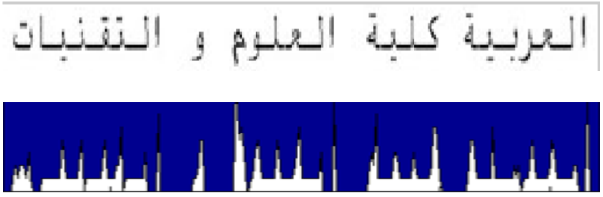

Figure 5. Example of vertical histograms

4.5. Words Segmentation into Characters

The words or parts of words found are examined one at a time. Each word is scanned from top to bottom to detect point diacritics and their location in the pixel array, only to have the body of the word is considered without points diacritics, then removing the base line of word (Figure 5), then a vertical histogram is established on the new image. The white space found in the histogram contains the preliminary segmentation points (the segmentation point generally corresponds to the end of the blank).

Once this is done, the segments obtained may correspond to complete or fragments of characters. Errors generated by the segmentation can be corrected in the classification phase, or be corrected during the segmentation by considering a processing step. In our case we are considering a processing step to correct segmentation errors.

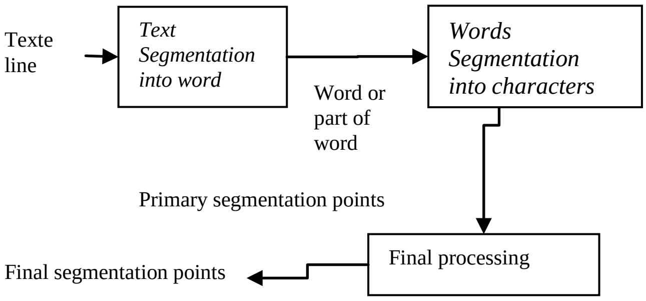
Texte line
Text Segmentation into word
Word or part of word
Words Segmentation into characters
Primary segmentation points
Final processing
Final segmentation points

Figure 6. Segmentation algorithm.

4.6. Processing after Segmentation

Errors that occur most frequently during the segmentation are detected, they are generally of over-segmentation, ie a character is cutting into two or more segments.

This processing attempts to correct these errors, each according to his case. The words are searched from left to right. Before start correction, the diacritics are placed with the characters, each in its place. [14].

5. FEATURES EXTRACTION

The choice of a vector of attributes to characterize an object can be tricky. Indeed, we must make a compromise between the size of the vector and content information. A vector of small size can give a poor performance of the classifier [15]. To characterize the character, we chose the following primitives: zoning, profiles and projection histograms that do not meet the fourth requirement (Independence of rotation), the geometric moments of Hu [16]. Zones are extract by zoning densities, we chose $n = m = 3$ (9 zones, the character is divided into three horizontal zones and each zone is divided into three vertical zones). Densities in each zone must be normalized by dividing the surface area. We consider profiles left and right features to obtain a vector. A location near the images was obtained of different sizes. The number of attributes is different from a character to another. To overcome this problem we have normalized the image to get the same size for all characters.

6. CLASSIFICATION

The classification phase takes two phases: The extraction phase and learning / recognition phase, since we opted for neural network as a classification method.

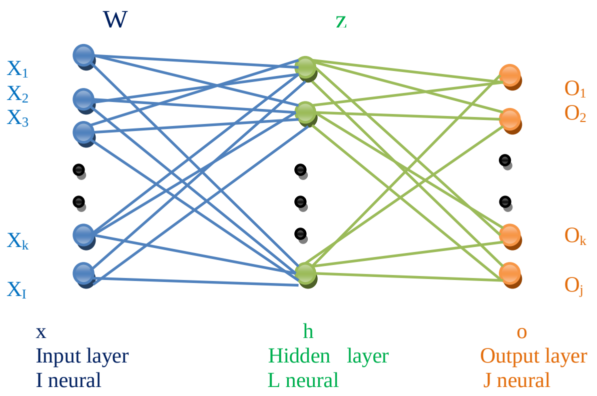

Figure 7. Neural network architecture.

Learning rule:

For a multilayer perceptron using the gradient method for minimization of errors :

- The correction of errors for the output layer:

$$z(t+1) = z(t) + \Delta_z(E)$$
$$= z(t) - \alpha \nabla_z(\mathrm{E})$$
$$= z(t) - \alpha \frac{\partial Ek}{\partial z}$$
$$= z(t) - \alpha * \frac{\partial Ek}{\partial ok} * \frac{\partial ok}{\partial zhk} * \frac{\partial zhk}{\partial hk}$$
$$z(t+1) = z(t) + \alpha * (\delta cachée, K) * hk$$

with

$$\delta cachée, k = (tk - ok) * ok * (1 - ok)$$
$$\mathrm{E_k} = \frac{1}{2}(\mathrm{tk} - \mathrm{ok})2$$

- Error correction for the hidden layer:

$$w(t+1) = w(t) + \Delta_w(E)$$
$$= w(t) - \alpha \nabla_w(\mathrm{E})$$
$$= w(t) - \alpha \frac{\partial Ek}{\partial W}$$
$$= w(t) - \alpha * \frac{\partial Ek}{\partial ok} * \frac{\partial ok}{\partial zhk} * \frac{\partial zhk}{\partial hk} * \frac{\partial hk}{\partial wxk} * \frac{\partial Wxk}{\partial W}$$
$$w(t+1) = w(t) + \alpha * (\delta cachée, K) * xk$$
$$\delta cachée, k = (\delta sortie, k) * Z * hk * (1 - hk)$$

7. EXPERIMENTALS RESULTS

In our application, based on the programming language Matlab, we used the images of ten printed Arabic characters with different positions as possible i.e. we worked on 32 characters, with ten different fonts for each, So we have a total of 320 characters. We used five images for each character for learning and the other five for testing.

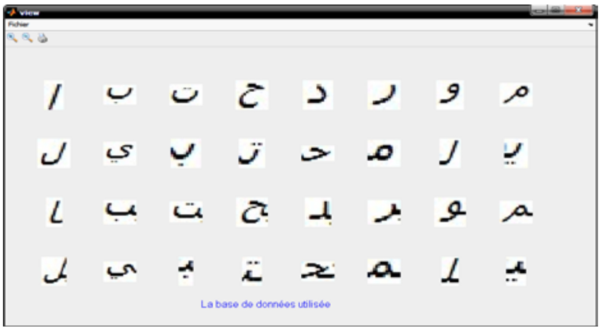

Figure 8. Sample of the database used

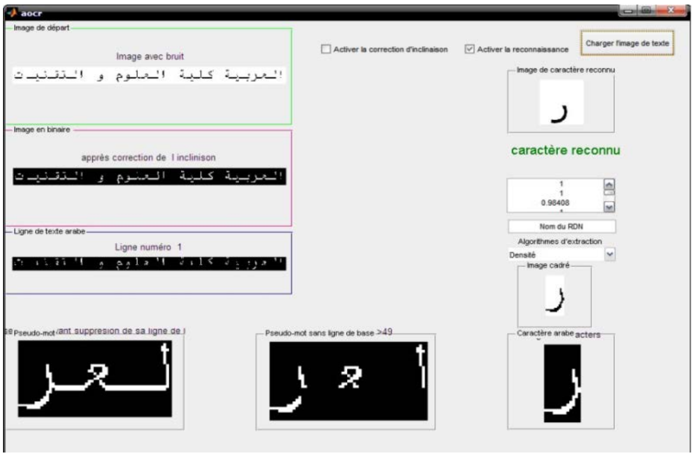
aocr
Image de départ
Image avec bruit
Image en binaire
apprès correction de l inclinison
Ligne de texte arabe
Ligne numéro 1
☐ Activer la correction d'inclinaison
☑ Activer la reconnaissance
Charger l'image de texte
Image de caractère reconnu
caractère reconnu
1
1
0.98408
Nom du RDN
Algorithmes d'extraction
Densité
Image cadré
Pseudo-mot avant suppresion de sa ligne de base
Pseudo-mot sans ligne de base >49
Caractère arabe

Figure 9. Interface for the first test of the segmentation algorithm

- Line and word segmentation: Our system identifies individual text lines exceeds 90% against the level of segmentation lines in word exceeds 80% because of overlapping horizontal or vertical of parts of word.
- Character recognition: The recognition rate obtained over 86% is moderately satisfactory, but as in any work, improvements are necessary to be made, making a partial list of later improvements in this work.

The results obtained are due of extraction algorithms used and methods of learning adopted

8. CONCLUSION

This study presented a method for recognition of Arabic characters using neural networks and features such as Moments of HU, density, histogram, and Profile. The reasons for using RDN consist of its computing time and efficiency. Therefore, the RDN is suitable for the recognition of Arabic characters because of its high performance. No effective techniques have been found for the recognition of Arabic script. This failure will be resolved in future study.

REFERENCES

[1] M Eden and M Hall. "*The characterization of cursives writing*". proc.4th symp.Informatics Theory, London. 1961: 287-299.

[2] N Benamara. "Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée". Thèse de doctorat, spécialité Génie Electrique, Université des sciences, des Techniques et de médecine de Tunis II. 1999.

[3] B Parhami, M Taraghi. "Automatic récognition of printed farsi texts". *Pattern récognition*. 1981; 14(l): 1-6.

[4] Amin, JF Mari. "Machine récognition and correction of printed Arabie text". *IEEE Transaction on System, man, cybernetics*. 1989 ; 19(5): 1300-1304.

[5] A Belaid. “Analyse de documents: de l'image à la représentation par les normes de codage”. *Cours de l'INRIA*. 1997: 32.

[6] P Burrow. “*Arabic handwriting recognition*”. Master of science thesis. School of Informatics, university of Edinburg, England. 2004.

[7] Belaïd et G Saon. Utilisation des processus markoviens en reconnaissance de l'écriture. *Revue Traitement du Signal*. 1997 ; 14(2): 161-177.

[8] IR Tsang. “Pattern recognition and complex systems”. *Thèse de doctorat, université d'Anterwerpen*. 2000.

[9] M Fakir, C Sodeyama. “Machine récognition of Arabie printed scripts by dynamic programming matching”. *Transaction on informatics Systems*. 1993; 76 (2): 235-242.

[10] H Emptoz, F Lebourgeois, V Eglin, Y Leydier. La reconnaissance dans les images numérisées. *OCR et transcription, reconnaissance des structures fonctionnelles et des méta-données*. 2003.

[11] C Olivier, H Miled, K Romeo-Pakker, Y Lecourtier. “Segmentation and coding of Arabie handwritten words”. *IEEE Proc. 13111 international conférence on pattern récognition* (ICPR'96), Vienne, Autriche, 1996. modeling and graphies (GMAG'03). 2003 : 264-268.

[12] D Motawa, A Amin, R Sabourin. “*Segmentation of Arabie cursive script*”. IEEE Proc. 461 international conférence on document analysis and récognition (*ICDAR'97*), Ulm, Germany. 1997 : 625-628.

[13] B Couasnon. “Segmentation et reconnaissance de documents guidées par la connaissance a priori: application aux partitions musicales”. *Thèse de doctorat de l'université de Rennes I*, France. 1996.

[14] L Souici, Z Zmirli, M Sellami. “Système connexionniste pour la reconnaissance de l'arabe manuscrit”. 1ères journées scientifiques et techniques *(JST FRANCIL)*, Avignon, France. 1997: 383-388.

[15] M Altuwaijri, M Bayoumi. “*A new thinning algorithm for Arabie characters using self-organizing neural network*”. Proc. IEEE. 1995: 1824-1827.

[16] MK Hu. “Visuel pattern recognition by moment invariants”. *IRE Transactions On Information Theory*. 1962.