Arabic Characters Recognition by Edge Detection Using Connected Component Contour(CO3)

Hind R. M. Shaban
Dept. of Computer/Mathematical and Computer Science College/
University of Kufa
hindrustum.shaaban@uokufa.edu.iq

Received in :6 May 2012 Accepted in :24 Septemer 2013

Abstract

In the present paper, Arabic Character Recognition Edge detection method based on contour and connected components is proposed. First stage contour extraction feature is introduced to tackle the Arabic characters edge detection problem, where the aim is to extract the edge information presented in the Arabic characters, since it is crucial to understand the character content.

The second stage connected components appling for the same characters to find edge detection. The proposed approach exploits a number of connected components, which move on the character by character intensity values, to establish matrix, which represents the edge information at each pixel location .

The third stage the euclidean distance and vector angle are combined by using a saturation-based combination for edge detection using connected component Contour(CO3) for each character.

The system has been tested on MATLAB environment with satisfactory results. Given a better device the result should increase in accuracy significantly. Fonts show that the accuracy of the proposed method is 97.4% correct characters identification in average. The contour code technique seems to be very promising producing top results. The experimental results confirm the effectiveness of the proposed algorithm.

Keywords: Characters Recognition ,Edge detection , Contour ,Connected Components.

Introduction

Pattern recognition in image processing encompasses several areas of research, viz., face recognition, signature recognition, text recognition, and fingerprint recognition. High accuracy text recognition or optical character recognition (OCR) is a challenging task for scripts of languages. The OCR research for the English script has matured. Commercial software is available for reading printed English text. For English and Kanji scripts, good progress has been made towards the recognition of printed scripts, and the focus nowadays is on the recognition of handwritten characters [1]. The speed of information access is increased because there is possibility of search and edit in contrast to pictures and the required space for information saving is decreased because text files that are extracted from pictures usually take less space than picture files[2].

Arabic character characteristics

There are several literature survey methods and techniques that are used for some of them are:

1- Recognition of handwritten Arabic characters by uing reduction techniques[8] apply four reduction techniques for handwritten Arabic character recognition. The applied reduction method are condensed nearest neighbour (CNN), edited nearest neighbour (ENN), Instance-based learning algorithm 2 (IB2) and Instance-based learning algorithm 3 (IB3). In addition, a technique that performs a pre-processing of the handwritten Arabic characters patterns is proposed.

2- License plate recognition algorithm for passenger cars in chinese residential areas[9]appling solution for the license plate recognition problem in residential community administrations in China. License plate images are pre-processed through gradation, middle value filters and edge detection. In the license plate localization module the number of edge points, the length of license plate area and the number of each line of edge points are used for localization. In the recognition module, the paper applies a statistical character method combined with a structure character method to obtain the characters. In addition, more models and template library for the characters which have less difference between each other are built. A character classifier is designed and a fuzzy recognition method is proposed based on the fuzzy decision-making method. Experiments show that the recognition accuracy rate is up to 92%.

3-Keywords image retrieval in historical handwritten Arabic documents[10] appling system for spotting and searching keywords in handwritten Arabic documents. A slightly modified dynamic time warping algorithm is used to measure similarities between words. Two sets of features are generated from the outer contour of the words/word-parts. The first set is based on the angles between nodes on the contour and the second set is based on the shape context features taken from the outer contour. The performance of the presented system was very encouraging in terms of efficiency and match rates.

The Arabic alphabet contains basically 28 letters which are written from right to left. Fifteen of them have dots and 13 are without dots. Dots above and below the characters ten of them have one dot (baa,jeem,khaa,thal,Zai,dhad,dha,gahin,faa and noon) three have two dots (taand qaf) and two have three dots (tha and sheen)Each character can take from two to five different shapes, according to its position (beginning, middle, end or isolated)[5] .This is the mean that Arabic recognition complex. The other reason is the similarities among the different characters and the differences among the same character.

Each letter can take from two to five different shapes, thus, roughly the alphabet set can expand to 84 different shapes according to the position of the letter (beginning, middle, end or isolated).

This is the first reason that makes Arabic recognition complex. The second reason is the similarities among the different letters and the differences among the same latter[1].

Arabic characters contain many fonts and shapes (up to four different shapes) depending on there relative position in the sub-word, Table (1) showed different forms of Arabic characters and figure(1) showed samples of data base system.

Proposed research methodology

This section describes the proposed methodology for the extracting of the contour between the two segmentation points. Constructing a connected component consists of growing sets of pixels that are connected and have the same value of a property. This could be accomplished by first finding a pixel with a given property value, then looking at all its neighbors, labeling each that has the same value as being in the same component, and so on. This leads to a somewhat random stepping through the character, and is somewhat inefficient. Scanning the image in a specified order can develop a more systematic and efficient algorithm. Contour extraction. The contour between two consecutive segmentation points is extracted using by the following few steps.
In the first step disconnects the pixels near the first segmentation point; disconnect the pixels near the second segmentation point. Find the nearest distance of the first black pixel from the first segmentation point and the four regions base rules [3].

1. Contour Extraction feature

The algorithm proposed by Avrahami and Pratt for the extraction of contour points from the grey-level image that is input by the scanner. Using this algorithm, the effect of the error introduced by the conversion of the grey-level image to a bi-level image can be avoided. It can extract contour points in sub pixel precision[4]. A novel feature extraction technique is proposed to extract the feature between the two contours. The values for feature extracted are structural features from the contour profile. So the feature is named as contour code feature[3]. The rate of change of slope along with the point find all 8-connected pixels of a region that are adjacent to the background select a starting pixel and track the boundary until it comes back with the starting pixel where it is changing is extracted. With the contour slop, a few additional values that count number of ascenders, number of descanters, start point, end point, etc. are taken into consideration to capture the structural properties of the contour profile. Contour boundary scan the image from right to left and from top to bottom until an 1pixel is found
1)Stop if this is the initial pixel
2)If it is 1, add it to the boundary
3)Go to a 04-neighbor on its left
4)Check the 8-neighbors of the current pixel and go to the first 1pixel found in clockwise order
5)Go to step 2
Constructing a connected component consists of growing sets of pixels that are connected and have the same value of a property. This could be accomplished by first finding a pixel with a given property value, then looking at all its neighbors, labeling each that has the same value as being in the same component, and so on. This leads to a somewhat random stepping through the image, and is somewhat inefficient. Scanning the image in a specified order can develop a more systematic and efficient algorithm. If assume the region B[i,j] then B[i,j] = 1 if (i,j)in the region, 0 otherwise figure(2)showed center points for the some characters.
The area of region B[i,j] is computed:

$$A = \sum_{i=1}^{N} \sum_{j=1}^{M} B[i, j]$$

……..……[1]

Transforming a character image into a feature vector is an important step of character recognition. Many efforts have been made to design effective methods to extract features. Better features provide more classification information. The feature of contour direction (FCD), which is based on pixel orientation, is one of the effective features.

For feature of contour direction we input pattern which is normalized in a 120×69 image. Then contour extraction and dot orientation is completed. The preprocessed image is divided into k×k uniform rectangular zones. The FCD is calculated as follows[5]:

$$p_i^j = (d_{i1}^j, d_{i2}^j, d_{i3}^j, d_{i4}^j)$$ ……..…… [2]

$$s_j = \sum_{i=1}^{t} p_i^j$$ ……..……... [3]

$$v = (s_1, s_1, \cdots, s_T)$$ ……..…….[4]

p_i^j is the vector of four direction elements of i-th pixel in j-th sub-area, and sj is that of j-th sub-area. v is the FCD of input pattern. T and t denote the number of sub-area and that of pixel in each sub-area respectively. Since each sub-area has four dimensions, the FCD vector for one character has k×k×4 dimensions.

A novel feature extraction technique is proposed to extract the feature between the two contours. The values for feature extracted are structural feature from the contour profile. So the feature is named as contour code feature. The rate of change of slope along with the point where it is changing is extracted. With the contour slope, a few additional values that count number of ascenders, number of descenders, start point, end point, etc. are taken into consideration to capture the structural properties of the contour profile. The contour code feature vector size is taken as 25. With the coordinate the slope of two consecutive points are estimated. The slope estimation is done using linear probing. Linear regression attempts to explain this relationship with a straight line fit to the data.

The following values are calculated and stored as a contour code in a single dimension vector.

Point of change: The point with respect to the main body of the contour where the slope is changing is taken.

Direction change (up/down): The point with respect to the main body of the contour where the direction is changing is taken. The change of direction is denoted by the contour, which is changing the direction upwards to downward or vice versa.

Number of ascenders: The number of point above the upper baseline is counted and stored.

Number of descender: The number of point below the lower baseline is counted and stored.

Start point: Start point of a character (position with respect to baselines) is detected and stored.

End point: End point of a character (position with respect to baselines) is detected and stored.

2.Connected components

A connected component is a set of connected pixels that share a specific property V. Two pixels, p and q, are connected if there is a path from p to q of pixels with property V. A path is an ordered sequence of pixels such that any two adjacent pixels in the sequence are neighbors. All of the pixels in the same object are connected. There are separated green objects. It is possible to find a path between any two pixels within any of the objects, but not between pixels in different objects.

Once region boundaries have been detected, it is often useful to extract regions which are not separated by a boundary. Any set of pixels which is not separated by a boundary is caledl

connected. Each maximal region of connected pixels is called a connected component. The set of connected components partition an image into segments .

Let ∂s be a neighborhood system
– 4-point neighborhood system
– 8-point neighborhood system
Let c(s) be the set of neighbors that are connected to the point s.
For all s and r, the set c(s) must have the properties that figure(3)which showed connected components for center points.
We compute extract connected set for each unlabeled pixel

```
Class label = 1
Initialize Yr = 0 for r 2 S
For each s ∈S {
if(Ys = 0) {
Connected set(s, Y,ClassLabel)
ClassLabel   classLabel + 1
}
 }
```

Edge detector uses two operators: euclidean distance and vector angle. The euclidean distance is a good operator for finding edges based on intensity and the Vector Angle is a good operator for finding edges based on hue and saturation[6].
The euclidean distance between two pixels is defined as[11]:

$$D_e(p,q) = \left[(x-s)^2 + (y-t)^2\right]^{\frac{1}{2}} \quad \text{..... [5]}$$

The euclidean distance and vector angle are combined by using a saturation-based combination method[7].

Experimental results

In this section a detailed experimental comparison of the above stated algorithms has been presented..
Ultimate erosion character using expression in matlab :
I1=bwulterode(I,method, connectivity)

Where I is image character that computes the ultimate erosion of the binary image I. The ultimate erosion of I consists of the regional maxima of the distance transform of the complement of I.
Specifies the distance transform method and the regional maxima connectivity. The types for finding distance are 'euclidean', 'cityblock', 'chessboard', and 'quasi-euclidean'. For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a distance function or metric if

(a) $D(p,q) \geq 0$ ($D(p,q)=0$ iff $p = q$),
(b) $D(p,q)=D(q,p)$, and
(c) $D(p,z) \leq D(p,q) + D(q,z)$.

The detector applies both operators to the RGB color space of an image, and then combines the results from each based on the amount of color in a region. There is a difference vector and a vector gradient version; we chose to implement the vector gradient version. The pixels with *D*8 distance from (x, y) less than or equal to some value r form a square centered at (x, y).
The vector angle between two pixels is approximated by:

$$\sin\theta = \left[1 - \left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|}\right)^2\right]^{1/2} \quad \text{..... [6]}$$

Where v1 and v2 are triplets (v = [R G B]). Because sin θ ≈ θ for small angles. Again, v1 and v2 are RGB triplets (v = [R G B]).

The **algorithm** for finding edges in the image is as follows:
1. Start
2. For each pixel in the image, take the 3x3 window of pixels surrounding that pixel.
3. Calculate the saturation-based combination of the Euclidean Distance and Vector Angle between the center point and each of the eight points around it.
4. Assign the largest value obtained to the center pixel.
5. When each pixel has had a value assigned to it, run the results through a threshold to eliminate false edges(slope threshold=1.5 and high threshold=15).
6.End

Figure(8) showed slope threshold=1.5 and high threshold=15 for Arabic character.
Figure(4) showed connected components for center region for each character.

We use some statements in matlab language like :
1-'remove': removes interior pixels. This option sets a pixel to 0 if all its 4-connected neighbors are 1, thus leaving only the boundary pixels on.

2-'skel': With n = information , removes pixels on the boundaries of objects but does not allow objects to break apart. The pixels remaining make up the image Figure(5)showed edge detection for character by remove and skeleton.

The area of objects for characters are shown in figure(6) the index for characters start from(1:أ) to (28:ي) and Figure(7)showed samples for contour for some characters.

Conclusion

Edge detection using connected component contour(CO3) seems to be well suited for Arabic character recognition. Many of the problems that arise when using connected component contour to recognize Arabic characters can easily be eliminated or reduced by adding routines that scales, centers etc.

Other problems like noise and there is not much one can do about those problems except improving the quality of the equipment used when scanning/reading characters .
We have tried to recognize each character written in Arabic language for free size by using normalization step for our algorithm .

The strength of algorithm that we have used is that it is even able to recognize blocks in which the characters are slightly joined.

This is possible due to the use of ingenious continuous matching algorithm and intelligently removing noise and combined euclidean distance and vector angle.

The experiments reported in the present paper showed that fast recognize differs qualitatively from different characters in the recognition of Arabic characters.

References

1. Majumdar, A.B. (2007) Basic Character Recognition Using Digital Curvelet Transform, Journal of Pattern Recognition Research , 1 (1):17-26.

2. Jenabzade,M.R.; Azmi,R.A.; Pishgoo,B.A. and Shirazi, S. A. (2011) Two Methods for Recognition of Hand Written Farsi Characters, International Journal of Image Processing (IJIP), 5 (4):512-520.
3. Verma,B.Z (2003) A Contour Code Feature Based Segmentation For Handwriting Recognition, Proceedings of the Seventh International Conference on Document Analysis and Recognition IEEE. 1 (1):70-79.
4. Itoh,K.A. and Ohno,Y.A. (1993) A curve fitting algorithm for character fonts, Electronic publishing, 6(3):195–205 .
5. Haraty , R.A. and Ghaddar ,C.A. (2004) Arabic text recognition, The International Arab Journal of Information Technology, 1(2): 156-163.
6. Izakian,H.A; Monadjemi ,S. A.; Tork ,B.L. and Zamanifar ,K.S. (2008) Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes, World Academy of Science, Engineering and Technology, 4(43): 67-70.
7. Nadernejad,E.A. (2008) Edge Detection Techniques: Evaluations and Comparisons, Applied Mathematical Sciences, 2 (31):1507 – 1520.
8. Salah ,M. Al-Saleh, Salameh A. Mjlae, and Salim A. Alkhawaldeh (2009) Recognition of Handwritten Arabic Characters by Using Reduction Techniques6 (6) : 918-923.
9. Lisheng Jin, Huacai Xian, Jing Bie, Yuqin Sun, Haijing Hou and Qingning Niu (2012) License Plate Recognition Algorithm for Passenger Cars in Chinese Residential Areas, International Journal on Document Analysis and Recognition (IJDAR).
10.Raid Saabni and Jihad El-Sana (2013) Keywords image retrieval in historical handwritten Arabic documents, Triangle R&D Center, Kafr Qarea, Israel, 22(1)
11.Liwei Wang ,Yan Zhang and Jufu Feng,School of Electronices Engineering and computer sciences,peking university,china.
http://www.cis.pku.edu.cn/faculty/vision/wangliwei/pdf/IMED.pdf

Table No. (1): The different forms of Arabic characters

Character	Isolated character	Initial	Middle	Final
Alef	ا			ـا
Ba	ب	بـ	ـبـ	ـب
Ta	ت	تـ	ـتـ	ـت
Tha	ث	ثـ	ـثـ	ـث
Jeem	ج	جـ	ـجـ	ـج
H’a	ح	حـ	ـحـ	ـح
Kha	خ	خـ	ـخـ	ـخ
Dal	د			ـد
Thal	ذ			ـذ
Ra	ر			ـر
Zai	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
Dha	ظ	ظـ	ـظـ	ـظ
Ain	ع	عـ	ـعـ	ـع
Gahin	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
Waw	و			ـو
Ya	ي	يـ	ـيـ	ـي

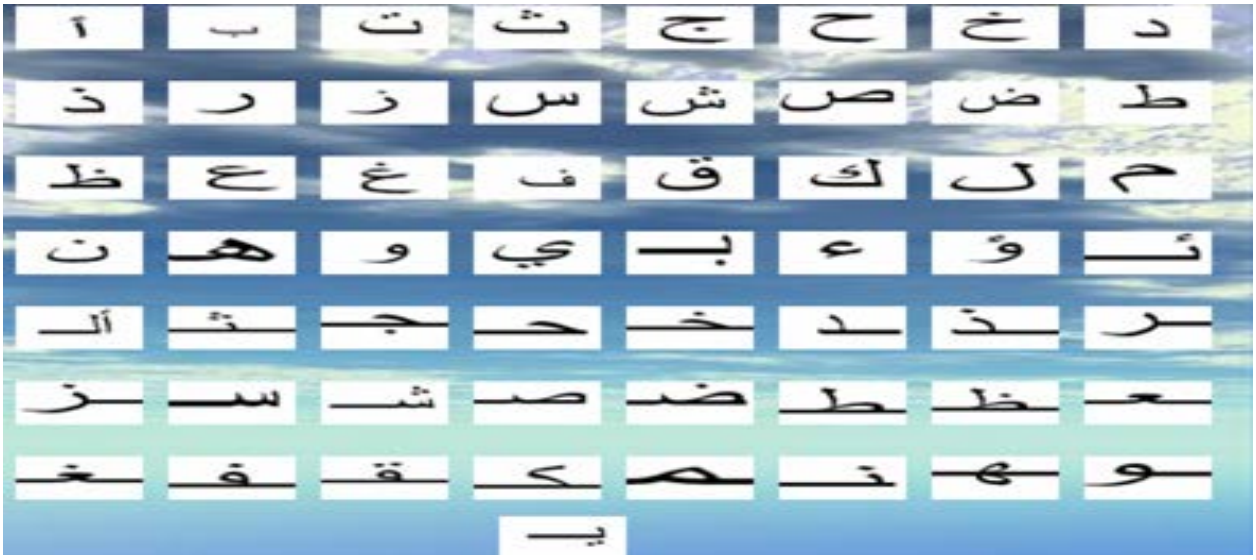


Figure No.(1):Samples of data base system

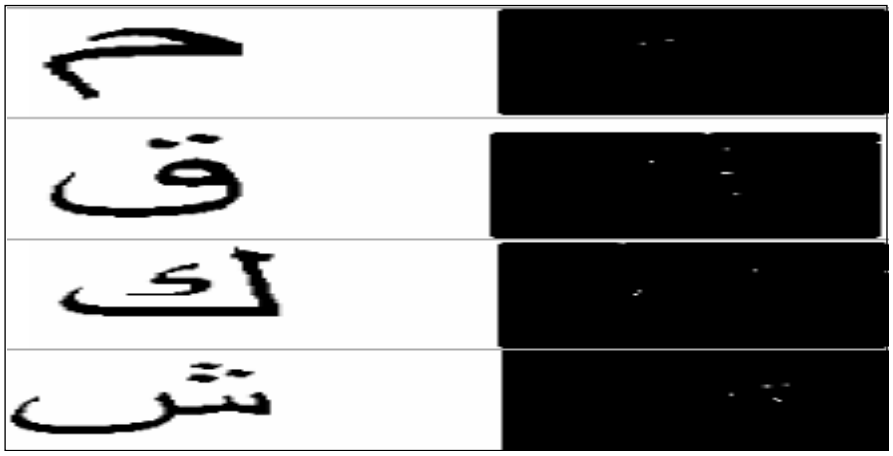


Figure No.(2):Center points for some characters.

Figure No.(3):Connected components for center points

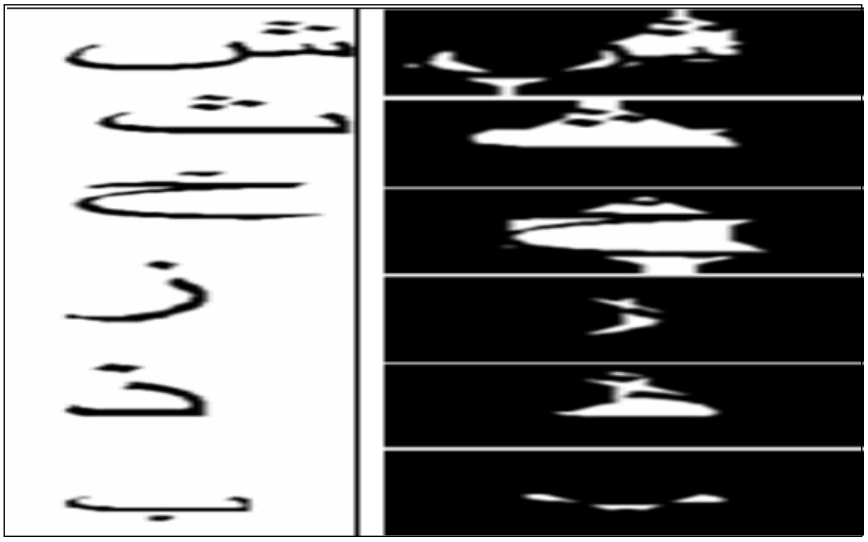


Figure No.(4):Connected components for center region for each character

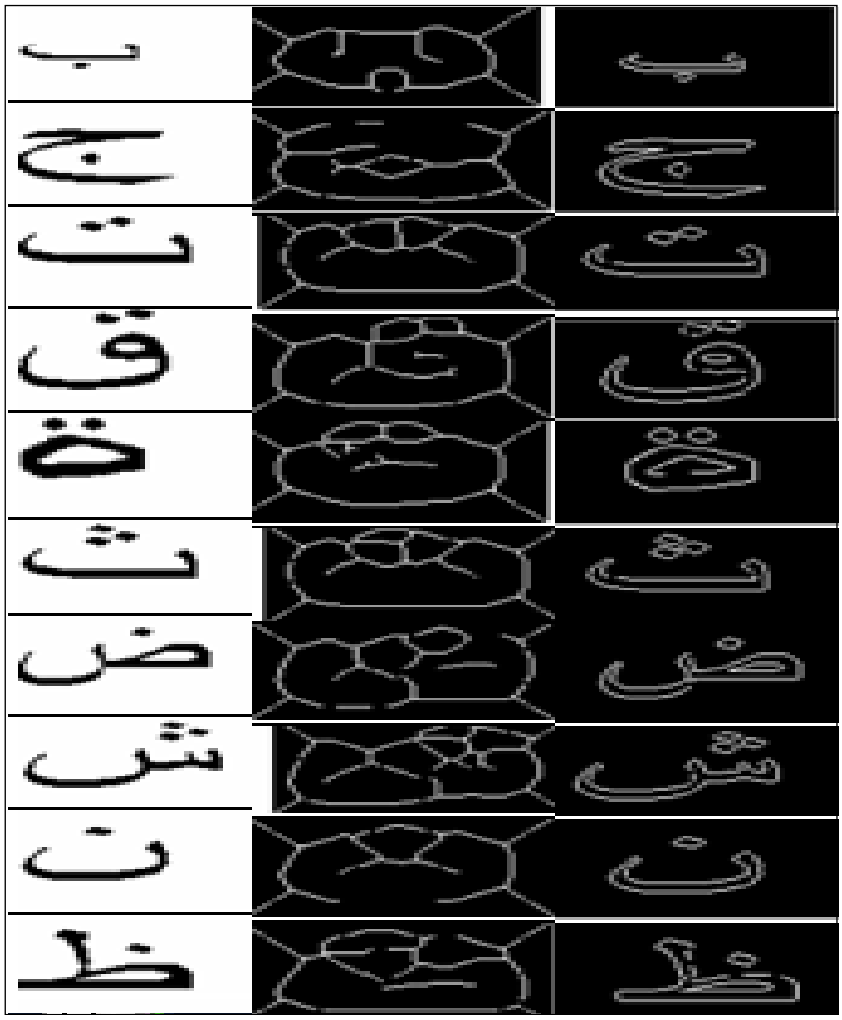


Figure No.(5):Edge detection for character by removing and skel

Character index	Area	Character index	Area
أ	8.0996e+003	ص	7.6828e+003
ب	8.1375e+003	ض	7.6589e+003
ت	7718	ط	7.5645e+003
ث	7.6826e+003	ظ	7528
ج	7.6231e+003	ع	7.7 438e+003
ح	7.6586e+003	غ	7.7 153e+003
خ	7.6199e+003	ف	7.9261e+003
د	7.9534e+003	ق	7.4749e+003
ذ	7.9051e+003	ك	7.4526e+003
ر	8.0531e+003	ل	7.5588e+003
ز	8.0225e+003	م	7.5243e+003
س	7.7 435e+003	و	7.9426e+003
ش	7.647 1e+003	ه	6.8744e+003
		ن	7.8476e+003
		ي	7.4476e+003

Figure No.(6):Area for each character

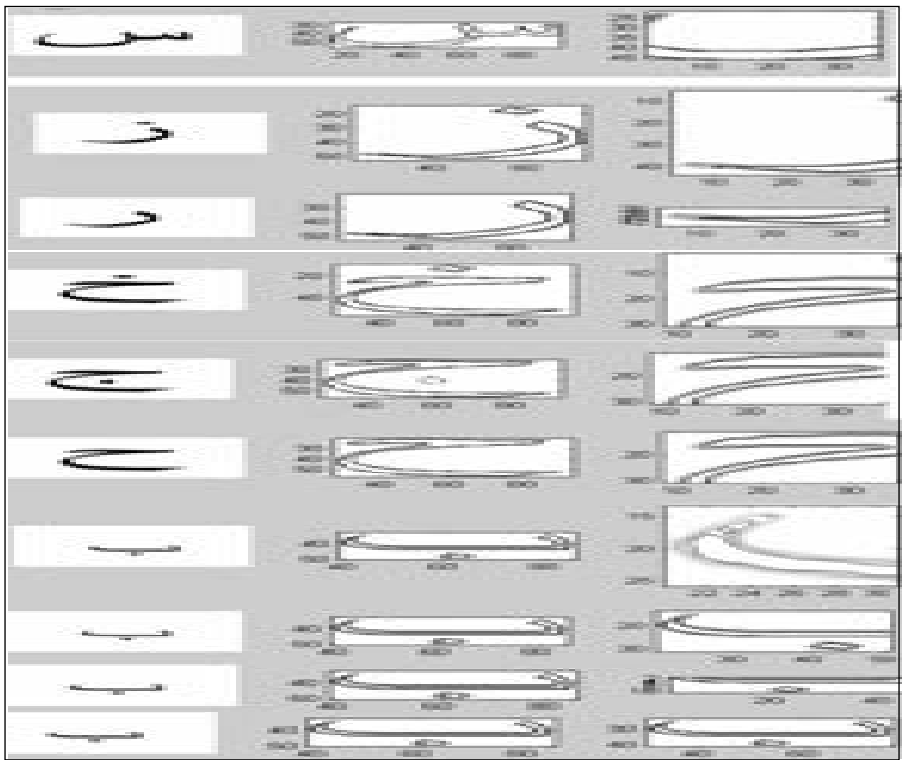
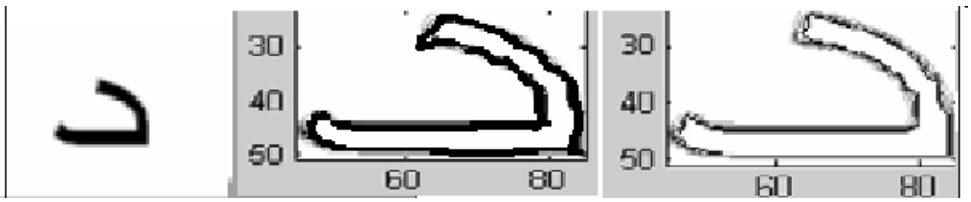


Figure No.(7):Samples for contour for some characters



(a) (b) (c)

Figure No.(8): (a) Original Arabic character(د) (b) Edge detector slope threshold=1.5 (c) Edge detector high threshold=15

تحديد الحواف لتمييز الحروف العربية باستخدام الكفاف وارتباط المكونات

هند رستم محمد شعبان
قسم الحاسوب/كلية الرياضيات وعلوم الحاسوب/جامعة الكوفة
hindrustum.shaaban@uokufa.edu.iq

استلم البحث في :6 اذار 2012 ، قبل البحث في 24 ايلول 2013

الخلاصة

اقترحت طريقة لتمييز الحروف العربية بكشف الحافة باستخدام كلا من contour واتصال وارتباط المكونات للحروف.
تم التوضيح والعرض في المرحلة الأولى ميزة استخراج كونتور لمعالجة الأحرف العربيةحافة مشكلة للكشف عن الحرف، اذ كان الهدف هو استخراج المعلومات الحافة المقدمة في الأحرف العربية، لأنه أمر بالغ الأهمية لفهم المحتوى والمعلومات الخاصة بالأحرف.
اما في المرحلة الثانية فنقوم بعملية ارتباط المكونات للحروف الخاصة نفسها بالمرحلة الأولى للعثور على الكشف عن الحافة. الحوارزمية المقترحة تستغل عدد من الارتباطات، للمكونات مبينة تباين قيم الكثافة المحلية للحرف و التي تمثل حافة المعلومات في كل موقع بكسل.
يتم الجمع في المرحلة الثالثة بين المسافة الإقليدية وزاوية المتجهات باستخدام مزيج المستندة إلى التشبع للكشف عن حافة طريق اتصال كونتور المكون (CO3) لكل حرف.
اختبر هذا النظام على بيئة MATLAB مع نتائج مرضية. نتائج النظام بينت دقة الطريقة المقترحة اذ كانت النتائج الصحيحة 97.4٪ وهذا يبين ان النتائج التجريبية تؤكد فعالية الخوارزمية المقترحة.

الكلمات المفتاحية : تمييز الحروف ،تحديد الحواف، الخط الكفافي ، ارتباط المكونات