ISSN 2278-7763

Noisy character recognition technique based on moments of inertiae

Faisel G. Mohammed[1], Wejdan A. Amer[2], Uhood S. Al-hasani[3]

[1&2&3](Computer Science Dep., Science College, Baghdad University, Baghdad, Iraq).
Email: [1]faiselgm73@gmail.com , [2]wejdan7449@yahoo.com, [3]uhooduhood@yahoo.com

ABSTRACT

In the current research work, improved and fast method was proposed to recognize the font character. This is achieved by learning and storing fonts (knowledge base). In this paper, the moment features for printed English character recognition are proposed. The moment descriptors have been developed as features in pattern recognition since the moment method was first introduced. Then, introduction of the character to make the comparison with the knowledge base. The noise was simulated into the data to approximate the real recognition. The creation of the noise made by the use of logical X-OR (Exclusive OR), namely the white pixel to be black and vice versa. To clean the noise incorporated three algorithms depending on which offer the speed and relative efficiency (Slow, Medium, Fast). After the comparison shows the character that found the system.

Keywords : Character recognition, Moment of inertia, Feature extractions, Noise removing.

1 INTRODUCTION

Optical Character Recognition Systems are getting more and more attention in recent decade. In many countries, OCR has been a part of their government sectors like post offices, Library automation, License Plate Recognition, Defense organization etc,. [1].

Optical Character Recognition is the process of translating images of handwritten, typewritten, or printed text into a format understood by machines for the purpose of editing, indexing/searching, and a reduction in storage size. There is a great need to accurately OCR printed materials: Much of the world information is held captive in hard-copy documents. OCR systems liberate this information by converting the text on paper into electronic form. Recognition is therefore defined as the task of text expressed in graphical format into its symbolic representation [2].

The standard steps of and OCR recognition is as follows [3], see fig 1:

1. Optical scanning for each character and the output will be BMP images.
2. Transforming the images into grayscale and binary using threshold process.
3. Preprocessing using thinning processing
4. Feature extraction using histogram processing
5. Recognition post-processing using neural network back propagation for training and testing process.

2 PROPOSED OCR TECHNIQUE DESCRIPTION

In this section we describe our proposed feature extraction method based on measurement of convexity of character strokes from different viewing directions. Figure 1 shows the system architecture of the proposed method.
a drop cap.

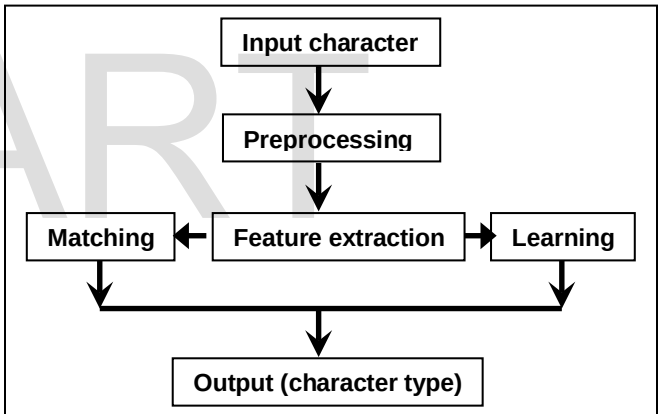

Fig. 1. The system architecture of the proposed method.

2.1 Preprocessing

Pre-processing is the preliminary step which transforms the data into a form that is more easily and effectively processed.

1. Binarization of the character
2. Noise Cleaning (3 models to remove noise were adopted. These models were showed in details in the following sections)

2.1.1 Binarization of the character

Image binarization is an important step for document image analysis and recognition. It convert an image up to 256 gray levels to black and white (1 and 0) images for which a threshold value. Every binarization algorithm gives variable results on different data sets.

Selection of appropriate binarization algorithm becomes very important for OCR performance [4].

ISSN 2278-7763

2.1.2 Noise Cleaning

The process of removing noise is a pre-processing step used in OCR system to improve accuracy of the result. The main task in preprocessing is to capture data and to decrease the noise that causes a reduction in the recognition rate and increases the complexities. Hence, preprocessing is an essential stage prior to feature extraction, as it controls the suitability of the results for the successive of the algorithm [5].

To clean the noise developed three algorithms that vary in time and performance (slow, medium, and fast). All three algorithms are based on a key feature of the characters, which is that the pixel which represents (part of) the character is very close together.

2.1.2.1 Slow Algorithm (Slow)

The slow control algorithm for each black pixel (black pixel we consider the value "1" in the corresponding two-dimensional array values) around the values and find The number of black pixel. So if a pixel has a round black predetermined number of pixel (Threshold) then this pixel is white ("zero" values in the table). So because they controlled all the neighbors, all black pixel algorithm, although very efficient even at very high noise is very slow and not recommended for slow computer systems (see Fig.2).

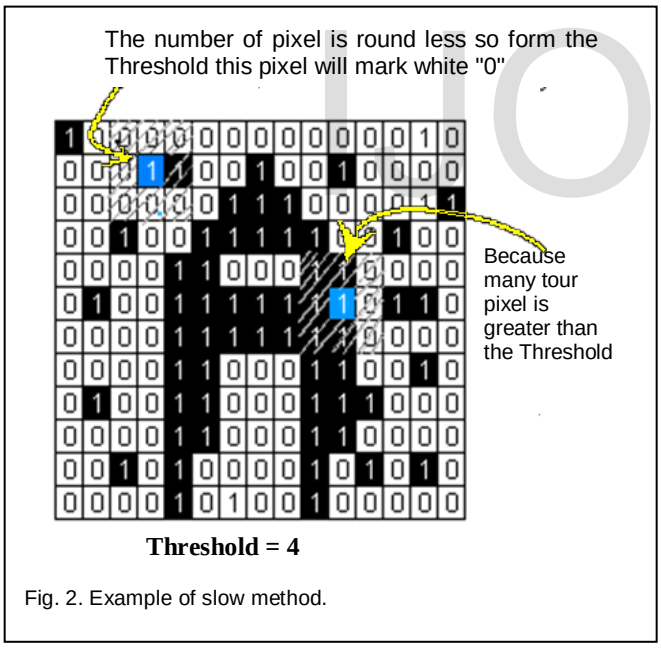

Fig. 2. Example of slow method.

2.1.2.2 Medium Algorithm (Medium)

This algorithm is similar to the previous algorithm, only the check is for a pixel group rather than each pixel separately (see Fig 3). So if a group of pixel the number of black pixel ("1") is less than the Threshold, then the whole team will be reset (it will be white), otherwise the whole team will remain unchanged.

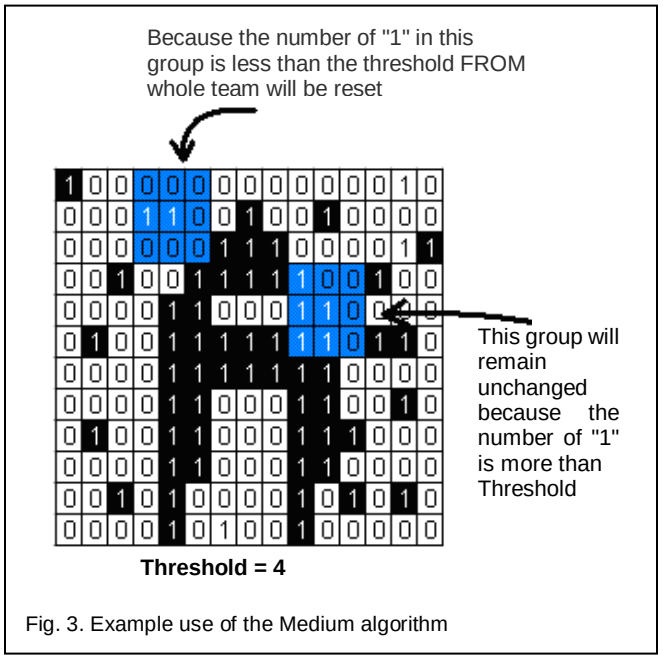

Fig. 3. Example use of the Medium algorithm

2.1.2.3 Fast algorithm (Fast)

In this algorithm, we achieve much better time cleaning data in relation to the previous one, but works with a maximum noise of 20%. For each black pixel of the character we find all the distances with neighbors (see Fig. 4 & Fig. 5) (the left, the right of the upper and lower neighbor). In this case we put a price Threshold, and so the pixel with less than the Threshold will be white ("0").

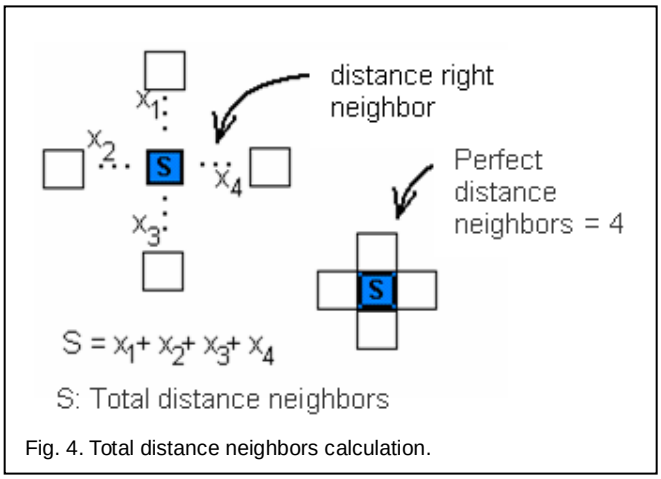

Fig. 4. Total distance neighbors calculation.

2.2 Feature extraction and learning

Extraction of potential feature is an important component of any recognition system. Selection of potential features is probably the single most important factor in achieving high recognition performance. In this paper, structural features considered as the potential features they are moment of inertia.

To teach initially use the "center of gravity of a flat surface, and the moments of inertia to find features to specific points of

ISSN 2278-7763

character which then stored so that learning can be done once.

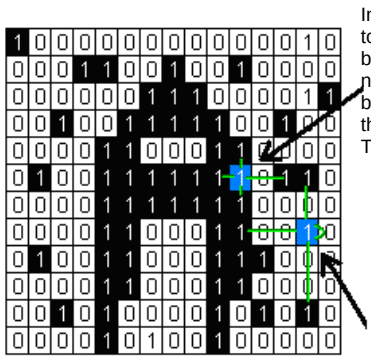

Fig. 5. Example of a total distance of two points.

2.2.1 Center for Weight Flatbed

Initially for each letter in the process of learning the center of gravity of the character. Thus we have the opportunity to identify the character even if it has moved, this is achieved because the characteristics of the torque is always calculated relative to the centroids of each character (fig. 6).

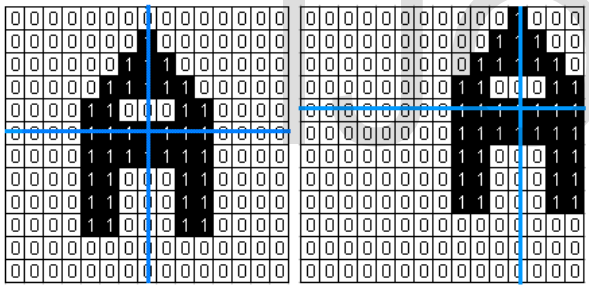

Fig. 6. The weight of the character "A" in two different positions of the character.

To calculate the center of gravity Si will be calculated coordinates Xc and Yc from the relationships:

$$X_c = \frac{\sum_{i=1}^{n}(Y_i - F_i)}{\sum_{i=1}^{n}(F_i)} \quad \text{.....(1)}$$

$$Y_c = \frac{\sum_{i=1}^{n}(X_i - F_i)}{\sum_{i=1}^{n}(F_i)} \quad \text{.....(2)}$$

2.2.2 Moments of Inertia

By calculating the moments of inertia calculate a representative profile for the character. The moments of inertia is calculated based on the center of gravity has been mentioned. So we will calculate the moment of inertia for the center-left weight, right on the bottom to the top and end for all data (see Fig 7). Overall it is necessary not to take the points from which passes the center of gravity as shown in the figure below, in fact it is better to integrate them into one of the two parties not to have incomplete data.

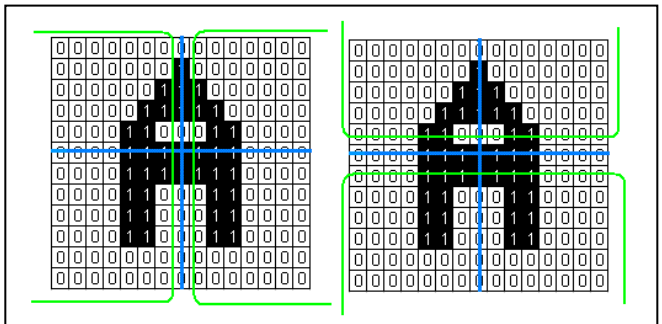

Fig. 7. The left picture shows the left and right of calculation and the right image of the lower and upper part of the calculation.

To calculate the moments of inertia J_x and J_y for a party using the following equation:

$$J_x = \int y^2 df \quad \text{.....(3)}$$

$$J_y = \int x^2 df \quad \text{.....(4)}$$

2.2.3 Classification minimum distance (metric Pythagoras)

The Pythagoras metric used to calculate the minimum distance between points (moments of inertia) of the character is entered by the user with points of all characters of a knowledge base. The minimum distance is calculated to represent the character trying to recognize. The formula of Pythagoras metric is:

$$Dj(x,y) = \sum_{i=1}^{n} |x_i - y_i| \quad \text{.....(5)}$$

3 Results and Conclusions

In this paper, OCR system is proposed. The proposed system uses moment of inertia and minimum distance metric (Pythagoras) classifier. The average recognition rate of numeral is 98.04%.

In any recognition process, the important steps to address the feature extraction and correct classification method [6].

The proposed method tries to address both the factors in terms of accuracy and time complexity. Figures 8, 9 and 10 shows an example of the output resulted from implementation of our proposed system.

ISSN 2278-7763

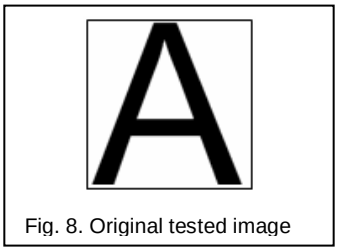

Fig. 8. Original tested image

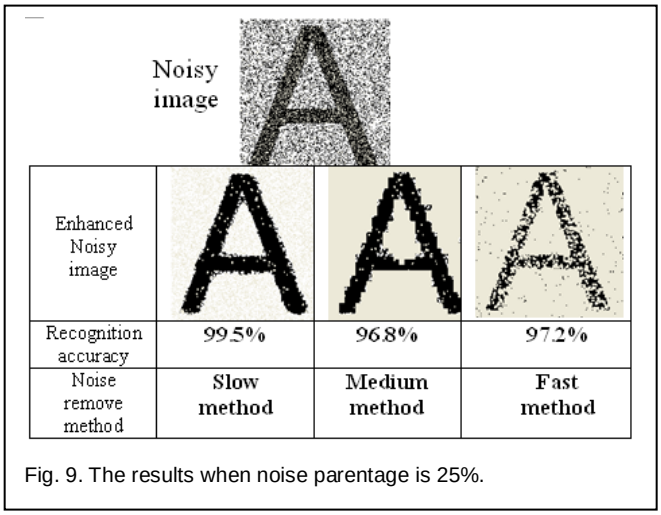

Fig. 9. The results when noise parentage is 25%.

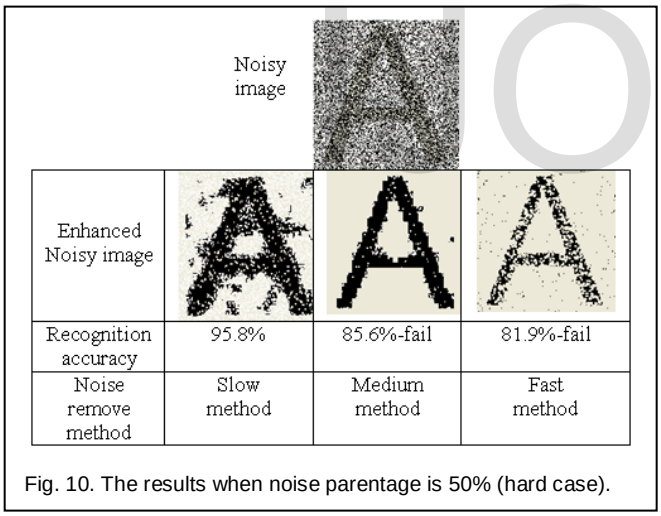

Fig. 10. The results when noise parentage is 50% (hard case).

REFERENCES

[1] Gunvantsinh Gohil1, Rekha Teraiya and Mahesh Goyani, "*Chain Code and Holistic Features Based OCR System for Printed DEVANAGARI script Using ANN and SVM*", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.1, January 2012.

[2] Ritesh Kapoor, Sonia Gupta & C.M. Sharma, "*Multi-Font/Size Character Recognition and Document Scanning*", International Journal of Computer Applications, P.p.0975–8887, Vol.23, No.1, Jun. 2011.

[3] Sari Dewi Budiwati, Joko Haryatno & Eddy Muntina Dharma," *Japanese Character (Kana) Pattern Recognition Application Using Neural Network*", IEEE, 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 17-19 Jul., 2011.

[4] Ravi Kant Yadav & Bireshwar Dass Mazumdar, "*Detection of Bold and Italic Character in Devanagari Script*", International Journal of Computer Applications (0975 – 8887) Volume 39– No.2, February 2012.

[5] K.S. Prasanna Kumar, et al, "*Optical Character Recognition (OCR) for Kannada numerals using Left Bottom 1/4th segment minimum features extraction*", Int.J. Computer Technology & Applications, Vol. 3, P.p., 221-225, IJCTA | JAN-FEB 2012.

[6] Dhandra B.V., Benne R.G. and Mallikarjun H. "*Printed and Handwritten Kannada numerals recognition using directional stroke density with KNN*", International Journal of Machine Intelligence ,Vol. 3, Issue 3, pp-121-125,2011.

