Eastern Arabic Handwritten Numerals Recognition

Marco Alfonse, Mohamed Almorsy, Mohamed Samir Barakat

***Abstract*— Character recognition plays an important role in the modern world. It can solve more complex problems and make human's job easier. An example is handwritten numerals recognition. This system can be used for many applications such as automatic recognition of the courtesy amount in bank cheques and recognition of postal code, as they require a high accuracy of recognition, and include many variations.**

In this paper, we have presented a new algorithm for segmenting connected Eastern Arabic numerals, a hybrid classifier to improve recognition accuracy. The hybrid classifier consists of Multilayer neural network and decision tree. The testing of the proposed recognition system indicates that the classification accuracy is in average 83%..

***Index Terms*— Character Recognition, Segmentation, Neural Networks, Classification.**

I. INTRODUCTION

Automated handwritten character recognition has increasingly gained popularity in the area of pattern recognition in recent years [1]. The need for fast processing of handwritten documents, due to the increasing amount of produced information, makes this area more important than ever. A number of research techniques for handwritten character recognition in various languages have been developed in recent years. Since features of the characters in a language may be quite different from those in other languages, methods for recognition depend on the languages involved.

This paper concentrates on Off-line Arabic handwritten numeral recognition which is the automatic transcription by computer, where only the image of that handwriting is available. Succession of operations in most of digital image recognition systems can be divided into three stages. The first one is the preprocessing, including thresholding, improving image quality (smoothing), segmentation and so on. The second stage is features extraction for avoiding data abundance and reducing its dimension. The Third stage is the classification.

II. THE PHASES OF HANDWRITTEN CHARACTER RECOGNITION SYSTEM

The handwritten character recognition system consists of the following stages [2] (figure 1):

Image acquisition: scanning the original image (converting it into a digitized form).

Extracting the region of interest: which consists of the numerals to be recognized.

Image preprocessing *"Conditioning"*: Performs operations that are required to enhance pattern image and make it ready for further processing stages.

Segmentation: separating numerals of given numerals' string using our proposed algorithms (image erosion, modified hybrid drop fall acid algorithm) to be used in feature extraction and recognition.

Feature extraction: extracting features from sub images that coming from segmentation stage to simplify recognize them.

Classification: determining the class of each sub image, a hybrid classifier is introduced.

A feedback between the segmentation and classification modules is used to allow the segmentation and recognition of connected numerals' string.

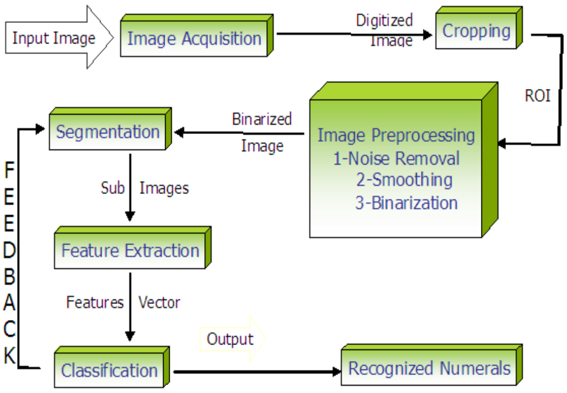


Figure 1: The phases of Handwritten Character Recognition System

A. Image Preprocessing

The preprocessing algorithms, techniques, and operators are used to perform initial processing that makes the primary data reduction and analysis task easier. The main task of preprocessing is the removal of artifacts from images, or the elimination of image information that are not required, and high-lighting the information required for the application [3][4].

Manuscript received August 30, 2009.

Marco Alfonse, Computer Science Department, Faculty of Computer & Information Sciences, Ain Shams University (e-mail: marco_alfonse@yahoo.com).

Mohamed Almorsy, Computer Science Department, Faculty of Computer & Information Sciences, Ain Shams University (e-mail: mohamed_almorsy@hotmail.com).

M. S. Barakat, Computer Science Department, Faculty of Computer & Information Sciences, Ain Shams University (e-mail: msamirhb@hotmail.com).

It includes noise removal, smoothing, and binarization.

The noise removal process concerns with enhancing the image by removing noise which is the byproduct of image acquisition. The smoothing process is to reduce the spikes and notches originated from the scanning noise. The Alpha-Trimmed filter [3][4] can be used to perform both noise removal and smoothing. The Alpha-trimmed filter is the average of the pixel values within the window (i.e. 3X3) but with some of the endpoints-ranked values excluded. The binarization process means converting image from grayscale to a black-and-white representation where black pixels represent the numerals to be recognized. The adaptive binarization is applied.

B. *Segmentation*

The segmentation process [5][6][7] is to divide the input numerals string into units that are suspected to represent individual numerals.

Segmentation problems:

1-Overlapping between characters :

Two numeral may overlap as shown in the figure 2, such problem that doesn't exist in English numerals.

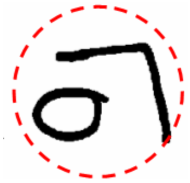


Figure 2: Overlapped characters

To solve this problem, we need to erode the image starting from the middle left to the right working with some sequence of black pixels recursively and grouping them as one character segment (as shown in figure 3).

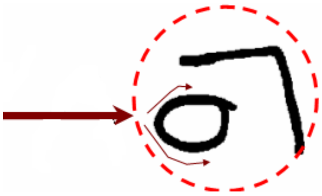


Figure 3: Region erosion

2- Connections between numerals:

Image erosion algorithm will simply detect numerals that are already segmented, another proposed algorithm will be applied on the connected numerals string (see figure 4) to separate it into digits. The proposed algorithm will proceed as follows: First, we start from the numeral middle left side and estimate the slope of the left side of the numeral (to determine whether to go up or down). Second, move with numeral edges using our rules(will be illustrated later) until reach its peak or bottom end, then start moving in the inverse direction until some bottle-neck point faced then starts cutting using average width and the slope of the line it is cutting. Figure 5 shows the image after applying the proposed segmentation algorithm for one iteration.

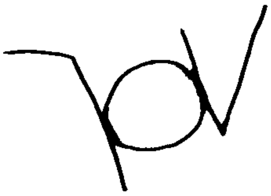


Figure 4: An example of a connected numeral string.

It is clear that image was taken as one numeral so now we will see the result of the application of the proposed algorithm on the output of the previous stage.

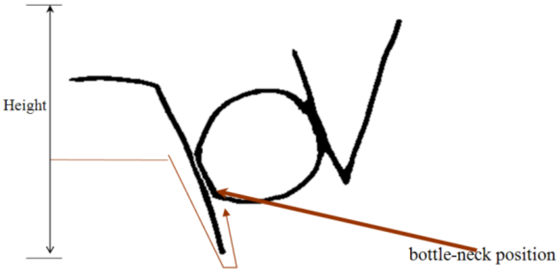


Figure 5: Image after applying the proposed segmentation algorithm.

This result will be applied to the image erosion algorithm. Again to detect that (6) Is one digit and (57) is another digit, and so on until no connected numerals will ever exist. The proposed algorithm is based on the classification module failure to recognize the output of the image erosion algorithm. In order to reduce the time of asking the classifier we apply the algorithm. First and then pass the results to the classifier to give its decision.

Movement rules for our team algorithm In pseudo code as follows:

If the slope of the left edge of the numeral has a negative slope as in (7) We will move down else move up as in(8).

A-For Moving down:

Rule 1: move right whenever right is white and upper is black.

Rule 2: move down-right whenever down-right is white and right is black.

Rule 3: move down whenever down is white and down-right is black.

Rule 4: move down-left whenever down-left is white and down is black.

Rule 5: move left whenever left is white and down-left is black.

After reaching the bottom of the numeral start moving up with the following rules :

Rule 1: move left whenever left is white and upper is black.

Rule 2: move up-left whenever upper-left is white and left is black.

Rule 3: move up whenever upper is white and up-left is black.

Rule 4: move up-right whenever up-right is white and up is black.

Rule 5: move right whenever right is white and up-right is black.

Rule 6: move up with considering slop of the line and average width of the areas on the two sides of the point to be cut.

B-For Moving Up:

Rule 1: move right whenever right is white and bottom is black.

Rule 2: move up-right whenever up-right is white and right is black.

Rule 3: move up whenever up is white and up-right is black.

Rule 4: move up-left whenever up-left is white and up is black.

Rule 5: move left whenever left is white and up-left is black.

After reaching the end of the image we then provide the output to the image erosion to detect the separated numerals and then pass it to the classifier to give the decision on the given numeral if accepted and recognized or not recognized(see chapter 7 for how to do so)which will activate this segmentor again.

So one benefit of such algorithm over the HDFA-algorithm is that we are sure to get the correct segment for recognition in contrast with winbank system that will require the operator to repair the cheque numerals when it's unrecognized.

C. *Feature Extraction*

Features to be extracted from the numerals will affect the accuracy of the recognizer, such features should be selected to be invariant to translation, scaling , rotation and writing styles(moments). One of the best available techniques to extract such features from the given numeral is Principle Component analysis technique (PCA), which finds an orthogonal set of directions in the input space and provides a way of finding the projections into these directions in an ordered fashion [8]. The orthogonal directions are called the eigenvectors of the correlation matrix of the input vector, and the projections the corresponding eigenvalues, so we can think of PCA as transfer of a set of correlated variables into a new set of uncorrelated variables, and map the data into a space of lower dimensionality. So we used PCA for such heavy task to be assured to get the features that will classify the numeral to its class without overlapping between classes. The PCA net will expect a matrix of (24 X 18) pixels of the numeral after being normalized to that size. Normalizing the image of the numerals so that they have the same impact on the PCA.

However the use of the features of PCA in feature extraction the classifier implemented by neural network has some drop in the classification accuracy (miss-classification) so we used another classifier (Decision Tree) based on some geometrical features extracted from the numeral

1-End points of the numerals.

2-Number of horizontal mid-line intersection with the numeral and their position

3- Number of vertical mid-line intersection with the numeral and their position

4- Numeral width of the numeral (necessary for zero).

These features are not completely reliable, and are used to test for the existence of value of such features of the output of the network numeral. The decision tree of such features will be illustrated in next section.

D. *Classification*

The classification based on a hybrid classifier of Neural-Network using output of PCA and Decision Tree using geometrical features extracted from numeral image(used as checker for NN output). The purpose of using hybrid classifier is to enhance the generalization accuracy of NN. Figure 6 shows the relationship between the classifier module and segmentation module.

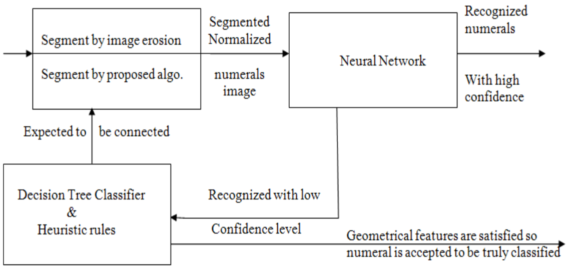


Figure 6: The relationship between the classifier module and segmentation module

From the previous figure it is clear that two classifiers have been used to implement classification module, now we will clarify each classifier, when it's used , and why we used a hybrid classifier.

First, MLP(multi-layer perceptrons): The output of the PCA used for features extraction, is fed into MLP for the classification of the given numeral to its class. This Classifier consists of two neural networks . the first network is a PCA network for feature extraction that its input is a 24 X 18 image and its output is 50 feature that passed to the second network that is a MLP neural network classifier that uses the input features to recognize the number. This Network trained by 4000 labeled patterns.

The parameters of this MLP are:

1- 0 hidden layers .
2- 10 output processing elements.
3- Step size is 0.1.
4- Momentum is 0.5.
5- The cost function is softmax

The network has some problems that result in poor performance which are wrong classification of numerals, The first problem is wrong classification for the input numeral , we solve this problem by passing this wrong recognized numeral to the second classifier.

The second problem is recognition with low confidence level and unrecognized (when the network decides that the given numeral is unrecognized as shown if figure 7). One solution to the first problem is to use another classifier to help clarifying the ambiguity in the numeral as varying in its shape (writing style). The solution to the second problem can be the same but to reduce time for classifying numeral we just test the numeral against our second classifier features that are expected to exist in the given numeral.

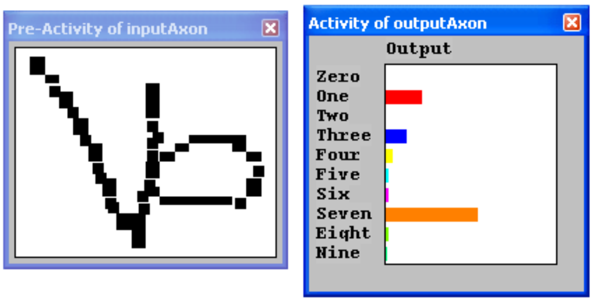


Figure 7: sample of unrecognized numerals by NN.

In figure 7, the network recognize the input number as seven however the number is seventy five, so we want the network to return unrecognized but the network can't do that alone (it must produce any output). This problem can be solved by using heuristic rules for thresholding the value of the neural network output for a given pattern these rules is stated below as.

First, "unrecognized" is issued when the max output (max1) of neural network is under some threshold which is selected by try and error (a=0.7,b=0.6) and not satisfied to geometrical tests as follows.

```
If  $max1 < a$  Then
//call Decision tree classifier
Signal unrecognized
Else If  $max1 > a$ And $max1*b < max2$  Then
//call Decision tree because network suspect in two values
Signal unrecognized.
Else
Signal numeral correctly recognized.
End if
```

Where :

a: used to determine the lower bound of confidence level.

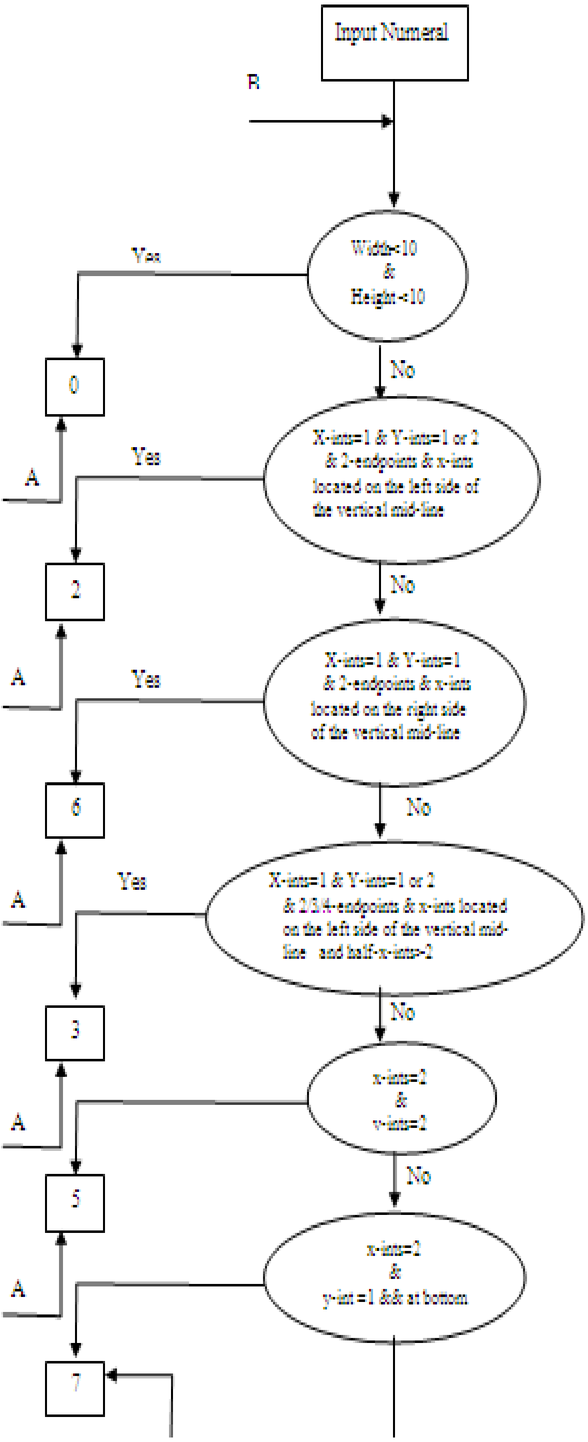
b: used to determine the degree of similarity between max. Two outputs

max1: is the max output of neural network when presented by the numeral image.

Max2: is the second max that is used to judge if the network suspect that the given numeral may be X or Y.

Second, Decision Tree classifier (figure 8) is used to improve the classification ratio of the Neural Network for wrong classification and assuring unrecognized numerals by NN to be more than one numeral by checking for the geometrical features of each possible number from zero to nine.

We use the decision tree classifier after network failure not directly to reduce time overheads incurred in going through such classifier, because before feeding the image to the Decision Tree classifier we need to do some preprocessing on the image to help in extracting the geometrical features of the image. This preprocessing is thinning of the image. The applied thinning algorithm is described in details in [9][10].



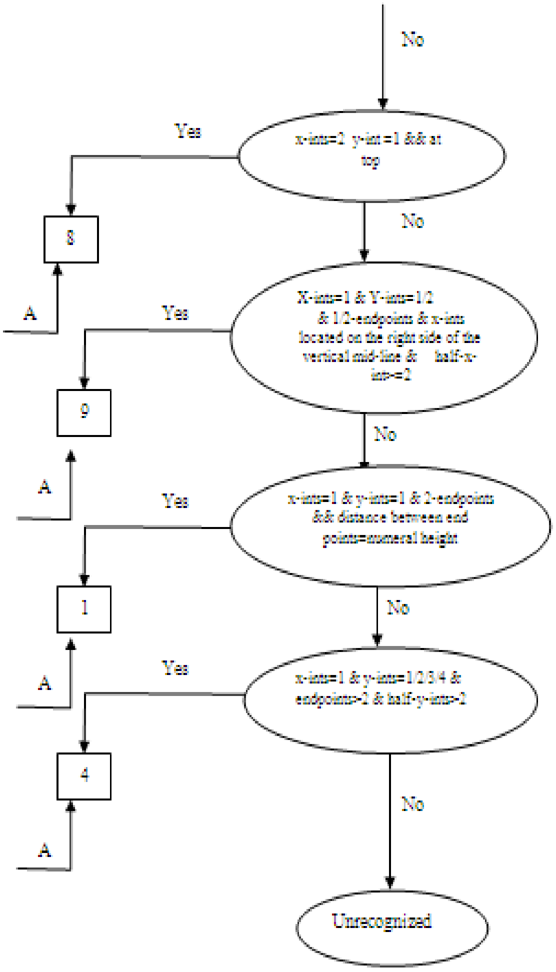

No
Yes
x-ints=2 y-int=1 && at top
No
8
A
Yes
X-ints=1 & Y-ints=1/2 & 1/2-endpoints & x-ints located on the right side of the vertical mid-line & half-x-ints<=2
9
No
A
Yes
x-ints=1 & y-ints=1 & 2-endpoints && distance between end points=numeral height
1
No
A
Yes
x-ints=1 & y-ints=1/2/3/4 & endpoints>2 & half-y-ints>2
4
A
No
Unrecognized

Figure 8: The Decision tree classifier

Where "A" represents an input image that need to be tested only for the existence of features for that numeral only, not needed to be classified (recognition with low confidence level, and unrecognized (when the network decide that the given numeral is unrecognized?). and "B" represents an input image need to be classified.

What's remaining to be illustrated is to identify when unrecognized is issued from the classification to the segmentor to start segment connected numerals and what rules are used to issue correct classified and test by decision tree. "unrecognized" is issued when geometrical tests are not satisfied as follows.

```
If num(max1) not satisfying geometrical conditions Then
If num(max2) not satisfying geometrical conditions Then
//go to segmentation module
Signal unrecognized.
Else
Signal num (max2) correctly recognized.
End if
Else
Signal num (max1) correctly recognized.
End if
```

Where :

a: used to determine the lower bound of confidence level.

b: used to determine the degree of similarity between max.

Two outputs

max1: is the max output of neural network when presented by the numeral image.

max2: is the second max that is used to judge if the network suspect that the given numeral may be X or Y.

III. Experimental Results

The results of testing the hybrid classifier module against segmented numerals from the training set and noisy data is illustrated table 1:

Table 1: Results of testing the NN

Numerals	Number of patterns	Correctly recognized patterns
0	50	33
1	50	40
2	50	44
3	50	40
4	50	41
5	50	44
6	50	43
7	50	45
8	50	45
9	50	41

The total number of patterns used for testing is 500, with 416 correctly classified patterns with accuracy of 83%. The system was tested against connected numerals results in 90% accuracy of the segmentation algorithm (correctly segment given numerals string into individual numerals).

IV. Conclusion

To recognize Arabic handwritten numerals, an initial segmentation of the image is performed (region erosion algorithm) and those digits are passed to the recognition module. If any digit is recognized with low confidence level, that digit is split using the proposed segmentation algorithm and recognition of the two resulting pieces is attempted. If one or both pieces are accepted, any remaining rejected pieces is feeded back to the segmentation algorithm. The usage of hybrid classifier and iterated segmentation- recognition had contributed significantly in overcoming the problems of connected and/or overlapped numerals.

References

[1] G N. Arica and F. Yarman-Vural, "An Overview of Character Recognition Focused on Offline Handwriting", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 31, No.2, pp. 216--233, 2001.

[2] J. Mantas, An Overview Of Character Recognition Methologies, Pattern Recognition, Vol. 19, No. 6, pp 425-430, 1986.

[3] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Prentice Hall; 3rd edition, August 31, 2007.

[4] Scott E Umbaugh, "Computer Vision and Image Processing: A Practical Approach Using CVIPtools", Prentice Hall, First Edition, 1998.

[5] Congedo, G., Dimauro, G., Impedovo, S., Pirlo, G., "Segmentation of Numeric Strings". Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. II 1038-1041, 1995.

[6] Blumenstein, M. and Verma, S., "A Neural Based Segmentation and Recognition Technique for Handwritten Words". IEEE International Conference on Neural Networks Vol. 3, 1738-1742, 1998.

[7] Lee, S.-W., Lee, D. J., Park, H. S., "A New Methodology for Gray-Scale Character Segmentation and Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (10) 1045-1050, 1996.

[8] Simon Haykin, "Neural Networks: A Comprehensive Foundation", Prentice Hall; 2nd edition, July 16, 1998.

[9] Nagendraprasad, M.V., Wang, P.S.P., Gupta, A., "Algorithms for Thinning and Rethickening Binary Digital Patterns". Digital Signal Processing 3, 97-102, 1993.

[10] Wang, P.S.P.and Zhang, Y. Y., "A Fast and Flexible Thinning Algorithm". IEEE Transactions on Computers 38 (5) 741-745, 1989.