# *Recognition of Printed Arabic Character Using Gray-Scale Matrices*

**Hanaa F.M. Mohammad**
*Computer Science Department*
*College of Education/Mosul University*

Received 2006/9/4 Accepted 2006/6/1

## الخلاصة

البحث يقدم طريقة لتمييز الحروف العربية المطبوعة . الحروف العربية تعتبر من أكثر الحروف استخداما في العالم وتطوير نظام تمييز حروف عربية صعب بسب طبيعة الحروف العربية المتميزة وكذلك التشابه في هيكل بعض الحروف .

التقنية المستخدمة في البحث تقسم الى ثلاث خطوات رئيسية :الخطوة الأولى هي تقسيم الوثيقة الصورية الى مقاطع صورية اصغر (صور الحروف) ثم عملية اكتشاف انحراف صورة الحرف وتصحيح الانحراف ان وجد. الخطوة الثانية هي عملية استخلاص الخواص باستخدام مصفوفات التردد اللوني Gray-Scale Matrices في الخطوة الثالثة تم استخدام تقنية K-Nearest-Neighbors لعملية التمييز. تم اختبار الطريقة على 45 نموذج لكل حرف . حيث تم تقسيم عينة كل حرف الى 20 نموذج تدريب و25 نموذج اختبار، النماذج المستخدمة في التدريب لا تظهر في الاختبار. عموما يتراوح المعدل العام للتميز بنسبة 90.3 %. وهذا يعتبر أداء جيد جدا. كل ما سبق يظهر بوضوح قدرة هذه الطريقة على تمييز الحروف العربية المطبوعة بصورة كفؤة.

## Abstract

An Optical Character Recognition (OCR) approach for printed Arabic script is presented in this paper, Which is one of the most popular scripts in the world. Development of an OCR system. For Arabic script it is difficult because Arabic characters are distinct and many structurally similar characters exist in the character set.

In the proposed approach, the technique can be divided into three major steps. The first step is digitization then do some pre-processing like segmentation to detect the slant of character and correct it .Second, feature extraction ,using gray-level matrices. Finally, the K-Nearest-Neighbors is used for classification. This method was tested using 45

patterns for each Arabic character with different fonts (simplified Arabic, tahoma, traditional Arabic). The sample images were divided into 20 training and 25 test images. Images in the test set did not appear in the training sets. This method performs extremely well with recognition rates 90.3%. This is a very good performance. All of this demonstrates that the new method is able to handle printed Arabic character task efficiently. It is a promising technique for recognition printed Arabic character.

## 1. Introduction

Optical character recognition (OCR), deals with the recognition of optically processed character rather magnetically processed ones. In a typical OCR system, input characters are read and digitized by an optical scanner. Each character is then located, segmented and the resulting matrix is fed into a preprocessor. Off-line recognition can de considered the most general case: no special device is required for writing and signal interpretation is independent of signal generation, as in human recognition[6].

The recognition of Arabic character has been an area of great interest for many years, and a number of research papers and reports have already been published in this area. There are several major problems with Arabic character recognition: Arabic characters are distinct and ideographic, many structurally similar character exist in the character set Table (1). Thus, classification criteria are difficult to generate [1][3][6].

The Arabic language has a rich vocabulary. More than 200 million people speak this language as their native speaking, and over 1 billion people use its character set, such as Persian and Urdu. Due to the cursive nature of the script, there are several characteristic that make recognition of Arabic distinct from the recognition of Latin script or Chinese [11][12].

The study of Arabic character recognition has been regarded since 1980s. However, in comparison with the other languages, such as Latin, Chinese and Japanese, there is a little work has been conducted on the automatic recognition of Arabic character [4][5].

Arabic is written from right to left. Since the proposed application area provide letters in an isolated form. Arabic has four forms for each letter depending on the position of the letter in each word. These are initial, medial, final and isolated, see Table (1).As more generalized system would need to train 112(28*4) separate classes rather than 15 classes (for letters) to accommodate all four forms.

**Table (1) The basic alphabets of Arabic character and their shapes at different positions in the word.**

| Name | Isolated | Started | Middle | End |
|---|---|---|---|---|
| Alif | ا | ا | ـا | ـا |
| Ba | ب | بـ | ـبـ | ـب |
| Ta | ت | تـ | ـتـ | ـت |
| Tha | ث | ثـ | ـثـ | ـث |
| Jeem | ج | جـ | ـجـ | ـج |
| Hha | ح | حـ | ـحـ | ـح |
| Kha | خ | خـ | ـخـ | ـخ |
| Dal | د | | | ـد |
| Thal | ذ | | | ـذ |
| Ra | ر | | | ـر |
| Zay | ز | | | ـز |
| Seen | س | سـ | ـسـ | ـس |
| Sheen | ش | شـ | ـشـ | ـش |
| Sad | ص | صـ | ـصـ | ـص |
| Dhad | ض | ضـ | ـضـ | ـض |
| Tta | ط | طـ | ـطـ | ـط |
| Za | ظ | ظـ | ـظـ | ـظ |
| Ain | ع | عـ | ـعـ | ـع |
| Gain | غ | غـ | ـغـ | ـغ |
| Fa | ف | فـ | ـفـ | ـف |
| Qaf | ق | قـ | ـقـ | ـق |
| Kaf | ك | كـ | ـكـ | ـك |
| Lam | ل | لـ | ـلـ | ـل |
| Meem | م | مـ | ـمـ | ـم |
| Noon | ن | نـ | ـنـ | ـن |
| Ha | ه | هـ | ـهـ | ـه |
| Waow | و | | | ـو |
| ya | ي | يـ | ـيـ | ـي |

## 2. Digitization

The first step in character recognition system is acquisition which is done by scanning an Arabic text with (300 dpi (dot per inch)) resolution. The input file for the system is a gray scale ((PGM) format (portable gray map)). Gray scale image files cover more pixels of the original image than binary images, so the file provides more information

about it. Also some feature like loops can be distorted in binary version of an image file, therefore unlike a gray scale file, useful information is corrupted[5].

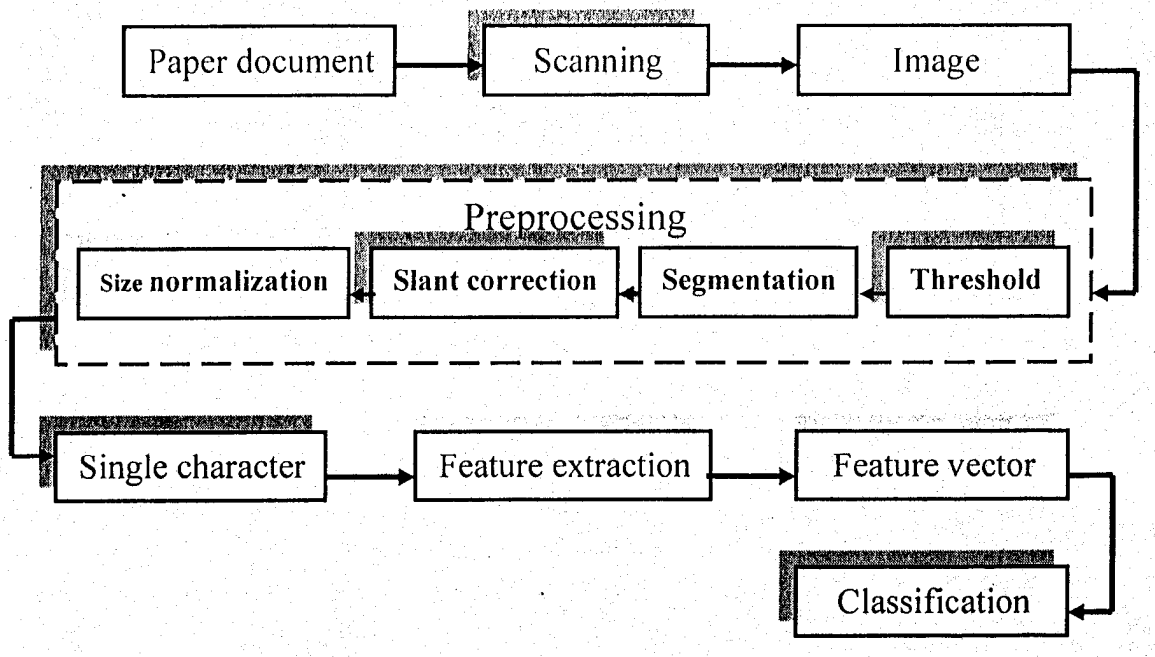
## 3. Preprocessing

It is necessary to perform several document analysis operations prior to recognizing text in scanned documents. In any OCR system. preprocessing includes the connection of segmentation and normalization, see figure (1), preprocessing receive a first binary image of a plurality of characters, Preprocessing generally consists of a series of image-image transformation. It dose not increase our knowledge of the contents of the document, but may help to extract it

### 3.1 Thresholding

The next step is to extract a binary (0,1) image from the obtained digital image. Image thresholding classifies the pixels of an image into the foreground (the writing) and the background. In the case of grayscale images, like images used in this research, the pixels initially have a value from 0-255, and the general idea of thresholding is to convert pixels above a certain level of gray into foreground and to convert pixels below the level into background. In the simplest method, global thresholding, a pre-determined constant T is used to threshold the image. Clearly, this

technique does not consider the difference between characters in a given image or between various images; contrast and brightness in an image can vary making this method too simplistic.



**Figure(1) : Steps in character recognition system.**

Depending on image quality, the background may be very dark or light, the writing may be fuzzy and light or dark and clear. For this reason, a dynamic process must be used to threshold the image. For example, another method is to use the pixels in the corner of the image which are assumed to be background and create a threshold value based upon these pixels. However, while this method does pay attention to the shade of the background, it still ignores the foreground and can lead to poor and spotty results.

A common method practiced is to use a histogram of the pixel values in the image; there should be a large peak indicating the general value of the background pixels and another, smaller peak indicating the value of the foreground pixels. A threshold can then be chosen in between the two peaks; this is known as valley-seeking [17]. This is a successful method. However images do not always contain well-differentiated foreground and background intensities due to poor contrast and noise. Perfect thresholding is a difficult task, and this method provides adequate results so a variation of this method has been implemented for use in this application.

## 3.2 Segmentation

The next step in the process of recognition is to break up the binary image into the images of each individual character. Then each character will be represented by a 2 dimensional bitmap array. This is one of the most difficult pieces of the OCR system. In all printed Arabic character, the width at a connection point is much less than the width of the beginning character. This property is essential in applying the baseline segmentation technique [6]. The baseline is a medium line in the Arabic word in which all the connection between the successive character tack place. If a vertical projection of bi-level pixel is performed on the word [6] (Eq. 1).

$$p(j) = \sum_{i} w(i, j) \qquad (1)$$

Where $w(i,j)$ is either zero or one and $i,j$ index the rows and columns, respectively, the peak point will have a sum greater than the average value ($AV$) (Eq. 2).

$$AV = (1 / Nc) \sum_{i=1}^{Nc} Xj \qquad (2)$$

And where $Nc$ is the number of columns and $Xj$ is the number of black pixels of the $j^{th}$ column [2]. Figure(2), illustrates the segmentation of an Arabic word into peaks

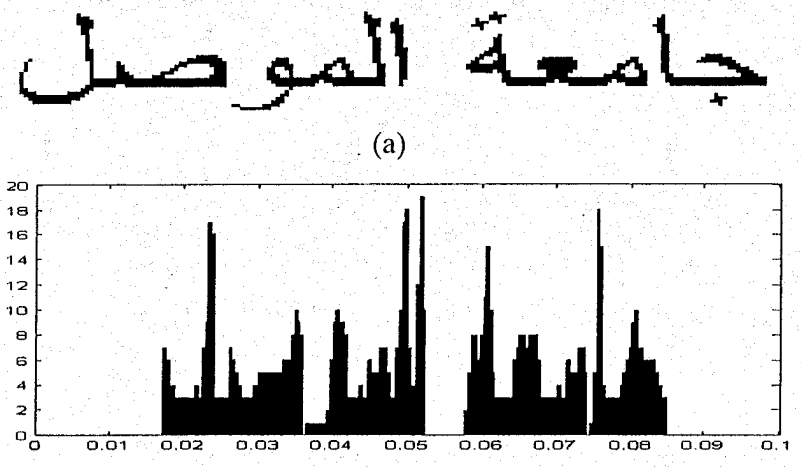



**Figure (2): An example of segmentation of the word into peaks (a) Arabic word (b) histogram.**

## 3.3 Slant Correction

Slant correction is intended to straighten a character so that its main vertical components stand perpendicular to its baseline and its main horizontal components lie parallel to its baseline. This method is employed because the recognition methods used can be sensitive to the pitch of a character and may fail if the image is tilted despite having a clear form. To accomplish this task, The horizontal/vertical projection profile is used.

The horizontal/vertical projection profile is a histogram of the number of black pixels along horizontal/vertical scan-lines. For a script with horizontal text lines, the horizontal projection profile will have peaks at text line positions and troughs at positions in between successive text lines. To determine the slant angle of a document, the projection profile is computed at a number of angles, and for each angle, the difference between peak and trough heights is measured. The maximum difference corresponds to the best alignment with the text line direction. This in turn determines the slant angle [7][14][16].

**Figure (3): (a) The character before slant correction (b) The character after slant correction.**

## 3.4 Size Normalization

In OCR, very small and very large word or character images are often scaled to standard size, even though the outlines of characters of different size in the same typeface are not congruent. Size normalization is used to reduce the variation in size. Directly scaling all images to an identical size will result in significant deformation in many case.

Size normalization for binary image $f(x,y)$ applied in this OCR, so that the size of the rectangle circumscribing the pattern is 32 x 32 pixel. Consequently, the normalization image $f'(x,y)$ is described as follow :

$$f'(x,y) = f(((width * x)/32) + \delta x, ((height * y)/32) + \delta y)) \quad (3)$$

Where width and height are that of the pattern, respectively. Then $\delta x$ and $\delta y$ are the horizontal and vertical distance between the left-top corners of the image and the rectangle, respectively.

## 4. Features Extraction

In principle, any texture analysis technique can be applied to extract features from each character image. Here established methods is implemented to obtain texture features, namely the gray-level matrix. It is often used as a benchmark in texture analysis [9].

## 4.1 A Gray Scale Matrices

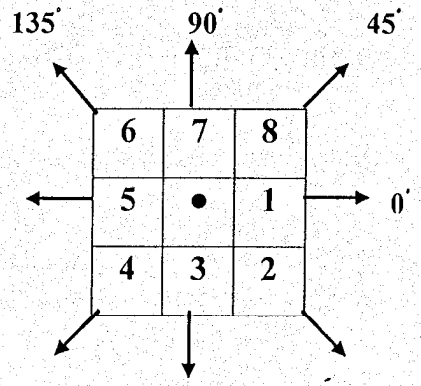
The gray Scale matrices technique sketched in this section is based on the repeated occurrence of some grey level configuration in the texture.

Let $f: L_x \times L_y \rightarrow I$ be an image, with dimension $L_x=1,2,\ldots,n_x$ and $L_y=1,2,\ldots,n_y$, and gray level $G=0,1,\ldots,m-1$. Let $d$ be the distance between two pixel position $(x_1,y_1)$ and $(x_2,y_2)$. The immediate neighbors of any pixel can lie on four possible direction :$\Theta = 0^o,45^o,90^o$ and $135^o$, as indicated in figure (4). The Gray Scale matrices is constructed by observing pairs of image cells distance $d$ from each other and incrementing the matrix position corresponding to the gray level of both cells [8]. This allows us to derive four matrices for each given distances $P(0^o,d)$, $P(45^o,d)$, $P(90^o,d)$, $P(135^o,d)$. For instance, $P(0^o,d)$ is defined as follows:

$$P(0^\circ, d) = \{P^\circ(i, j); i \in [0, m], j \in [0, m]\} \quad (4)$$

Where each P(i,j) value is the number of time when:
$F(x_1,y_1)=i$, $f(x_2,y_2)=j$, $|x_1-x_2|=d$ and $y_1=y_2$ append Simultaneously in the image. $P(45^o,d)$, $P(90^o,d)$, $P(135^o,d)$ are define similarly.

The matrix are normalized and features derived from them. Many feature can derived directly [8][17]. Figure(4-b) considering a 4 x 4 image with gray scale in the range 0 to 3. the gray scale matrices in the $0^o$, $45^o$, $90^o$, and $135^o$ directions are shown in figure(4- c).




(a)

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

(b)

$$P(1,0^o)=\begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \qquad P(1,45^o)=\begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$P(1,90^o)=\begin{pmatrix} 9 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix} \qquad P(1,135^o)=\begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(c)

**Figure (4):( a) Eight nearest-neighbor resolution cell, (b) A 4*4 image with 0 to 3 gray scale values, (c) Calculation of gray Scale matrices in four direction.**

## 5. The *K* Nearest-Neighbors Classifier

*k*-Nearest Neighbors approximation method is a very simple, but powerful method. It has been used in many different applications and particularly in classification tasks. The key idea behind the *k*-NN is that similar input data vectors have similar output values. One has to look for a certain number of nearest neighbors, according to Euclidean distance, and their corresponding output values to get the output approximation. We can calculate the estimation of the outputs by using the average of the outputs of the neighbors in the neighborhood.

When using the $K$ nearest-neighbors classifier ($K$NN), for each class $V$ in the training set, the ideal feature vectors are given as $fv$. Then we detect and measure the features of the unknown character (represented as $U$). To determine the class $R$ of the character we measure the similarity with each class by computing the distance between the feature vector $fv$ and $U$ the distance measure used here is the Euclidean distance. Then the distance computed $dv$ of the unknown character from class $V$ is given by

$$dv = \left[ \sum_{j=1}^{N} \left( U_j - f_{vj} \right)^2 \right]^{1/2} \qquad (5)$$

where $J$=1,2,..., N (N is the number of the features considered).

The character is then assigned to the class $R$ such that:

$$d_R = \min \left( d_v \right) \qquad (6)$$

where ($R$=1,...., no of classes)[15].

## 6. Experimental Result

A number of experimental were carried out to show the effectiveness of the proposed algorithms. Forty five sample for each character were selected (using the font: simplified Arabic, Tahoma, traditional Arabic). For the purpose of the classification the sample images were divided into 20 training and 25 test images. Images in the test set did not appear in the training sets.

Feature were extracted using five distances ($d$=1,2,3,4,5) and four directions $\Theta$ =(0°,45°,90°,135°). This gives each input character image 20 matrices of dimension 2*2. For each 2*2 gray scale matrix derived from a binary character image, there are only three independent values due to the diagonal symmetry. The three values are used directly as features. So we have 60 = (4*5*3) features per character image; different combinations of feature sets, e.g. features at $d$=1, 2, 3 and four directions (given above) were used (i.e. there were a total of 36 features (3*3*4)), etc. Table (2).

**Table (2) The calcification accuracy of the gray scale matrix technique**

| Distance | d=1,2,3,4 | d=1,2,3 | d=1,2 | d=1 |
|---|---|---|---|---|
| Calcification rat | 90.3 | 83.8 | 81.2 | 75.2 |

This method performs extremely well with recognition rates 90.3%.

## 7. Conclusion

In this paper we have presented a technique for recognizing printed Arabic characters. A number of experiment have been conducted. The experiment used 45 sample for each character. Features were extracted from character image by using Gray Scale Matrix technique. Recognition was performed through using *K* Nearest-Neighbors (*K*-NN), classification. The results obtained were very promising and recognition as high as 90.3% was indicated.

All of this demonstrates that the new method is able to handle printed Arabic character task efficiently. It is a promising technique for recognition printed Arabic character.

## References


[1] Abuhaiba, I.S.I. and Mahmoud, S.A., 1994, "Recognition of Handwritten Cursive Arabic Character", IEEE Transaction on Pattern Analysis and machine Intelligence ,Vol .16.No 6:664-672.

[2] AL-Zubaidy, L., 2002, "Arabic Machine Printed/Handwritten Character Recognition Using Neural Network", Ph.D. Thesis, Mousl university.

[3] Amin, A. 1997,"Arabic Character Recognition, A Survey", 4th International Conference Document Analysis and Recognition (ICDAR 97).

[4] Amin, A., 1997, "Arabic Character Recognition, Handbook of Character Recognition and Document Image Analysis". World Scientific Publishing Company.

[5] Amin, A. and Kavianifar, M., 1999, "Automatic Recognition of Printed Arabic Text Using Neural Network classifier", IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. 20,No. 1.

[6] Amin, A., 2000, "Recognition of Printed Arabic Text Based on Global Features and Decision Tree Learning Techniques", Pattern Recognition , Vol. 33,No. 8, : 1309-1323.

[7] Akiyama, T., Hagita N., 1990, "Automatic Entry System for Printed Documents", Pattern Recognition , 23: 1141–1154.

[8] Gotlied, G., and kreyssig H., 1990, "Texture Descriptors Based on Co-Dccurrence Matrices", computer vision, Graphics, and Image processing, 51:70-86.

[9] Haralick,R.M. K. Shanmugan, and ITshak Dinstein, 1973, "Texture for image classification". IEEE Transactions on Systems, man and Cybernetics, 3(6):610-621.

[10] Haralick, R.M., 1979,"Statistical and Structural Approaches to Textures", Proc. IEEE, vol. 67, pp. 786-804.

[11] Khorsheed, M., 2000, "Automatic Recognition Of Words In Arabic Manuscripts" Ph.D. thesis, Churchill College, University Of Cambridge.

[12] Klassen, T., 2001, "Towards Neural Net work Recognition Of Handwritten Arabic Letter", Master Thesis, Dolhousie University, Halifax,U.S.A.

[13] Mitchell, T.M., 1996, "Machine Learning". McGraw Hill, New York, NY.

[14] Pavlidis, T, Zhou J., 1992, "Page segmentation and classification", Computer Vision Graphics Image Process.54: 484–496.

[15] Said, H. E. S. Tan, T. N. and Baker, K. D., 1998, "Personal Identification Based on Handwriting", Pattern Recognition, vol.33, no.1, pp.149-160.

[16] Trier, O.D. Jain,k.a.,Taxt,t., 1996, "Feature Extraction Methods for character Recognition – A survey". Pattern Recognition. Vol.29,no4,pp.641-662.

[17] Tsai, Wen-Hsiang, 1984, "Moment-Preserving Thresholding: A New Approach" Document Image Analysis. p. 44-59.

[18] Visen, N.S. Paliwal, J., 2002, "Effect of Gray Level Quantization on Textural Classification of Cereal Grains using Machine Vision". AIC 2002 Meeting CSAE/SCGR Program , Paper No. 02-308.

على حل موضعي (Local solution) للمعادلة التكاملية – التفاضلية اللاخطية (3) مع الشرط الابتدائي (4) وذلك باستخدام شرط ليبشز (15)و(16) وبأخذ قيمة $\alpha$ $(0 < \alpha < 1)$ .

## REFERENCES


1. AL – Aidaroos H.A., Thesis M.Sc. Iraq, Mosul University (1999).
2. Barret J.H., Canad. J. Maths,No. 6: 529-541 (1954).
3. Bassam, M.A., Eurdie Reive and Aynewardte Mathematic (1965).
4. Butris, R.N., Thesis M.Sc. Iraq, Mosul University, (1984).
5. Diethelm K., Ford N., Numerical Analysis Report ., 379: (2001).
6. Diethelm K., Ford N., Appl. Math &Comput (2003).
7. Maindari F.,Fractionals calculus, No. 5: 291-238 (1997).
8. Meerschaert M.C.,J.Comput, No. 1:249-261(2006).
9. Scalas E,R., gorenflo and Mainardi F,A284, 376-384 (2000).