Computing and Informatics, Vol. 26, 2007, 403–420

A NEW FEATURE EXTRACTION METHOD FOR TMNN-BASED ARABIC CHARACTER CLASSIFICATION

Khalid Saeed

Faculty of Computer Science, Bialystok Technical University
Bialystok, Poland
e-mail: aida@ii.pb.bialystok.pl

Majida AlBakoor

Faculty of Science, Aleppo University
Aleppo, Syria
e-mail: m-bakoor@scs-net.org

Manuscript received 3 March 2006; revised 14 December 2006
Communicated by Michael M. Richter

Abstract. This paper describes a hybrid method of typewritten Arabic character recognition by Toeplitz Matrices and Neural Networks (TMNN) applying a new technique for feature selecting and data mining. The suggested algorithm reduces the NN input data to only the most significant and essential-for-classification points. Four items are determined to resemble the distribution percentage of the essential feature points in each part of the extracted character image. Feature points are detected depending on a designed algorithm for this aim. This algorithm is of high performance and is intelligent enough to define the most significant points which satisfy the sufficient conditions to recognize almost all written fonts of Arabic characters. The number of essential feature points is reduced by at least 88 %. Calculations and data size are then consequently decreased in a high percentage. The authors achieved a recognition rate of 97.61 %. The obtained results have proved high accuracy, high speed and powerful classification.

Keywords: Arabic characters, Backpropagation neural networks, Toeplitz matrices

1 INTRODUCTION

In recent years, automation process of written and spoken text recognition has met new approaches, easy to implement for fast executing and higher success rate [1, 2]. This process should involve an interface facilitating fast communication between man and machine [3]. Statistical, structural and neural approaches are the most popular methods used to serve that aim [4, 5]. Character recognition is a notable field in text recognition, and its applications are very important for computerization procedures. Recognition of spoken and written texts is very often implemented successfully using neural networks as classifiers [6, 7]. Many recognition approaches are applied on Arabic and Latin scripts using feature extraction [8, 9]. Other methods for recognition of hand-printed characters are based on structural description and inductive logic programming as those given in [10]. In [11], the minimal eigenvalues of Toeplitz matrices are used to reduce the size of feature vectors by about 30 %. For the recognition of Arabic words without segmentation, Toeplitz models and their lowest eigenvalues have been developed and shown in [12] and [13].

This paper introduces a new method for Arabic character feature extraction and feature points reduction to the minimal possible number for the sake of the character recognition. Then for classification and recognition, an algorithm resulting from fusing Toeplitz matrices and backpropagation neural networks is applied [14]. The character image is filtered by the proposed 3×3 neighborhood median filtering for noise elimination and image smoothing [15, 16]. The image is thinned according to the algorithm in [11, 17]. The feature points are detected as branch, start or end points of the small lines constructing the character as will be shown in Figure 2. Therefore, the huge number of image pixels is reduced to only the most significant ones that are basic to enter with to the stage of classification. This is done by a new simple algorithm, which is designed for this purpose. The resulting feature vector from this algorithm enters the TMNN algorithm for classification and recognition.

2 ARABIC CHARACTER PROCESSING

This research focuses on recognizing the Arabic character written in Arial, Courier, Transparent and Simplified Arabic fonts of three different sizes (10, 12 and 14). The image in its *bitmap* (*.bmp) or *Tagged Image File Format* (*.tiff) is filtered up by eliminating the noise obtained from the scanner by using the supposed 3-by-3 neighborhood median filter for smoothing the image. The colored image is changed into a binary one, pixels of the image-filling are 1's while the background pixels are 0's. If a word is considered for recognition, it should be segmented into its letters before processing. Thinning process is used as a pre-processing stage to form a thin-line and to manipulate data freely and precisely. The image thinning and the skeletonization procedure followed in this work are performed by the algorithm KMM presented in detail in [11, 17]. Although the obtained shape is one-pixel width with continuous lines, it still keeps the important character characteristics. Feature extraction is one of the post-processing stages that provides an easier process for image matching.

This stage should therefore prepare the most significant and feasible characteristics from the image and deliver them as the basic data to the next stages for precise classification. Two ways are studied and executed on Arabic character classification: The first one is applied to all feature points, and the second only to the significant ones.

2.0.1 Neural Networks NN

The image is first thinned [11, 17]. Here, the coordinates of the feature points are represented in two vectors and considered as the input to the NN. The first vector is defined by the x-coordinates of the feature points, and the second is determined by the y-coordinates. The vectors are resized to have 64 elements each and scaled into the interval $[0, 1]$ to represent a reasonable input to the NN. The neural network will then contain 128 neurons in the input layer (coming from the two feature vectors, 64 each) and 500 neurons in the hidden layer. This number of neurons in the hidden layer was found experimentally. We have experienced 100, 200, 400 and 500 neurons. The highest performance was at 500 neurons. Moreover, the number of iterations was 884 for 100 neurons, while it was only 334 for 400 and a bit more (386) for 500 neurons, but with the best success rate. Table 1 shows how the rate of successful recognition clearly increases when increasing the number of neurons from 400 to 500.

	Letter recognition results %		
Number of neurons	ت (Simple 12)	د (Trans 14)	ر (Arial size 10)
400	82	82	97
500	96	98	98

Table 1. The behavior of NN classifiers for two different numbers of neurons in the hidden layer. The letters are selected from different experiments on varieties of fonts and sizes.

Now, the number of neurons in the output layer was chosen to be 18 neurons. This number, in turn, comes from reducing the number of Arabic alphabet letters, which is 28 (Table 2), to 18 classes (Table 3).

The reduction of letter classes is done by grouping the letters of exactly the same shape, but differing in the existence of 1, 2 or 3 dots above or under them (Table 2) into subclasses. Examples of such letters are the letters (ج, ح, خ), or (ب, ت, ث) and so on. The resulting group of letter-classes is shown in Table 3.

This reduction of classes increases the speed of the system significantly and hence decreases the performing time to a minimum.

The neural network is trained by a simple computer program by considering the momentum coefficient as 0.25, the learning rate as 0.05, and the sum square error as 0.00000585. Bipolar function is used as an activation function, and the weights

Sq.	Letter, Name	Isolated	End	Middle	First	Sq.	Letter, Name	Isolated	End	Middle	First
1	A, Alif	ا	ـا	ـا	ا	15	Dh, Dhad	ض	ـض	ـضـ	ضـ
2	B, Baa	ب	ـب	ـبـ	بـ	16	Tth, Ttaa	ط	ـط	ـطـ	طـ
3	T, Taa	ت	ـت	ـتـ	تـ	17	Zh, Zhaa	ظ	ـظ	ـظـ	ظـ
4	Th (θ), Thaa	ث	ـث	ـثـ	ثـ	18	Ea, Ain	ع	ـع	ـعـ	عـ
5	J, Jeem	ج	ـج	ـجـ	جـ	19	Gh, Ghain	غ	ـغ	ـغـ	غـ
6	Hh, Hhaa	ح	ـح	ـحـ	حـ	20	F, Faa	ف	ـف	ـفـ	فـ
7	Kh, Khaa	خ	ـخ	ـخـ	خـ	21	Qq, Qqaf	ق	ـق	ـقـ	قـ
8	D, Dal	د	ـد	ـد	د	22	K, Kaf	ک	ـک	ـکـ	کـ
9	Th, Thal	ذ	ـذ	ـذ	ذ	23	L, Lam	ل	ـل	ـلـ	لـ
10	R, Raa	ر	ـر	ـر	ر	24	M, Meem	م	ـم	ـمـ	مـ
11	Z, Zay	ز	ـز	ـز	ز	25	N, Noon	ن	ـن	ـنـ	نـ
12	S, Seen	س	ـس	ـسـ	سـ	26	H, Haa	ه	ـه	ـهـ	هـ
13	Sh, Sheen	ش	ـش	ـشـ	شـ	27	W, Waw	و	ـو	ـو	و
14	Ss, Ssad	ص	ـص	ـصـ	صـ	28	Y, Yaa	ي	ـي	ـيـ	يـ

Table 2. Arabic alphabet in its 28 main letters with each given in its four different possible forms [11]

are generated randomly. The general computations are performed in MATLAB. At a training time of 55 seconds with 386 cycles, the NN is trained on 168 characters of Courier font of multi size (10, 12, 14).

Letter, Name	A, Alif	B, Baa	J, Jeem	D, Dal	R, Raa	S, Seen
Equivalent Arabic letter	ا	ب	ج	د	ر	س
Letter, Name	Ss, Ssad	Tth, Ttaa	Ea, Ain	F, Faa	Qq, Qaf	K, Kaf
Equivalent Arabic letter	ص	ط	ع	ف	ق	ك
Letter, Name	L, Lam	M, Meem	N, Noon	H, Haa	W, Waw	Y, Yaa
Equivalent Arabic letter	ل	م	ن	ه	و	ي

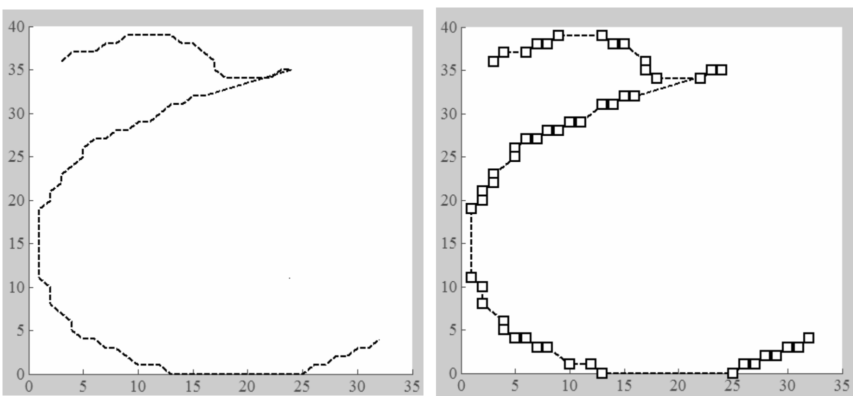
Table 3. The output of NN: 18 classes

2.0.2 Toeplitz Matrices and Neural Networks – *The Modified TMNN Hybrid Approach*

The TMNN algorithm is a hybrid method, which fuses Toeplitz Matrices minimal eigenvalues for image description and Neural Networks for image classification into one system [14]. In this work this method is modified to select only the essential points of the character image and form Toeplitz matrices from them. These points extraction should be performed carefully as they play the most important role in the precise object image description leading to right verification and hence successful recognition. After Toeplitz matrices are constructed, their minimal eigenvalues are constructed to define the feature vectors of the given object image. The percentage of points concentration for each part of the studied character is also considered to add another useful element to the resulting feature vector. These elements are the input to the NN.

2.1 Characteristic Points Extraction

After thinning the character image (Figure 1 a)) using the algorithm in [11, 17], the characteristic points are detected as illustrated in Figure 1 b). Such letter-image points are characterized as the start, end or branch points of the straight fragments (lines) in a character. The technique used in the selection and extraction is simple and depends on the sum of the 8-neighboring 0-1 pixels of the tested point [18]. This primary reduction leads to an elimination of at least 38.1 % of the letter image points.



a) b)

Fig. 1. The Arabic letter ح – HH after thinning (a) and characteristic points detection (b)

2.2 Characteristic Points Reduction to the Most Essential Ones (Essential Points)

The primarily detected points in Figure 1 b) are reduced further to the most significant ones (called essential points in this paper). The reduction steps are clarified by the flow chart in Figure 5. The reduction steps are accomplished according to the following algorithm.

2.2.1 Algorithm for Essential Points Evaluation

1. The characteristic points $a_i(x_i, y_i)$ $i = 1..n$ shown in Figure 1 b) are first found as the n pixels extracted from the thinned letter-image of Figure 1 a).
2. The central pixel is taken as a starting point $M(x_k, y_k)$. It is defined as the center of the extracted image with the coordinates x_k and y_k. Figure 2 shows the position of the central point.
3. The distance d_i between the centre and all the image pixels is calculated; $\|d_i\| = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$ for $i = 1..n$.
4. The point $a_i(x_i, y_i)$ is an essential point if one of the conditions R_1 or R_2 is satisfied: $R_1 = (d_{i+1} > d_i)$ and $(SN(d_{i+1} < d_i) > 2)$ and $R_2 = (d_{i+1} < d_i)$ and $(SN(d_{i+1} > d_i) > 2)$; SN is the number of the successive times where $(d_{i+1} < d_i)$ or $(d_{i+1} > d_i)$ are satisfied.
5. The steps from b. to d. are repeated with $M(x_i, y_i)$ in its new position. Now M is in the top left corner (the origin of the $x - y$ plane).
6. Again change the position of point M to have it this time at $X_k = 32$, $Y_k = 0$, the ending point of the image at the x-axis. The new distribution of vectors is shown in Figure 4.
7. The final set of essential points is the union of the sets of essential points determined at each step and hence the final feature vector will contain all the essential points collected from all individual vectors.

The reduction of the characteristic points has eliminated at least 81 % of the characteristic points. This procedure basically helps decrease the data size and the amount of computations. Figure 5 illustrates the flow-chart of the designed algorithm.

The importance of this reduction seems actual when the obtained results are as good as before the modification but with an array of less data size by such an average amount as 80 %. As an example, after the application of this algorithm, the Arabic letter – HH is represented with only 10 essential points now (Figure 6) out of 84 primary (after thinning) pixels (Figure 1 a)). Therefore, an elimination of 88.1 % of the letter image points is obtained (a reduction of 74 pixels) in this studied example. The letter ح in its last shape after this processing is shown in Figure 6. It is ready to further considerations for matching and classifications.

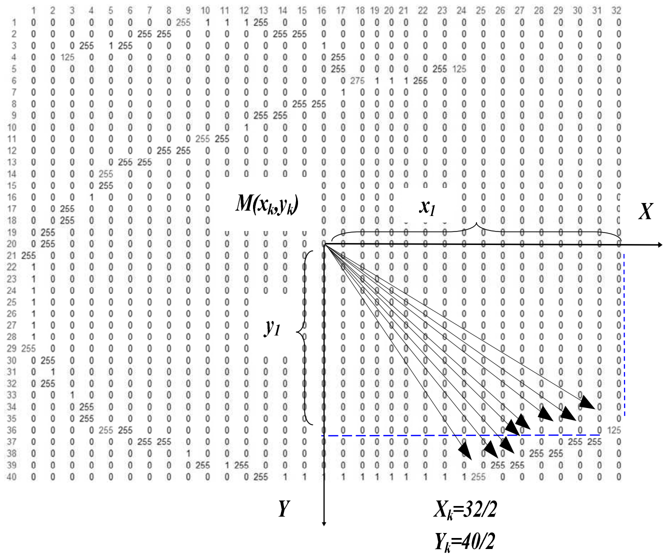


Fig. 2. The point M is the center of the image with the coordinates $X_k = 32/2$ and $Y_k = 40/2$ (total number of pixels on the x and y axes divided by 2, respectively)

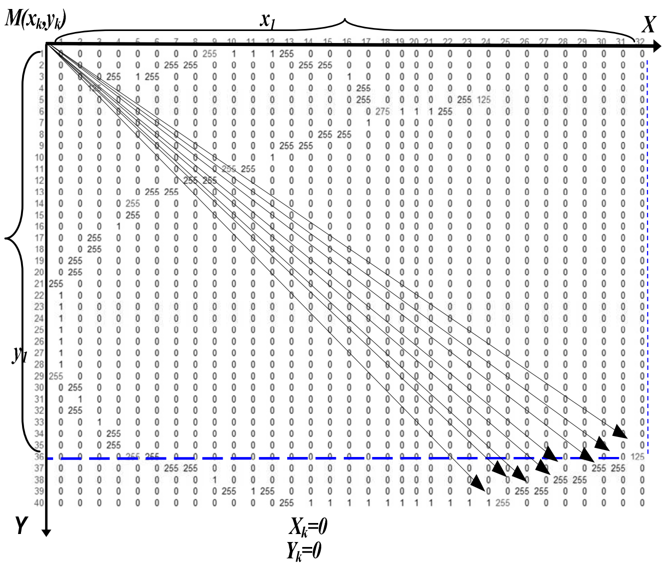


Fig. 3. The point M in its new position for further computations

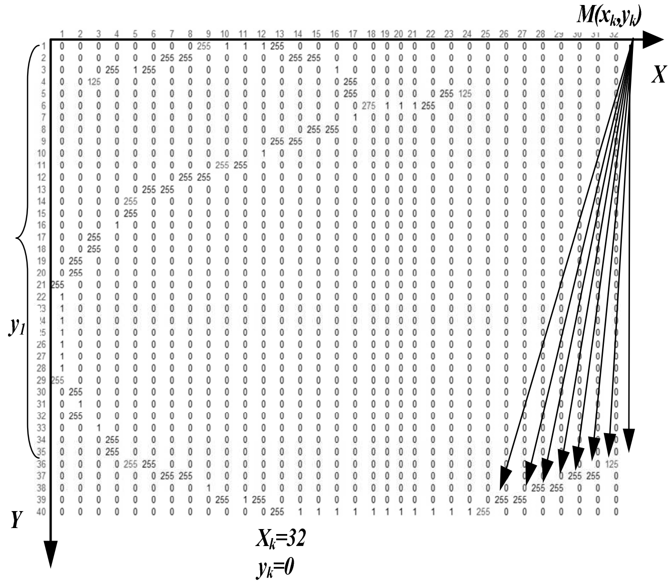


Fig. 4. Calculating the new vectors from M in its new position

Table 4 shows the results of data reduction for other Arabic letters and their recognition success rate.

2.3 Image-Vector Matching for Similarity and Classification

In order to obtain helpful information about the character, Toeplitz matrices are evaluated, and then their minimal eigenvalues are calculated. These values form the elements of the feature vectors which are used to describe the character for matching and classification. The description is done according to the algorithm of Toeplitz matrices and their eigenvalues. The details of this algorithm are beyond the topics of this paper but can be found in [11, 13, 19, 20, 21] for their applications in object description, voice-signal analysis and speaker identification. The proof of Toeplitz minimal eigenvalues invariance to both scaling and rotation when applied to script recognition is given in detail in [11]. The following simple code clarifies the steps of the feature vectors extraction according to the Toeplitz-based algorithm [11, 19, 20]:

```
for k12=1:k9-1
```

REM k9 is the number of essential points after completing the reduction to them

```
    dv=zeros(k12,k12);
```

REM dv is Toeplitz form with zero initial value. Its dimension increases by 1 in each loop after computing their corresponding eigenvalues

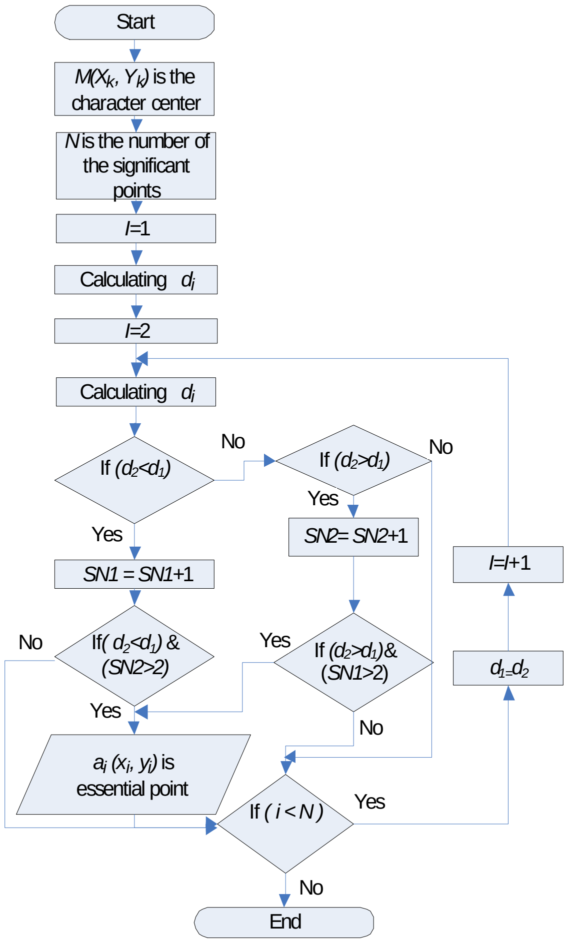


Fig. 5. The flow chart of the essential points verification

```
for k13=1:k12
    k14=k13;k15=1;
    while (k14<=k12)
        dv(k13,k14)= radiation(1,k15);
```

REM The elements of the array are the successive differences of the slopes (tan θ) at the points

```
        k14 =k14+1;
        k15 =k15+1;
    end
    k14=k13;k15=1;
```

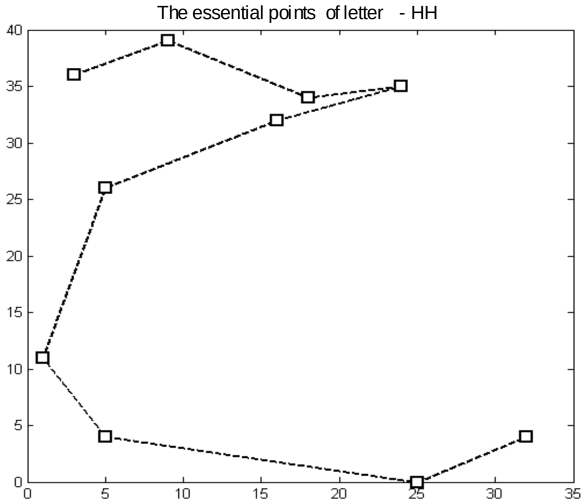


Fig. 6. The essential points of the Arabic letter letter ح – HH

Letter	Number of feature points after:			Recognition rate after last reduction %
	thinning (Fig. 1a)	first reduction (Fig. 1b)	last reduction (Fig. 6)	
alifA	34	10	3	91.18
Baa	67	27	6	91.04
Hhaa	84	52	12	85.71
Dal	46	21	7	84.78
Raa	55	26	6	89.09
Seen	114	54	4	87.72
Ssad	130	64	13	90.00
Ttaa	170	36	9	94.71
Ain	99	39	7	92.93
Faa	102	32	6	94.12
Qqaf	105	55	6	94.29
Kaf	84	29	5	94.05
Lam	86	24	4	95.35
Meem	90	37	4	95.56
Noon	84	36	5	94.05
Waw	88	39	6	93.18
Yaa	78	34	4	94.87

Table 4. The number of points after successive reduction and the recognition rate for different letters after the last reduction to only the essential points

```
    while (k14<=k12)
        dv(k14,k13)=radiation(1,k15);
```

REM The radiation is the minus tangent of the vector passing through the corresponding essential points. dv is Toeplitz matrix. v is the vector of the eigenvalues, and the second_min is the vector of minimal eigenvalues.

```
        k14 =k14+1;
        k15 =k15+1;
    end
end [d1,v]=eig(dv);fig1=10000000;
```

REM Compute the minimal Eigenvalues of Toeplitz matrices.

```
for k16=1:k12
    if v(k16,k16) <= fig1
        fig1= v(k16,k16);k17=k16;
    end
end
asecond_min(1,k12)=fig1;
end
```

Each vector is then resized to 20 elements according to the following simple code:

```
if n>20
         scanfile11 = imresize(scanfile1,[1 ,20], 'nearest');
         scanfile44 = imresize(scanfile4,[1 ,20], 'nearest');
     else
```

REM If the number of the vector elements is less than 20, then the remaining elements will be filled with zero values

```
         if n<20
                 scanfile11=scanfile1; scanfile44=scanfile4;
             for it1=n+1 :20
                scanfile11(1,it1)=0; scanfile44(1,it1)=0;
             end
         else
                 scanfile11=scanfile1; scanfile44=scanfile4;
         end
    end
```

Then the resulting vector is scaled into the interval $[-1, +1]$ to resemble the input of the neural network. The extracted character-image is divided into four parts, the distribution percentage of the essential feature points in each part is computed. The new four items are created and they are added to the input matrix to carry more information about the letter image. Then an array of 44 items is obtained and fed to the NN.

The matrix obtained from the feature vector, as described above, is considered as the input of the NN. The neural network contains 44 neurons in the input layer, 500 neurons in the hidden layer, and 18 neurons in the output layer. The training parameters of the NN are set up according to these values: the momentum coefficient is 0.25, the learning rate is 0.05 and the sum square error is 0.00000585. This NN uses bipolar function as an activation function. The NN weights and biases are generated randomly. The values range of the input vector is ranked between -1 to $+1$. Figure 7 shows the NN training.

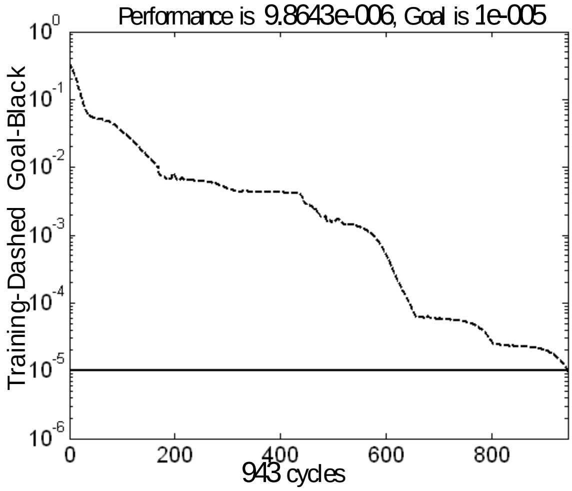


Fig. 7. The NN training on the Arabic character of Courier and Arial fonts

3 RESULTS AND DISCUSSION

Two methods are implemented for the classification. In the first, all feature points are used as an input to the NN. In the second method, however, NN and minimal eigenvalues of the Toeplitz matrices are applied to the essential points only. The results of comparison are given below.

3.1 Classification with NN only

The neural network is examined upon trained characters of Arial and Courier fonts at sizes 10, 12 and 14. The rate of recognition is 100 %. After the NN is trained, it is examined upon a new data according to these two stages:

First stage: the NN is tested on novel data which are 84 letters of Arabic Transparent font of multi size $(10, 12, 14)$. At size 10 of Arabic Transparent font, the rate of recognition is 100 %; the NN recognizes 28 letters out of 28 input ones. At size 12 and 14 of the Arabic Transparent font, the rate of recognition is 96.6 %; the NN recognizes 27 letters out of 28 input ones.

Second stage: the NN is tested on new data – 84 letters of Simplified Arabic font of sizes 10, 12 and 14. The NN recognizes 28 letters out of 28 input ones at size 10 of Simplified Arabic font while it recognizes 27 letters out of 28 input ones at sizes 12 and 14 of Simplified Arabic font. The rate of recognition is therefore 96.6 %.

3.2 Classification with the Hybrid System TMNN

An NN training is done using data composed of Arial and Courier fonts at sizes 10, 12 and 14. When the NN is tested on it, the rate of recognition is 100 %. Many experiments have been performed on novel data to classify the Arabic character:

At size 10 and 12 of the Arabic Transparent font, the NN recognizes 26 letters out of 28 input ones and the rate of recognition is therefore 92.9 %. At size 14, the NN recognizes 28 letters out of 28 input ones leading a recognition rate of 100 %.

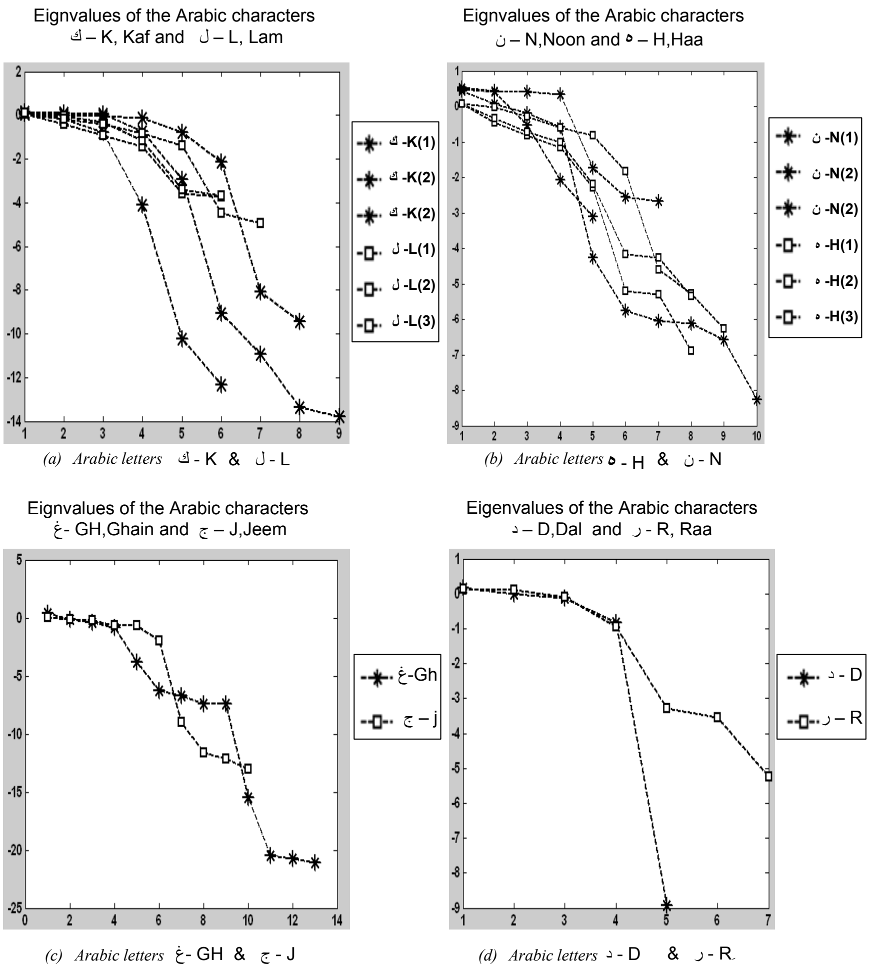
At sizes 10 and 12 of Simplified Arabic font, the NN recognizes 27 letters out of 28 input ones, and the rate of recognition is 96.4 %. At size 14, the NN recognizes 26 letters out of 28 input ones showing a 92.9 % rate of recognition. Therefore, the font change has no big effect on classification results.

In some cases, however, the results needed special improvement. For example, the Arabic letter ك – k is sometimes recognized as the Arabic letter ل – l, the Arabic letter ه – h as the letter ن – n, and also the Arabic letter غ – gh as the letter ح – hh or the Arabic letter د – d as the letter ر – r. These misclassification cases reduced the overall success rate and the system efficiency, which otherwise could be even 100 % for the machine written letters (see Table 5) for the results of recognition rate computations of such letters. It is worth mentioning that this problem faces most of the known algorithms of cursive letter identification [22, 23].

Character	ط	غ	ق	ك	د	ه	ي
Recognition rate	100%	91%	91%	83.3%	91%	75%	100%

Table 5. Results of classification using essential points only

The behavior of the minimal eigenvalues for the images of these letters is illustrated in Figure 8. ك is exactly like ل when removing the zigzag above it, ه is similar to ن when scaling the latter, the letter د always confuses with ر. When handwriting the last two letters, the reader always has a problem in distinguishing between them and usually the problem is solved linguistically. Mathematically, their feature vectors are $[0.13, 0.10, -0.08, -0.95, -3.29]$ and $[0.17, 0.01, -0.11, -0.82, -8.93]$ with their first five elements respectively. They are too similar to classify.



(a) Arabic letters ك - K & ل - L

(b) Arabic letters ه - H & ن - N

(c) Arabic letters غ- GH & ج - J

(d) Arabic letters د - D & ر - R

Fig. 8. Similarity in minimal eigenvalues of Arabic characters

The solution to the problem of misclassification in this work was, however, solved by the following simple technique based on Table 3 and the dots verification and counting. The dots, zigzags and other letter signs appearing in, above or below the letter body, are removed after regrouping from Table 2 to Table 3. The consideration of such signs will take place in an additional stage for subclass classification.

3.3 Comparison Between NN and Hybrid TMNN Classification

A comparison between the two approaches described in this paper (NN and TMNN) was made and introduced here. The input data to NN is 128 while it is 44 for TMNN.

The hybrid TMNN technique is based on reducing the number of inputs to the NN from 128 to 44. The cost of the calculations needed for the extraction of the set of essential points, construction of the Teplitz matrices and calculation of eigenvalues would be computed from the time consumed in both methods. The execution time of MATLAB performing in both of the NN and TMNN methods is almost the same. Hence, the input data to NN is about three times bigger than that fed to the TMNN for almost the same time of calculations.

The rate of recognition in NN approach was 99.4 % because of the perfect description of the character when considering all its pixels after thinning (Figure 1 a)). However, the time consumption and the big size of the data arrays are huge. On the other hand, the rate of recognition in the modified hybrid TMNN approach was 97.61 %, which really is a success and a great achievement with such minimal data within a negligible time. Despite the high reduction percentage (88.1 %), it was possible to reach such high success rate of recognition. The data reduction has led to less input data to the classifying algorithm and hence to less computations. The results are summarized in Table 6.

	Recognition rate%			
Method	Arial 10, 12, 14	Courier 10, 12, 14	Transparent 10, 12, 14	Simplified 10, 12, 14
NN	100	100	97.7	97.7
Hybrid	100	100	95.19	95.24

Table 6. Recognition percentage of the both NN and modified TMNN methods. The average of NN recognition is 99.4 % while that by the hybrid TMNN recognition is 97.61 %.

4 CONCLUSIONS

In this paper a Toeplitz matrices and NN based classifying hybrid system is introduced. The new approach differs from the other so far worked out systems in that the degree of data reduction is maximum. Only the essential points of the letter image in their minimal number are used to classify and recognize the Arabic character. Filtering the image up is used for smoothing it with the use of a neighborhood median filter. The image is thinned, features are selected and hence the feature vectors are defined by two approaches: NN method and the modified TMNN algorithm. The feature points coordinates form the elements of two feature vectors, one for the x-coordinates and the second for the y-coordinates of the extracted points. Then the vectors are resized, scaled to 128 elements and are considered as the input of the NN. The recognition rate of the classified characters is 99.4 %. The extraction of features in the modified TMNN is implemented to locate the characteristic points. Then the number of these selected points is reduced by a designed algorithm based on calculating the distance between a fixed point and the characteristic letter points. Two feature vectors are formed using Toeplitz matrices and their minimal eigenvalues. The percentage of the distributed points in each part of the studied character is considered as an additional vector. The resulting matrix is passed to the NN for recognition. The rate of recognition for classified characters was 97.61 %. The results of the comparison between the two methods is illustrated; the recognition rate has proved high in the second method, in spite of the reduction of feature points of at least 88.1 %. The current and near future work concerns the working on the hardware implementation of our algorithm and on the reduction of the hidden number of neurons. The MATLAB Toolbox is rich indeed, but if it comes to the minimal eigenvalues of Teplitz matrices – their computation and reduction still need some modification for faster systems. The number of hidden neurons is still high although it has given the best results of recognition and the highest rate of success. The authors are working on reducing the effective optimal number of hidden neurons from 500 to 100 without affecting the success rate, which has so far proved to be the highest at 500 neurons.

Acknowledgements

The authors are indebted to the anonymous referees for their constructive and valuable comments, and for their extremely helpful suggestions that improved the quality of the presented work and led to an extended version of the submitted paper. Our thanks are due to the Editor-in-Chief of CAI Dr. Ladislav Hluchý for giving us the opportunity to submit this work and to Dr. Ivana Jaššová, the CAI Editorial Assistant for her prompt response to our inquiries and her continuous help whenever needed.

This work was supported by the Rector of Bialystok Technical University – grant No. W/WI/10/07

REFERENCES

[1] Arica, N.—Yarman-Vural, F. T.: Optical Character Recognition for Cursive Handwriting. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, 2002, No. 6, pp. 801–813.

[2] Sarfraz, M.—Nawaz, S. N.—Al-Khuraidly, A.: Offline Arabic Text Recognition System. GMAG, pp. 30, International Conference on Geometric Modeling and Graphics (GMAG '03), 2003.

[3] Hussain, F.—Cowell, J.: Character Recognition of Arabic and Latin Scripts. Vol. IV, pp. 51, Fourth International Conference on Information Visualisation (IV '00), IEEE Computer Society, 19-21 July, 2000, London, UK.

[4] SANFELIU, A.: Progress in Pattern Recognition, Image Analysis and Applications. 9^{th} Iberoamerican Congress on Pattern Recognition, CIARP 2004, Puebla, Mexico, 2004.

[5] LOONEY, C. G.: Pattern Recognition Using Neural Networks. New York Oxford University Press, 1997.

[6] BAHI, H.—SELLAMI, M.: Combination of Vector Quantization and Hidden Markov Models for Arabic Speech Recognition. AICCSA, pp. 0096, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA '01), June 26–29, 2001., Lebanese American University, Beirut, Lebanon.

[7] SOMAYA ALMA'ADEED—COLIN, H.—DAVE, E.: Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach. ICPR, pp. 30481, 16^{th} International Conference on Pattern Recognition (ICPR '02), Vol. 3, August 2002.

[8] SALOUM, S. S.: Arabic Hand-written Text Recognition Using Critical Points. Sixth Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, Florida, USA 2002.

[9] TOUJ, S.—BEN AMARA, S.—AMIRI, H.: Global Feature Extraction of Off-Line Arabic. SMC 2002, Hammamet, Tunisie, October 2002.

[10] ADNAN AMIN: Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming. ICDAR, pp. 0333, Sixth International Conference on Document Analysis and Recognition (ICDAR '01), Seattle, September 10–13, Washington, USA, 2001.

[11] SAEED, K.: Image Analysis for Object Recognition. Bialystok Technical University Press, Bialystok, Poland, 2004.

[12] TABĘDZKI, M.—SAEED, K.: View-Based Word Recognition System. Proc. 4^{th} Intern. Conf. on Neural Networks and Artificial Intelligence – ICNNAI '06, Brest, Belarus, 2006, pp. 145–148.

[13] TABDZKI, M.—SAEED, K.: A View-Based Teplitz-Matrix-Supported System for Word Recognition without Segmentation. Proc. 6^{th} Intern. Conf. on Intelligent System Design and Applications – ISDA '06, Jinan, Shandong, China, 2006.

[14] SAEED, K.—TABEDZKI, M.: Cursive-Character Script Recognition using Toeplitz Model and Neural Networks. 7^{th} ICASC – International Conference on Artificial Intelligence and Soft Computing, June 7–11, Zakopane, Poland, 2004.

[15] GONZALEZ, R. C.—WOODS, R. E.: Digital Image Processing. 2^{nd} Edition, Prentice Hall, NJ, 2002.

[16] UMBAUGH, S. E.: Computer Imaging: Digital Image Analysis and Processing. CRC Press, FL, USA 2005.

[17] SAEED, K.—RYBNIK, M.—TABĘDZKI, M.: Implementation and Advanced Results on the Non-interrupted Skeletonization Algorithm. In: Skarbek, W. (Ed.), Lecture Notes in Computer Science – LNCS 2124, Springer-Verlag Heidelberg, Germany, 2001, pp. 601–609.

[18] GHUWAR, M.—SKARBEK, W.: Recognition of Arabic Characters – A Survey. Polish Academy of Science, Release No. 740, Warsaw, Poland, 1994.

[19] SAEED, K.: Computer Graphics and Analysis: A Method for Arbitrary Image Shape Description. Intern. Journal of Machine Graphics and Vision – MGV, Vol. 10, 2001,

No. 2, Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, pp. 185–194.

[20] Saeed, K.—Nammous, M.: A Simple Speech-and-Speaker Identification System. IEEE Trans. on Industrial Electronics, Vol. 54, 2007, No. 2, pp. 887–897.

[21] Tabędzki, M.—Saeed, K.: A View-Based Toeplitz-Matrix-Supported System for Word Recognition without Segmentation. Proc. 6th Intern. Conf. on Intelligent Systems Design and Applications – ISDA '06, Vol. 2, Jinan, Shandong, China, 2006, pp. 1207–1212.

[22] Cheung, A.—Bennamoun, M.—Bergman, N. W.: An Arabic Optical Character Recognition System Using Recognition-Based Segmentation. Pattern Recognition, Vol. 34, 2001, No. 2, pp. 215–233.

[23] Jain, A. K.—Duin, R. P. W.—Mao, J.: Statistical Pattern Recognition: A Review. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, 2000, No. 1, pp. 4–37.

Khalid Saeed graduated and received his B. Sc. degree in electrical engineering – electronics and communications in 1976 from Baghdad University, College of Engineering. He studied for his M. Sc. and Ph. D. in Poland in Wroclaw University of Technology at the Faculty of Electronics – Telecommunications and Acoustics (M. Sc. defended in 1978 and Ph. D. – distinguished in 1981). He worked with Al-Fateh University in 1981–1986 in Tripoli and Misurata and in the Higher Institute of Electrical Engineering of Zliten in 1987–1990. Now he works for Bialystok Technical University with the Computer Engineering Department in the Faculty of Computer Science, being the Head of Computer Engineering Department (1994–1999), Faculty Vice-Dean (1997–2000) and the Head of Research Team of Information Processing Systems (2000–present).

Majida AlBakoor graduated and received her B. Sc. degree in mathematical sciences – informatics in 1995 from Aleppo University, College of Sciences. She studied for her postgraduate degree at the same University (diploma in computer aided economics in 1996, diploma in mathematical sciences and informatics in 2002). Since then she has been working for her M. Sc. degree in informatics in Aleppo University. She is working for Informatics Department of Agriculture Management and also with the Intermediate Institute of Computer Engineering in the same city. She is the author of two papers published at International Conferences. She is a member of the Scientific Committee in Syrian Computer Society.