NEURAL NETWORK BASED SEGMENTATION ALGORITHM FOR ARABIC CHARACTERS RECOGNITION

Nada A. Rasheed
University of Babylon
College of Basic Education

Abstract

This paper presents a novel holistic technique for classifying Arabic handwritten text documents, which it is performed in several steps. First, the Arabic handwritten document images are segmented into their connected parts. A simple heuristic segmentation algorithm is used which finds segmentation points in printed and cursive handwritten words. Second, several features are extracted from these connected parts and then combined to represent a word with one consolidated feature vector. Finally, Neocognitron type of the neural network is used to learn and classify the different fonts into word classes.

1. Introduction

Object recognition is a very difficult task. Despite many efforts to solve this problem here still are no perfect solutions (Khalid *et al*, 2003).

Character recognition is a long-standing, fundamental problem in pattern recognition. It has been the subject of a considerable number of studies and serves many useful applications (Ehsan *et al*, 2003).

Artificial Neural Networks have proven to be successful in many areas of pattern recognition**.** Some researchers have used conventional methods for segmentation and recognition, while others have used ANN based methods for the character recognition process**.** Segmentation plays an important role in the overall process of handwriting recognition. Unfortunately, not only is it a vital process but it is also one that has not achieved very accurate results. This research attempts to integrate both conventional and intelligent methods for the segmentation of difficult printed and handwritten words, followed by the accurate recognition of characters.

A simple heuristic segmentation algorithm is used which finds segmentation points in printed and cursive handwritten words. A neural network trained with valid segmentation points from a database of scanned**,** handwritten words is used to assess the correctness of the segmentation points found by the heuristic segmentation algorithm.

Following segmentation and verification, the resulting characters are identified by another Artificial Neural Network is used.

The remainder of the paper is broken down into four sections. Section 2 briefly describes the Characteristics of the Arabic Writing, Section 3 provides several Preprocessing steps are performed, Segmentation using a heuristic algorithm & Neural network trained with Neocognitron algorithm follows in Section 4, Conclusion is drawn in Section 5, and Recommendations follows in Section 6.

2. Characteristics of the Arabic Writing

Arabic language is a widely used language as more than one billion people use Arabic in either their daily activities or religion-related activities (Salama & Zaher, 2008).

Arabic is written from right to left and is always cursive. It has 29 basic letters and eight diacritics (Gheith *et al*, 2008). Printed and handwritten Arabic text is cursive and Arabic characters can have four different shapes due to their position within the word. Moreover, Arabic character shape can be changed dramatically in different fonts (Mostafa, 2004). The table below

shows the 29 letters and their various forms. Each letter has multiple forms depending on its position in the word. Each letter is drawn in an isolated form when it is written alone, and is drawn in up to three other forms when it is written connected to other letters in the word. For example, the letter Ain has four forms: *Isolated* form (ع) and *Initial*, *Medial*, and *Final* forms (ععـ ع), respectively from right to left. Moreover, letters Hamza, Teh, and Alef have other forms, as shown in the table below Within a word, every letter can connect from the right with the previous letter. However, there are six letters that do not connect from the left with the next letter see the table (Gheith *et al*, 2008).

The shapes of Arabic characters in different positions.

Character Name		Isolated	Initial	Middle	Final	Character Name		Isolated	Initial	Middle	Final
Alif	ألف	ا	ا	ـا	ـا	**Dhad**	ضاد	ض	ضـ	ـضـ	ـض
Ba'	باء	ب	بـ	ـبـ	ـب	**Tta'**	طاء	ط	طـ	ـطـ	ـط
Ta'	تاء	ت	تـ	ـتـ	ـت	**Dha'**	ظاء	ظ	ظـ	ـظـ	ـظ
Tha'	ثاء	ث	ثـ	ـثـ	ـث	**A'in**	عين	ع	عـ	ـعـ	ـع
Jeem	جيم	ج	جـ	ـجـ	ـج	**Ghain**	غين	غ	غـ	ـغـ	ـغ
H'a'	حاء	ح	حـ	ـحـ	ـح	**Fa'**	فاء	ف	فـ	ـفـ	ـف
Kha'	خاء	خ	خـ	ـخـ	ـخ	**Qaf**	قاف	ق	قـ	ـقـ	ـق
Dal	دال	د	د	ـد	ـد	**Kaf**	كاف	ك	كـ	ـكـ	ـك
Thal	ذال	ذ	ذ	ـذ	ـذ	**Lam**	لام	ل	لـ	ـلـ	ـل
Rai	راي	ر	ر	ـر	ـر	**Meem**	ميم	م	مـ	ـمـ	ـم
Zai	زاي	ز	ز	ـز	ـز	**Noon**	نون	ن	نـ	ـنـ	ـن
Seen	سين	س	سـ	ـسـ	ـس	**Ha'**	هاء	ه	هـ	ـهـ	ـه
Sheen	شين	ش	شـ	ـشـ	ـش	**Waw**	واو	و	و	ـو	ـو
Sad	صاد	ص	صـ	ـصـ	ـص	**Ya'**	ياء	ي	يـ	ـيـ	ـي

3. The Preprocessing

After the images were acquired, they were converted into monochrome bitmap (BMP) form. Before any segmentation or processing could take place, it was then necessary to convert the images into binary representations of the handwriting. The dimension of the image used in this work is (250x250) pixels.

The word images require some manipulation before the application of any segmentation. This process prepares the image and improves its quality in order to eliminate irrelevant information and to enhance the selection of the important features for recognition. This is known as preprocessing. It is performed to improve the robustness of features to be extracted.

Moreover Preprocessing steps are performed in order to reduce noise in the input images, and to remove most of the variability of the handwriting. It is well known that a person's situation differs in each word at the same time. This leads to changes in inclination angles of the same person's words. Hence, the rotation algorithm must be used to unify word orientation in a horizontal manner to overcome this problem.

It is important to compute the angle (θ), which is used in the rotation operation. The rotation of an image requires the calculation of a new position for each point of the image after the transformation. Each image point is rotated through an angle (θ) about the origin, which varies from one word to other and can be calculated according to the inclination angle. The following Algorithm is used for this purpose.

Rotation Algorithm:

Input:
X^{old} , Y^{old} The pixel which will be Rotated
L, R, D, U the boundaries of image
Output:
X^{new}, Y^{new} The Rotated pixel
θ is the angle of rotation
Program body:
Step1 : Calculating the angle (θ)
Step1.1: Calculating (S_1, S_2) which are represented the number of columns in the left and the right of object to cutting the word in the first On-pixel for up and down.

S_1=L+(R -L +1.0) /8.0

S_2=L+7.0*(R -L+1.0) /8.0

Step1.2: Calculating the averages between first on-pixel for up and down the cutting points

$C_1(x_1,y_1)$, $C_2(x_2,y_2)$

Step1.3:

Using the equation $\theta = \tan^{-1}(\frac{y_2 - y_1}{x_2 - x_1})$

Step2:

Step2.1: for i=0 … n

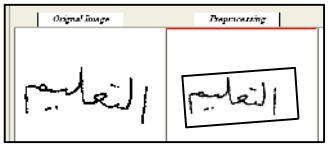
Step2.2: for j=0 … m

$X^{new} = X^{old} * \cos(\theta) - Y^{old} * \sin(\theta)$

$Y^{new} = X^{old} * \sin(\theta) + Y^{old} * \cos(\theta)$

Step3: rotated pixel at [X^{new} , Y^{new}]

end for



Before **After**

Figure 1. Image Rotation

4. Segmentation using a heuristic algorithm

A simple heuristic segmentation algorithm was implemented which scanned handwritten words for important features to identify valid segmentation points between characters. The algorithm first scanned the word looking for minima's or arcs between letters, common in handwritten cursive script. In many cases these arcs are the ideal segmentation points, however in the case of letters, such as "ص", **"م"** and "ة", an erroneous segmentation point could be identified. Therefore the algorithm incorporated a "hole seeking" component which attempted to prevent invalid segmentation points from being found.

If an arc was found, the algorithm checked to see whether it had not segmented a letter in half, by checking for a "hole". Holes, are found in letters which are totally or partially closed such as an "ص", "ن" and so on. If such a letter was found then segmentation at that point did not occur. Finally, the algorithm performed a final check to see if one segmentation point was no1 too close to another. This was done by ascertaining if the distance between the last segmentation point and the position being checked was equal to or greater than the average character width of a particular word. If the segmentation point in question was too close to the previous one, segmentation was aborted. Conversely, if the distance between the position being checked and the last segmentation point was greater than the average character width, a segmentation point was forced. The heuristic algorithm is:

Segmentation Algorithm

Step 1. Average character size for the current word is calculated, by scanning for segregated characters and noting their width and height.
Step 2. If a column of pixels exists, check its properties. Else go to Step 9.
Step 3. If the pixel density is zero, then segmentation point found. Go to Step 2.
Step 4. Check either side of point to verify whether it is located in a valley or minima. If so, go to Step 5. Else return to Step 2.
Step 5. Calculate how many columns have been passed since the last segmentation point.
Step 6. If the number of columns is greater than the average size of the character, go to Step 7. Else go back to Step 2.
Step 7. Check to see whether the point is part of a partially enclosed or totally enclosed character.
Step 8. If Step 7 is false, then segmentation point found. Repeat by going to Step 2.
Step 9. End of segmentation procedure.

5. Training of the ANN

To train the **ANN** with accurate segmentation points, the output from the heuristic segmentation algorithm is used. It was save good segmentation by the algorithm into a file, which processed then by a further step to produce training file for the ANN.

For this step, Neural Network trained with the Neocognitron algorithm. It was initially designed to recognize handwritten alphabetic characters. It is an example of a hierarchical net, the architecture consists of several layers of units. The unit within each layer are arranged in a number of square arrays. A unit in one layer receives signals from a very limited number of units in the previous layer similarly; it sends signals to only a few units in the next layer. The input units are arranged in a single 19 x 19 square array. The first layer above the input layer has 12 arrays, each consisting of 19 x 19 units. In general the size of the arrays decrease as we progress from the input layer to the output layer of the net, as shown below.

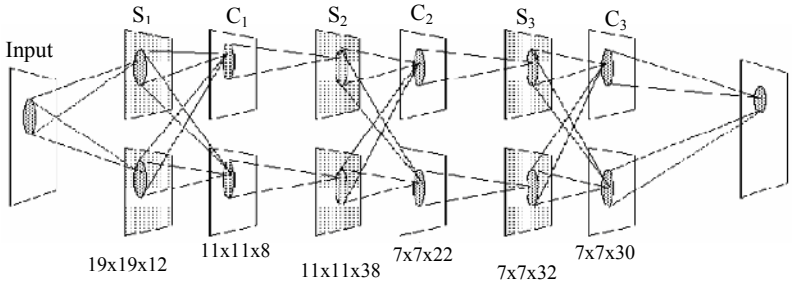


Figure 2. Layout of Neocognitron

The layers are arranged in pairs, an S-layer followed by a C-layer. The S arrays are trained to respond to the identification of valid segmentation points in the handwritten words used for experimentation, individual characters were extracted. This was achieved by beginning a search at the origin of each word and looking for a segmentation point. Once a segmentation point was found, the first character was subsequently extracted. Subsequently the end of the character was set to the starting point of the next character and a search for the next segmentation point ensued and so on. This was repeated for all words.

Neocognitron Algorithm

c_i output from C unit
s_i output from S unit
v output from V unit
w_i adjustable weight from C unit to S unit
w_0 adjustable weight from V unit to S unit
t_i fixed weight from C unit to V unit
u_i fixed weight from S unit to C unit
The signal sent by the inhibitory unit V is

$v = \sqrt{\sum\sum t_i c_i^2}$ Where the summations are over all unit that are connected to V in any array and over all arrays. The input layer is treated as the C_0 level

$$x = \frac{1-c}{1-vw_0} - 1$$

where $c = \sum_i c_i w_i$

is the net excitatory input from C unit and v w_0 is the net input from the V unit. The output signal is

$$s = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

the output of a C layer unit is a function of the net input it receives from all of the units, in all of the S arrays.
The net input is

$$c_{-in} = \sum_i s_i u_i$$

The output is

$$c = \begin{cases} \dfrac{c_{-in}}{a + c_{-in}} & \text{if } c_{\_in} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The parameter (a) depends on the level and is 0.25 for Level 1,2 and 1.0 for level 4.

6. Conclusions

A heuristic segmentation and recognition technique using Neocognitron net has been presented. Preliminary experiments were conducted on real-world handwritten words. The ANN-based segmentation technique produced very good results using a preliminary handwriting database. After segmentation the individual characters were used to train and test a further neural network. The classification of letters also produced some very encouraging results.

7. Recommendations

The following recommendation for further research can be identified:

1- The job above can be used for recognize Arabic numerals 0,1,…., 9. It will be more effective than Arabic word recognition.
2- If you need to know more information about Neocognitron net program and to benefit from it, please send a message to this e-mail abdulahnada@yahoo.com.

References

1- Ehsan N.; N. Mezghani ; A. Mitiche and R. d. B. Johnston (2003) "**Online Persian/Arabic Character Recognition by Polynomial Representation and aKohonen Network**", INRS- ´ Energie, Mat´eriaux et T´el´ecommunications, 800, de La Gaucheti`ere Ouest, Montreal (Qc), H5A 1K6, Canada, nourouzi*f*neila, mitiche.

2- Gheith A. A.; K. S. Younis and M. Z. Khedher (2008) "**Handwritten Arabic Character Recognition Using Multiple Classifiers Based On Letter Form**",Computer Engineering Department, University of Jordan, Amman 11942, Jordan, in Proc. 5th IASTED Int'l Conf. on Signal Processing, Pattern Recognition, & Applications, Innsbruck, Austria.

3- Khalid S.; M. Tab_dzki and M. Adamski (2003) "**A New Approach for Object-Feature Extract and Recognition",** Faculty of Computer Science Bialystok University of Technology, Wiejska 45A, 15-351 Bialystok, Poland.

4- Mostafa M. G. (2004) "**An Adaptive Algorithm for the Automatic Segmentation of Printed Arabic Text** ",Computer Science Department, Faculty of Computer Science, King Abdul Aziz University, Al-Madinah Al-Munawwarah, P.O. Box 344, Saudi Arabia, المؤتمر الوطني السابع عشر للحاسب الآلي (المعلوماتيةفـ يخدم ةضـ يوفـال رحمن)،جامع ةالمل كعبـ دالعزيز ،المدين ة المنورة .

5- Salama B. and Z. AL Aghbari (2008) "**Holistic Approach for Classifying and Retrieving Personal Arabic Handwritten Documents**", Department of Computer Science, University of Sharjah, P.O. Box 27272,UAE,P.565, 7th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES, (AIKED'08),University of Cambridge, UK.