Recognition for Arabic Character Based on Edge and BPNN

Liying Zheng

Abstract—**This paper proposed a new method for recognizing machine printed Arabic characters. First, four edges (left, upper, lower, and right edge) are extracted from the character image. Next, some features are extracted from the edges. Finally, a BPNN, isolated forms of Arabic characters are recognized. The new method is tested on two fonts and 9 sizes, and its recognition rate is over 97%.**

Index Terms— **Recognition, Arabic Character, Edge, Back propagation neural network.**

I. INTRODUCTION

Machine simulation of human reading has been the subject of intensive research for many years. A lot of work has been done on English, Chinese, as well as Japanese characters. As a language spoken by over 200 million people, the research work conducted on Arabic character began in 1980s. Since then various recognition methods have been proposed, such as the methods based on image density[1]-[5], the methods based on artificial neural networks[6]-[8], and the methods based on primitive features and decision tree [9],[10]. However, in comparison with other languages, there is a little work has been conducted on automatic recognition of Arabic character [11].

In this paper, a new character recognition method, which is based on four edges of Arabic character, as well as back propagation neural networks (BPNN), is proposed. First, the four edges, which are termed as left, upper, right, and lower edge, respectively, are extracted from a character image. Next, some features are extracted from the edges. Finally, isolated forms of Arabic characters are recognized with a back propagation neural network (BPNN). The proposed method is tested on two fonts and 9 sizes; its average recognition rate is over 97%, and in most cases is about 99%.

The paper is organized as following. In section2, algorithms of edges extraction and representation are introduced. Feature extraction and Recognition method using BPNN are clarified in section3 and 4, respectively. The Experimental results and some analysis are given in section 5. Some conclusions are given in section 6.

II. EDGE EXTRACTION AND REPRESENTATION

A.. Edge Extraction

In this paper, four edges of an Arabic character image are adopted for character recognition; therefore, the first stage of our algorithm is edge extraction.

Manuscript received July 2, 2008. This work was supported in part by Basic Science Foundation of Harbin Engineering University.

L. Zheng is with School of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China. Email: zhengliying@hrbeu.edu.cn.

Let $E_i=\{e_{i1},e_{i2},\ldots,ei_{Ni}\}$ (1)

where E_i is the ith edge of a character, the values 1 through 4 of suffixes i represent left, upper, right and lower edge, respectively; e_{ip} is the pth point of E_i; N_i is the length of E_i.

Extraction process for each E_i is similar, except that the scanning direction is varied, which are left-right, up-bottom, right-left, and bottom-up for E_1, E_2, E_3, and E_4, respectively. Following are the main steps for extracting E_1.

Step1. p=0;

Step2. Scanning a character image from left to right, and recording horizontal projection value at the pth row as $HPro[p]$;

Setp3. If $HPro[p]$>2, let e_{1p} be equal to the column position of the first black pixel at the pth row; if there is no such black pixel, e_{1p}= 0.

Step4. p=p+1; if p smaller than the height of the character image (N_1), return to Step2; otherwise go to Step5;

Setp5. Using (2) to smooth E_1.

$$e_{1p}=(e_{1p-1}+e_{1p}+e_{1p+1})/3 \quad (2)$$

where p=2,3,…,N_1.

Fig.1 shows the edges of four Arabic characters (Ain, Sad, Noon, Dal) having been extracted with above algorithm.

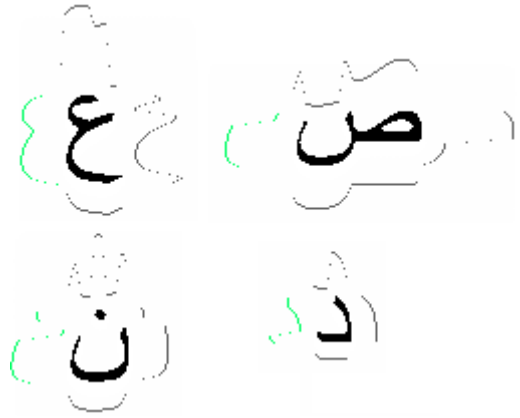

Fig.1 Edges of some Arabic characters

B. Edge Representation

To extract features from character edges, a list of symbols of each edge is needed. The process of representing edge, E_i, is following:

Step1. Computing the difference of E_i with (3) and (4).

$$dE_i=\{de_{ip}\} \quad (3)$$

$$de_{ip}=\begin{cases}1 & if \quad e_{ip}-e_{ip-1}>0\\ 0 & if \quad e_{ip}-e_{ip-1}=0\\ -1 & if \quad e_{ip}-e_{ip-1}<0\end{cases} \quad (4)$$

where dE_i is the difference of E_i and de_{ip} is its pth point.

Step2. Using following rules to find out three types of

line---straight line, slant line with acute angle and slant line with obtuse angle.

Rule1 If there are p and q satisfying

$$p < q, de_{ip-1} \neq 0, de_{ip+1} = 0$$

and $$de_{ip} = de_{ip+1} = \Lambda = de_{ip-1} = 0$$

then p, q, and p-q-1are start point, end point, and length of a straight line, respectively.

Rule2 If there are p and q satisfying

$$p < q, de_{ip-1} \geq 0, de_{ip+1} \geq 0$$

and $$de_{ip} = de_{ip+1} = \Lambda = de_{ip-1} = -1$$

then p, q, and p-q-1 are start point, end point and length of a slant line with acute angle, respectively.

Rule3 If there are p and q, satisfying

$$p < q, de_{ip-1} \leq 0, de_{ip+1} \leq 0$$

and $$de_{ip} = de_{ip+1} = \Lambda = de_{ip-1} = 1$$

then p, q, and p-q-1 are start point end point and length of a left slant line with obtuse angle.

Step3. If the length of a line is greater than 2, record its start point, end point as well as its length .

Step4. Representing E_i by following symbol list.

$$S_i = \{(t_{im}, s_{im}, e_{im}, l_{im})\} \quad (5)$$

where t_{im} is the type of the mth line of E_i, t_{im} = 0, 1, -1, denote straight line, slant line with acute angle and slant line with obtuse angle, respectively; s_{im}, l_{im} and e_{im} are its start point, end point, and length, respectively; Mi is the number of lines within E_i.

Step5. Combining continuous lines satisfying following conditions into one line.

Let three continuous lines are (t_{ip}, s_{ip}, e_{ip}, l_{ip}), (t_{ip+1}, s_{ip+1}, e_{ip+1}, l_{ip+1}) and (t_{ip+2}, s_{ip+2}, e_{ip+2}, l_{ip+2}).

Condition 1: If $t_{ip} = t_{ip+1}$, s_{ip+1}-e_{ip}<2, and $|E_i(e_{ip})-E_i(s_{ip+1})|$<3, the second line is deleted while the first line becomes(t_{ip}, s_{ip}, e_{ip+1}, e_{ip+1}- s_{ip-1}), and $M_i = M_i$ -1.

Condition 2: If $t_{ip} = t_{ip+2}$, l_{ip+1} <3, and $|E_i(e_{ip})-E_i(s_{ip+2})|$<3, the second and the third line are deleted while the first line becomes (t_{ip}, s_{ip}, e_{ip+2}, e_{ip+2} - s_{ip-1}) , and $M_i = M_i$ -2.

Condition 3: If $\frac{|E_i(s_{ip}) - E_i(e_{ip})|}{e_{ip} - s_{ip}} > 4$, delete the first line and $Mi = Mi$ -1.

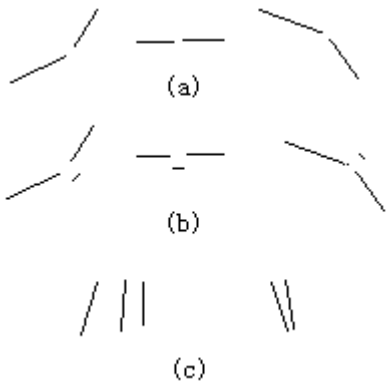

Fig.2 lines satisfying (a) Condition1, (b) Condition2, and (c) Condition3.

Condition1 means two continuous lines resembling the lines shown in Fig.2(a) will be combined; Condition2 means the three lines, resembling the lines shown in Fig.2(b) will be combined; Cndition3 means the line, whose angle (see Fig.2(c)) belongs to (75°,105°)will be deleted.

III. FEATURE EXTRACTION

Features extracted from the four edges consist of four types. The first type of features is number of lines in each edge. The second type, which is computed by using the similar formula with (6), is number of straight lines in lower edge, and that in right edge. The third type, which is computed with (7), is a ratio of the number of straight lines in lower edge (and right edge) and the character width (and height). The fourth type is the length of the longest straight line in upper edge**,** and that in right edge.

$$LN_i = \begin{cases} LN_i + 1 & if \quad \frac{|E_i(e_{ip}) - E_i(s_{ip+1})}{s_{ip+1} - e_{ip}} > 4 \\ LN_i & others \end{cases} \quad (6)$$

where LN_i is the number of lines in edge E_i (i= 1,2,3 and 4); initial value of LN_i is 0, and p=1,2,…,M_i-1.

$$SR_i = \frac{\max\limits_{j=1,2,...SNi} \{SL_{ij}\}}{C_i} \quad (7)$$

where $\{SL_{ij}\}|_{j=1,2,...SNi}$ is the set of straight lines of edge E_i; C_i = height and width of the character for i =3 and 4, respectively.

Besides above four types of features, the ratio of height and width of a character is also used during the recognition process.

VI. ARABIC CHARACTER RECOGNITION

A three-layer BPNN is employed in recognition stage. The BPNN's inputs are the features extracted in section 3; therefore, there are 11 input nodes. Each output nodes represent an isolated Arabic character (including character Lam-Alif); so, there are 29 output nodes. The initial number of hidden nodes is 20; then a pruning algorithm is chosen to decide the optimal number of hidden nodes. Final architecture of the BPNN is 11-15-29. Basic back propagation algorithm is employed to train our BPNN, i.e. weights are changed by an amount proportional to the error at that unit times the output of the unit feeding into the weight. Other parameters of the BPNN are following. Node transfer function is Sigmoid function; terminal condition is either training steps greater than 5000, or training error small than 0.1; learning rate is 0.03.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The recognition algorithm, which is discussed deeply in section 2 and section 3, has been implemented using Visual C++ program and tested with more than 500 samples of Arabic character. These samples, which are written in two fonts, Simplified Arabic and Arabic Transparent, and 9 sizes, 8, 9, 10, 11, 12, 14, 16, 18, 20, are captured using a scanner with a resolution of 300 DPI. The recognition rates for each size of Simplified Arabic are 97.4%, 97.5%, 98.6%, 99.6%, 100%, 100%, 100%, 100%, and 100%, respectively, while that of Transparent are 97.2%, 97.8%, 98.6%, 99.5%, 100%, 100%, 100%, 100%, and 100%, respectively.

From above testing results we noticed that

i The results for the two fonts are similar. This is because

these two fonts are alike and the features we extracted are invariant for these fonts.

ii In most cases, the recognition rate is greater than 99%, which clearly is a high rate. Comparing with other Arabic character recognition method, such as the method proposed in [9], this rate is also high, but in comparison to them, our method is very simple.

iii The recognition rates for small sizes are lower than that of the larger sizes. This is due to the fact that for small sizes the method sometimes can not distinguish the valid stroke and the noise, as well as the fact during binarization process, some continuous strokes are separated.

VI. CONLUSIONS.

Character recognition is one of the most important stages for any character recognition system. A new Arabic character recognition method, which is based on the four edges of Arabic characters and BPNN, has presented. The new method has been tested with a lot of samples, and high recognition rate has recorded. The proposed method is applied for isolated form of Arabic character, but it can be used for recognizing other three forms (beginning, middle and end form) with a little change in BPNN's architecture.

REFERENCES

[1] H. M. M. Hosseini and A. Bouzerdalm. "A system for Arabic character recognition", *Proc. 2th Australian and New Zealand Conf. on Intelligent Information Systems*, 1994, pp:120-124.

[2] J. Alherbish and R. Ammar,"High Performance Arabic Character Recognition", *Journal of Systems and Software*, 1998, 44,pp: 53-71

[3] N.M.Wanas, M.R.El-Sakka and M.S.Kamel. "Multiple classifier hierarchical architecture for handwritten Arabic character recognition", *International Joint Conf. on Neural Networks*, 1999, pp:2834-2838

[4] F.Bouslama and H.Kishibe. "Fuzzy Logic in the recognition of machine printed Arabic characters", *The 6th International Conf. on Neural Information Processing*, 1999, pp:16-20

[5] H.Al-Yousefi and S.Udpa. "Recognition of Arabic characters," *IEEE Trans. on Pattern Analysis Machine Intell.*, 1992, 14(8),pp: 853-857

[6] M.M. Altuwaijri and M.A.Bayoumi. "Arabic text recognition using neural networks", *IEEE International Symposium on Circuits and Systems*, 1994 ,pp; 415-418

[7] H.Y.Y. Sanossian and M.Al-karak. "An Arabic character recognition system using neural network", *Proc. of IEEE Signal Processing Society Workshop*, 1996,pp: 340-348

[8] L.Zheng. "Machine Printed Arabic Character Recognition Using S-GCM", *Proc. 18th International Conf. on Pattern Recognition*, 2006, pp:

[9] B.M.F.Bushofa and M.Spann. "Segmentation and recognition of Arabic characters by structural classification", *Image and Vision Computing*, 1997, 15,pp:167-179

[10] A.Amin. "Prototyping structural description using decision tree learning techniques", *The 16th International Conf. on Pattern Recognition*, 2002, pp:76-79

[11] A. Amin. "Off-line Arabic character recognition-a survey", *Proc. of the 4th International Conf. on Document Analysis and Recognition*, 1997,pp:596-599