

Features Extraction Method for Arabic Characters Based on Pixel Orientation Technique

MOHAMED A. ALI¹, KASMIRAN BIN JUMARI², SALINA ABD. SAMAD²
 Computer department¹, Elec., Electronics & System Engineering department²
 Faculty of Science, Fakulti Kejuruteraan
 Sebha University, Universiti Kebangsaan Malaysia
 LIBYA, MALAYSIA

Abstract: - This paper presents a features extraction module for isolated handwritten Arabic characters. The collected core features are based on pixels orientations according to Freeman chain code. The input to this module is Arabic character (in its basic-shapes i.e. without diacritics). The features extractor module, fed with a skeleton of an isolated character basic-shape, yields global and local features. Feature vector of 12 elements are used. Two features are global while the remaining 10 elements are locals. Neural network classifier is used for aggregating the features for classification decision making.

Key-Words: - Features extraction, Arabic handwritten recognition, Optical Character Recognition (OCR)

1 Introduction

Arabic Off-line handwriting character recognition has been a difficult problem to machine learning. It is hard to mimic human classification where specific writing features are utilized. Recent surveys have shown that present technology has still a long way to catch up in terms of robustness and accuracy [1]. Compared with machine-printed character recognition, the prime difficulty in the research and development of handwritten character recognition systems is in the variety of shape deformations [3].

Feature extraction module is one of stages in an Arabic optical character recognition system that we are developing. The main advantage of feature extraction module is that it removes redundancy information from the data and represents the character image by a set of numerical features. The features extractor module, fed with a binary image (skeleton) of an isolated basic-shape character, yields global and local features.

2 Features extraction strategies

Many different types of features extraction methods have been identified in the literature that may be used for numeral and character recognition [1]. Given that large number of feature extraction methods, a novice to the field is faced with the following question: Which feature extraction method is the best for a given application? This question has led us to characterize the available feature extraction methods, so that the most promising methods could be sorted out.

One could dispute that there is only a limited number of independent features that can be extracted from a character image, and hence whatever set of features being used is not so important. However, the extracted features must be invariant to the expected distortions and variations that the characters may have in a specific application.



In practice, the requirements of a good feature extraction method make selection of the best method for a given application a challenging task. It must also be considered whether the characters to be recognized have known size and orientation, whether they are machine printed, handwritten or hand-printed, and to what extent they are degraded.

Choice of feature extraction method, however, limits or dictates the nature and output of the preprocessing steps. Some feature extraction methods work on gray level images of single characters, while others work on solid 4/8-connected symbols segmented from binary raster image, thinned symbols/skeletons or symbol contours.

Each of these methods may be applied to one or more of the following representation forms; gray level character image, binary character image (solid and outer contour) and thinned character (skeleton).

2.1 Fourier descriptive-based feature extraction

The very first attempt was to use Fourier descriptors method as a features extractor since its output is very compatible with the classifier (Learning Vector

Quantization) which we planned to use in classification stage. Fourier descriptors are used to represent the skeleton of characters rather than the boundary. It has been noticed that Although Fourier Descriptors have many advantages, like exact representation of different segments that consisting a skeleton, they also have a number of disadvantages. One major drawback has to do with the detection of small spurs on the boundaries of characters. For instance, it is difficult to distinguish between the shape of two Arabic characters (seen - ) and (sheen - ). However, it must also be mentioned that this anticipated drawback might also be considered an advantage for filtering noise on the boundary. For these reasons we decided not to use it and, instead, use another structure-based feature extraction technique called stream-following method.

2.2 Structure-based Feature Extraction

In general, there are two approaches to structure-based features extraction; global and local approaches [4]. Global approaches analyze the character as a whole, whereas local approaches obtain features related to segments of that character. Global features may be more easily detected and are not as sensitive to local noise or distortions as are local features. Local approaches are usually more time-consuming than global ones but they provide more accurate verification. A method based on graphic representations "stream-following" is followed to achieve the structure-based features extraction.

2.2.1 Stream-following method

Our approach is similar to that of Nadler's approach [2]. His approach to the stream-following method is very simple. He used a 2x1 window to detect the lowest-level primitive. The 2x1 window scan the image from the left bottom to the top. Then the next scanning takes place overlapped one column to obtain information of connectivity of the given image. He used a graph that has four kinds of vertices, S (Start), E (End), C (Close), and O (Open). He called these vertices as primitives of a given image. For coding purposes, he used one more primitive N (Null). The naming of these primitives reflects the dynamic method of feature extraction based on the stream-following approach.

Similarly, our objective is to construct a graphic representation as shown in Fig. 1. Now we will describe how to obtain such graphic representation using (3x3 window) stream-following and different scanning orientation. Stream-following as we

mentioned is dynamic in nature, so the scanning order is quite important.

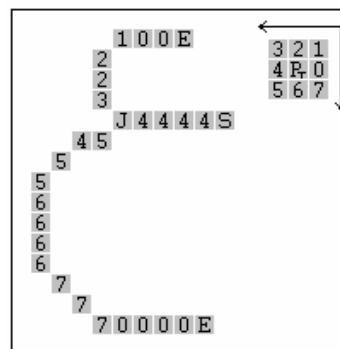


Fig. 1 scanning and recording of pixels directions

3 Features extraction module

Since we are dealing with Arabic script, we start scanning the image from top-right to bottom-left looking for the first black pixel. A Freeman's code based 3x3 window is used to trace all pixels consisting the character skeleton. The freeman code primitives can be described as follow; 0=East, 1=Northeast, 2=North, 3=Northwest, 4=West, 5=Southwest, 6=South and 7=Southeast as shown in Fig. 1.

However, empirically it has been realized that, in handwriting recognition, the similarity between a training set and an actual input stream is relatively weak and even in training set, the homogeneity in shape within a class is not so strong.

Using a high dimensional feature space is one of the solutions to deal with the poor homogeneity in order to add discriminatory power to a classifier [5-6]. Intuitively speaking, a high dimensional feature set which is selected in order to maximize recognition performance usually generates excessive separation in a 'good enough' input. In a cost-optimal approach, selection of features usually generates a trade-off problem between the computational and storage cost and the recognition performance.

A simple technique is used for feature selection in order to keep the features number as low as possible and yet they provide good representations.

In our feature extraction module, we used a feature vector of 12 elements. Two features are global (Black-to-White pixels ratio and aspect ratio) while the remaining 10 elements are locals (8 for chain code directions, one for number of junctions and one for number of loops). The utilized features are simple topological attributes. Yet, they carry enough

information to discriminate between the different characters.

The features extractor module, fed with a binary image (skeleton) of an isolated basic-shape character, yields global and local features. The local features are extracted from character images by applying a set of operators. The mechanism of each operator is explained as follow:

The first operator is a set of eight counters. Each one of these counters counts the direction transition of black pixel according to the Freeman chain code. Pixels tracing on the character skeleton in the image is from right to left as shown if Fig. 1. It has been noted that scanning the image from different prospects results different features sets. In other words, choosing different start-points and/or following different tracing routs on the skeleton will definitely yields different local features sets. To avoid this problem we impose some rules, which govern the pixel selection and tracing process:

- i. Since we are dealing with Arabic script, starting point is detected by scanning the image from top-right to bottom-left. Applying this type of scanning ensures that the detected starting point is the right most pixel on character skeleton.
- ii. During tracing, it generates a link list of directional codes. One special case is occurred when tracing reaches a junction point. From a junction point multiple directions are traceable as depicted in Fig. 2 In order to solve this problem a tracing priority is imposed. Priority of next move is given to the pixel at a direction of minimum value according to Freeman code. For instance if current pixel P_T has two neighboring pixels connected at direction (3 & 5) as shown below, according to this rule the next move will be into direction-3. Other route (direction-5) will be traced as soon as the algorithm exhausted tracing all the connected pixels in direction-3 and record their directional codes.

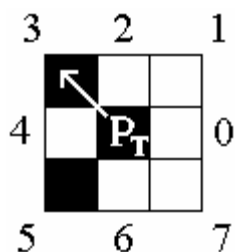


Fig. 2 Priority of tracing pixels

- iii. We assigned different colors for different pixels of interest (i.e. starting point pixel, end point

pixel, junction pixel, cross pixel ...etc.). This type of color coding helped the algorithm efficiently in analyzing the pixel under test and takes proper decision for the next step in skeleton tracing.

- iv. Pixel which is connected to only one pixel we name it as end pixel.
- v. Pixel connected to more than two pixels is assigned as a junction pixel P_j . A counter J is used for this purpose. This counter is used to count the number of junctions in each character image. This counter is the second local feature operator.
- vi. If either branch starts from a junction pixel is traced and ends at the same junction pixel we raise a flag that a loop is detected. A counter L is used for this purpose (loops counter). This counter is the third local feature operator. This counter is used to counts the number of loops in each character image. However, if it was not the case (i.e. it ends at an end pixel) then the tracing is restarted from the last junction it started from and starts tracing the other branch/s.
- vii. Traced pixels are turned from black to red color to prevent tracing a route more than once.

We apply these rules in both training and testing stages to ensure that the features extracted from a character image during the testing stage are identical to those extracted from the prototype of the same character during the training stage.

We enriched the local feature set with two global features giving information about the overall shape of the cursive character

The first global feature is an operator P_{BF} which is simply a counter that computes the ratio of the number of foreground pixels (black pixels) with respect to background pixels (white pixels) in the character image. Now if F_g and B_g are the numbers of foreground and background pixels respectively then:

$$P_{FB} = \frac{F_g}{B_g} \tag{1}$$

The second global feature is an operator that measures the aspect ratio (width/height ratio) of the character image. It is clearly shown in Fig. 3 how the aspect ratio widely differs from one character to another. As it goes from (a) to (c), aspect ratio is greater than one, almost equal to one and less than one respectively.

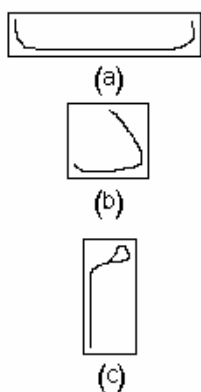


Fig. 3 aspect ratio is differ from one character to another

4 Test and result

Now by applying our features extraction algorithm on the character given in Fig. 1, the features obtained are given in the Table 1.

TABLE 1 Features extracted from character given in Fig. 1

E	NE	N	NW	W	SW	S	SE	No.J	No.L	A.Ratio	B/W Ratio
6	1	2	1	5	3	4	3	1	0	0.70	0.14

Practically, the features gathered from each input character image are appended to a data file "Feature.dat" shown in Fig. 4. The first line is reserved for number of features being extracted in this case it is "12" the second line started with #, it is simply comment which is ignored by the classifier. From fifth line and on, each line contains the following ordered data; Black-to-White pixels ratio, chain code counters (E, NE, N, NW, W, SW, S and SE), Number of Junctions, Number of Loops, Aspect Ratio and character Identifier. Character identifier is a character label (class) which could be any string, for instance a string "Ain-I" is given to the character shown in Fig. 1 to indicate that this character is "Isolated Ain". Character identifier is used only during the classifier training stage and, however, it is not needed during the test stage or real recognition stage. We define each row of data in this file as *code-vector*, so each character is represented by one code-vector. Dal-E, Baa-I, Faa-E, Ain-B, SenR-B, and Hha-M shown in Fig. 4 are character identifiers of Arabic basic-shape characters (ا ب ت ث ج د ه) respectively.

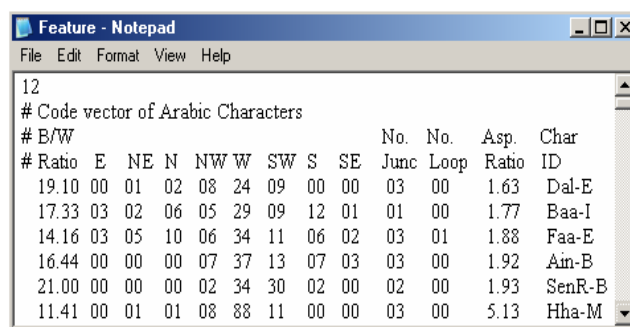


Fig. 4 A sample of data file contains features extracted from different Arabic characters

5 Conclusion

A reliable features extraction module based on pixels orientation is presented. The module utilizes two types of features; global and local features to optimally represent the character under process. The module, fed with a binary image (skeleton) of an isolated basic-shape character, yields global and local features. Although the total number of features extracted by this module is only 12 features yet these features give excellent representation of characters.

References:

- [1] Trier O. D., Jain A. K. and Taxt T., Feature extraction methods for character recognition – Survey, *Pattern Recognition*, Vol. 29, No. 4, 1996, pp. 641-662
- [2] Nadler, M., Sequentially-local picture operators. *2nd IJCP*. 1974, pp. 131-135.
- [3] Khorsheed, M.S. Clocksin, W.F., Spectral features for Arabic word recognition, *Proc. Inter. Conf. On Acoustics, Speech, and Signal Processing* Vol.6, 2000, pp. 3574-3577
- [4] Yu, D. Yan, H., Separation of single-touching handwritten numeral strings based on structural features, *Pattern Recognition*, Vol. 31, No. 12 1998, pp. 1835-1847.
- [5] Liwei W., Xiao W. and Jufu F., On image matrix based feature extraction algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 36, No. 1, 2006, pp. 194-197
- [6] Kenneth E. H., Deniz E., Kari T. and Jose C. P., Feature Extraction Using Information-Theoretic Learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 28, No. 9, 2006, pp.1385-1392.