

## A NEW HYBRID SYSTEM FOR RECOGNITION OF HANDWRITTEN-SCRIPT

Khalid Saeed <sup>1)</sup> and Marek Tabędzki <sup>2)</sup>

Faculty of Computer Science, Bialystok University of Technology, Wiejska 45A, 15-351 Bialystok, Poland

<sup>1)</sup> e-mail: aidabt@ii.pb.bialystok.pl, <sup>2)</sup> e-mail: tabedzki@ii.pb.bialystok.pl

<http://aragorn.pb.bialystok.pl/~zspinfo/>

**Abstract:** A new method for object recognition and classification is presented in this paper. It merges two well-known and tested methods: neural networks and method of minimal eigenvalues. Each of these methods answers for a different part of recognition process. Method of minimal eigenvalues makes preparatory stage of analysis – of coordinates of characteristic points we get the vector describing given image. Next, it is recognized and classified with neural network. Gathering of characteristic points we perform with our view-based algorithm, but other methods should also do. In this work, method was applied for words in Latin alphabet – handwritten and machine-printed. The obtained results are promising.

**Keywords:** Object, Letter, word Recognition; Minimal Eigenvalues; Neural Networks

### 1. INTRODUCTION

In this paper, a new approach for capital Latin-letter classification and recognition is presented. This method merges neural networks [1] and algorithm of minimal eigenvalues of Toeplitz matrices [2], to take the advantages of these two methods.

It uses View-Based Approach [3, 4] for letter shape analysis. It examines four “views” of every letter and gathers nine uniformly distributed characteristic points of each view.

These values are treated as the coefficients of Taylor’s series, and used to create Toeplitz

matrices, according to the algorithm of minimal eigenvalues [2]. Then, for each matrix, the series of twenty minimal eigenvalues is calculated to describe the given letter. The series of these eigenvalues is the base for further classification with neural networks. A three-layered neural network is used and trained with the method of backpropagation.

As an example showing the testing of a set of letters, consider the letters constructing the word “SOFTWARE” written 125 times by different hands and ways. Fig.1 shows six selected samples of this handwritten word.



Fig.1 – Database sample

Word segmentation is not a part of this work, so we assume that the word is already separated using one of the known algorithms of word segmentation [5, 6, 7] to get separate characters for the sake of classification and recognition according to the presented in this work algorithm.

## 2. VIEW-BASED APPROACH

This method was firstly presented and fully described in our previous work [3, 4, 8]. Here, we repeat the main propositions of that work only for convenience.

This method is based on the fact, that for correct character-recognition a human usually needs only partial information about it – its shape and contour.

This method examines four “views” of each single script extracting from them a characteristic vector, which describes the given script or character. The view is a set of points that plot one of four projections of the object (top, bottom, left and right) – it consists of pixels belonging to the contour of the character and having extreme values of one of its coordinates. For example, the top view of a letter is a set of points having maximal  $y$  coordinate for a given  $x$  coordinate.

Next, characteristic points are marked out on the surface of each view to describe the shape of that view (Fig.2) The method of selecting these points and their number may vary from letter to another. In the considered examples, nine uniformly distributed characteristic points are taken for each view.

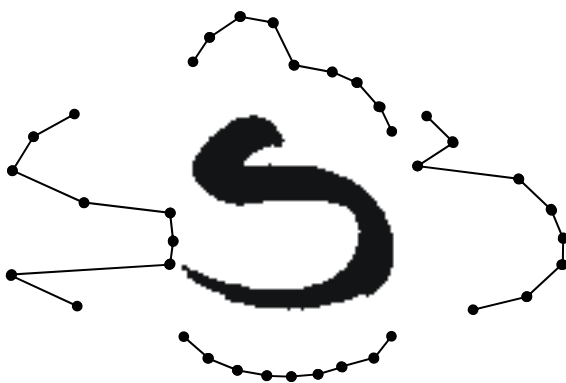


Fig.2 – Selecting characteristic points for four views

The next step is calculating the  $y$  coordinates for the points on the top and down views, and  $x$  coordinates for the points on left and right views. These quantities are normalized so that their values are in the range  $\langle 0, 1 \rangle$ .

Now, from 36 obtained values the characteristic vector is created to describe the given letter, and which is the base for further analysis and classification.

## 3. MINIMAL EIGENVALUES CRITERION

The normalized values are then considered as the coefficients of Taylor’s series:

$$T(p) = c_0 + c_1p + c_2p^2 + \dots + c_n p^n + \dots, \quad (1)$$

which are used to form Toeplitz matrices, according to the algorithm of minimal eigenvalues [2, 9]. Each of the four views is dealt with individually. Hence, we get four Taylor series and four Toeplitz matrix sets. Then the determinants of these matrices are formed (Eq.5).

$$M = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \dots \\ 0 & \alpha_0 & \alpha_1 & \dots \\ 0 & 0 & \alpha_0 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}, \quad (2)$$

$$M^* = \overline{M}^T = \begin{bmatrix} \overline{\alpha_0} & 0 & 0 & \dots \\ \overline{\alpha_1} & \overline{\alpha_0} & 0 & \dots \\ \overline{\alpha_2} & \overline{\alpha_1} & \overline{\alpha_0} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}, \quad (3)$$

$$[H] = \frac{1}{2}[M + \overline{M}] = \frac{1}{2} \begin{bmatrix} 2\Re\alpha_0 & \alpha_1 & \alpha_2 & \dots \\ \overline{\alpha_1} & 2\Re\alpha_0 & \alpha_1 & \dots \\ \overline{\alpha_2} & \overline{\alpha_1} & 2\Re\alpha_0 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}, \quad (4)$$

Assuming  $\alpha_0 = c_0, \alpha_n = 2c_n, c_{-n} = c_n, n = 1, 2, \dots$ . Then,

$$[H] = \begin{bmatrix} c_0 & c_1 & c_2 & \dots \\ c_{-1} & c_0 & c_1 & \dots \\ c_{-2} & c_{-1} & c_0 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}, \quad (5)$$

where  $[H]$  is the infinite Hermitian matrix and is of Toeplitz type. From these determinants, calculate the minimal eigenvalues  $\lambda_{\min} \{D_i\}$  for  $i = 0, 1, 2, \dots, n$ . Then, the following feature vector is formed:

$$F = \{\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_n\} \quad (6)$$

For every matrix, we calculate a series of twenty minimal eigenvalues, but for further analysis, we do not use all of them, but every fourth. Therefore, we get a vector of twenty elements describing the given letter, which is the base for further classification with neural network.

## 4. NEURAL NETWORKS

We use a three-layered neural network trained by the method of gradient descent with momentum

backpropagation [10, 11, 12]. As a transfer function, we take the hyperbolic tangent sigmoid function (Eq.7).

$$F(n) = \frac{2}{1 + e^{-2n}} - 1 \quad (7)$$

The 20-element vectors (obtained from the previous step) were presented on input of a network. In the output, we get a code indicating the tested letter. Each of the eight outputs stands for one of the examined letters. For the learning stage we use part of our data set (25 examples for each letter) to form our Learning Set. The remaining images are used as a Test Set – to verify the propriety of the learning phase. At the learning stage, every vector from Learning Set was presented to the network 100 times (in random order). After the learning stage, characters from Test Set were presented to the

network. The results were compared to the expected values, to find if they were classified right.

## 5. RESULTS

The size of the network has been chosen experimentally: a settlement between efficiency and accuracy was obtained for 8 and 15 neurons in the first and the second hidden layer respectively, but the results shown below take into consideration other performed experiments, too. The obtained effectiveness has achieved approximately 79% of correct letter-classification.

Table 1 presents the recognition results showing the effectiveness of the hybrid algorithm and its recognition rate for all tested characters using both approaches of the minimal eigenvalues of Toeplitz forms and the neural networks in their described above forms.

**Table 1. Results of recognition for the case of minimal eigenvalues with neural networks**

character	S	O	F	T	W	A	R	E
S	80.40%	9.20%	1.40%	0.40%	0.60%	1.20%	2.00%	4.80%
O	1.80%	91.80%	0.00%	0.00%	1.00%	0.80%	0.80%	3.80%
F	0.20%	0.40%	74.40%	6.80%	3.20%	2.20%	5.20%	7.60%
T	0.20%	0.00%	4.00%	94.80%	0.80%	0.00%	0.20%	0.00%
W	4.00%	12.60%	1.40%	4.00%	63.80%	8.60%	3.00%	2.60%
A	3.20%	0.40%	0.20%	0.00%	0.40%	94.60%	0.80%	0.40%
R	0.40%	3.40%	10.00%	2.00%	2.40%	3.40%	68.20%	10.20%
E	2.40%	5.20%	7.20%	2.00%	3.40%	2.20%	19.20%	58.40%

The table shows that the script **S** is recognized correctly (as **S**) in almost 80%, whereas incorrectly in 9.20% as **O** and in 1.40% as **F**, and so forth. The highlighted values are the correctly recognized scripts within this group of letters in one of the letter class tested. As mentioned before, the class considered in this work is that of the letters from the word SOFTWARE.

For comparison, experiments with the use of only neural network were also performed. Simply, the neural network was used for straightaway classification, without the stage of minimal eigenvalues calculation. In that instance, the neural

network has 36 inputs, which is the number of characteristic points we have considered. Other conditions remain the same.

Table 2 presents the complete results of the word SOFTWARE tested by neural networks only. As we can see the attained effectiveness in this case is much higher (90%) than the result when using the hybrid approach. Since the general hybrid algorithm has given high recognition rate [13], then these results are treated as an exception. That is why in the cases of low rate of recognition, the size of the network is increased as will be proved and shown in the next examples.

**Table 2. Recognition results for Neural Networks only**

character	S	O	F	T	W	A	R	E
S	91.40%	4.00%	0.20%	0.60%	0.00%	0.60%	0.20%	3.00%
O	0.20%	96.20%	0.00%	0.00%	0.20%	0.40%	2.20%	0.80%
F	1.00%	0.20%	78.40%	8.80%	0.60%	1.40%	4.00%	5.60%
T	0.40%	0.00%	3.40%	95.20%	0.00%	0.20%	0.60%	0.20%
W	0.20%	2.00%	0.00%	0.00%	97.20%	0.40%	0.20%	0.00%
A	2.80%	2.40%	0.00%	0.20%	1.40%	93.00%	0.20%	0.00%
R	0.00%	3.00%	0.60%	0.40%	0.20%	4.80%	90.60%	0.40%
E	1.40%	9.80%	2.40%	0.80%	0.40%	0.20%	7.00%	78.00%

Experiments with other letters have also been performed. The results are as good as those with SOFTWARE word letters.

## 6. OTHER EXAMPLES

### Example 1

For comparison, Table 3 shows the results of other experiments. This time classification was done

with classical methods – on the base of minimal Manhattan distance of vectors formed from the series of minimal eigenvalues of Toeplitz matrices. The learning set makes the base, with which the characters from the test set are compared. As can be seen, the rate of recognition is not as high as in the method with neural networks - it approximately reaches the rate of 70.88% for the correctly recognized characters.

**Table 3. Results of recognition for classical methods of classification**

character	S	O	F	T	W	A	R	E
S	83.00%	10.00%	0.00%	0.00%	1.00%	1.00%	5.00%	0.00%
O	1.00%	92.00%	0.00%	0.00%	0.00%	2.00%	3.00%	2.00%
F	0.00%	3.00%	54.00%	7.00%	4.00%	6.00%	6.00%	20.00%
T	0.00%	0.00%	10.00%	89.00%	0.00%	0.00%	0.00%	1.00%
W	4.00%	7.00%	15.00%	1.00%	50.00%	4.00%	3.00%	16.00%
A	2.00%	2.00%	0.00%	0.00%	0.00%	93.00%	2.00%	1.00%
R	2.00%	12.00%	7.00%	4.00%	3.00%	9.00%	56.00%	7.00%
E	3.00%	11.00%	8.00%	0.00%	1.00%	7.00%	20.00%	50.00%

### Example 2

This example is presented to check and prove the reasonability of our system parameters selection (such as the size of the neural network used or the size of Toeplitz matrices in the case of minimal eigenvalues approach).

The next two tables, namely Table 4 and Table 5, present the results of the experiments carried out for varieties of characteristic number of points in a view. Table 4 contains the results for 13 points per view while Table 5 contains 5 points per view. The series of minimal eigenvalues is still of the same size – 20 per view, from which 25% is selected. As seen, the choice of a bigger number of characteristic points has led to better results (81% correctly recognized scripts), whereas the decreasing of the number of points had led to worse results (only 61%

correctly recognized scripts). This result comes from the obvious fact that the greater the feature points are, the better is the script-description and hence the recognition is of higher rate.

Therefore, and because of this result and conclusion, the next experiments have been carried out with the number of characteristic points being set to 13. The cases, in which the increasing of the number of minimal eigenvalues may give worse results, again are an exception. The behavior of the minimal eigenvalues is similar to that of the neural networks discussed above in Section 5. Therefore, a similar conclusion can be drawn, at least for the so far experimental research results: The greater the number of minimal eigenvalues is, the better are the results.

**Table 4. Recognition rate with more characteristic points – 13 per view**

character	S	O	F	T	W	A	R	E
S	87.60%	1.80%	0.60%	0.00%	0.80%	1.40%	5.00%	2.80%
O	1.40%	90.40%	0.60%	0.00%	0.40%	0.00%	3.00%	4.20%
F	0.80%	1.60%	74.00%	2.80%	1.00%	3.40%	6.80%	9.60%
T	0.40%	0.00%	7.00%	89.40%	1.00%	0.20%	2.00%	0.00%
W	2.80%	8.80%	2.20%	2.20%	76.60%	1.60%	2.60%	3.20%
A	2.40%	0.80%	0.20%	0.20%	1.20%	93.80%	1.20%	0.20%
R	1.20%	4.60%	8.00%	2.40%	3.00%	3.80%	68.20%	8.80%
E	1.60%	5.40%	13.80%	0.60%	0.80%	1.60%	10.00%	66.20%

**Table 5. Recognition rate with less characteristic points – 5 per view**

character	S	O	F	T	W	A	R	E
S	73.20%	8.20%	1.00%	1.80%	1.80%	5.00%	5.60%	3.40%
O	2.00%	81.80%	2.40%	0.00%	2.60%	0.40%	8.60%	2.20%
F	0.00%	3.00%	41.00%	9.40%	11.00%	2.00%	13.00%	20.60%
T	0.00%	0.20%	20.60%	68.80%	1.60%	0.00%	3.00%	5.80%
W	1.80%	8.80%	18.60%	0.80%	53.20%	8.00%	2.60%	6.20%
A	13.40%	4.40%	2.00%	0.60%	2.00%	75.20%	2.20%	0.20%
R	5.20%	7.40%	11.80%	2.60%	3.20%	3.80%	58.40%	7.60%
E	1.60%	6.40%	23.80%	2.20%	11.00%	4.60%	16.00%	34.40%

**Example 3**

Table 6 presents the results of experiments performed for neural network with 13 and 23 neurons in first and second hidden layer respectively, former experiments were performed for neural network with fewer number of neurons (with 6 and 12 neurons in the first and second hidden layers, respectively.) A bigger network may give better results, but also affects efficiency – it takes more time to train a bigger network.

The number of *correctly recognized letters* was also calculated; it is 82.12%. As can be seen, the obtained results are slightly higher, when compared to former examples. The size of the learning set should be adjusted to the size of the network – there are formulas defining this dependence [14]. However, the choice of the optimal size of networks

and learning sets is not a part of this publication and is left to a future discussion. It should be noticed, however, that the number of the neurons in hidden layers depends strictly on the number of classes to classify – it is possible to determine the maximum number of the neurons necessary to recognize the given number of classes [15]. Enlarging the network will not result necessarily in improved effects. However, increasing the number of neurons in hidden layers is essential when considering larger number of scripts, the whole alphabet, for example. In future works the authors will introduce the results of experiments with the whole alphabet. This is done because of the increased number of classes to recognize and hence the increase in the size of the learning set.

**Table 6. Results of recognition for bigger networks – 13 and 23 neurons**

character	S	O	F	T	W	A	R	E
S	89.68%	1.08%	0.22%	0.00%	1.72%	0.86%	2.15%	4.30%
O	1.94%	87.74%	1.29%	0.00%	2.37%	0.22%	3.44%	3.01%
F	0.65%	0.43%	79.14%	2.80%	0.43%	0.65%	7.74%	8.17%
T	0.00%	0.00%	3.23%	95.27%	0.65%	0.00%	0.43%	0.43%
W	1.08%	2.58%	3.44%	1.94%	84.30%	2.80%	3.01%	0.86%
A	1.29%	0.00%	0.65%	0.00%	1.72%	94.41%	1.72%	0.22%
R	1.29%	4.30%	9.03%	2.80%	4.30%	6.67%	66.88%	4.73%
E	3.87%	6.67%	10.32%	0.00%	1.08%	0.86%	17.63%	59.57%

The next two examples concern bigger networks.

**Example 4**

In this example, the use of the minimal eigenvalues approach is slightly altered. As noticed in Section 3 above (“Minimal Eigenvalues Criterion”) for each of the four views, the series of 20 minimal eigenvalues was computed, but for further analysis, only every fourth of these values was used and these twenty values were forming the characteristic vector describing each letter. Now for each of the four views of a letter 30 minimal eigenvalues are calculated instead of 20. It gives a

series of 120 values for each letter. Table 7 shows the results obtained for an input vector composed of 20 of these values (5 per view). It gives approximately 81% *properly recognized letters*. Comparing these results with the former examples, the modified method’s effectiveness is lower. Consequently, a further extending of the minimal eigenvalues of Toeplitz forms does not necessarily improve the results, if still considering the same size of the input data.

**Table 7. Recognition rate using 20 of 120 minimal eigenvalues**

character	S	O	F	T	W	A	R	E
S	83.66%	4.30%	0.00%	0.00%	0.86%	1.51%	3.44%	6.24%
O	1.29%	90.75%	0.43%	0.00%	1.51%	1.08%	0.22%	4.73%
F	0.00%	0.65%	68.39%	3.87%	2.58%	0.65%	5.38%	18.49%
T	0.22%	0.00%	3.66%	94.84%	0.86%	0.00%	0.00%	0.43%
W	2.80%	1.94%	0.65%	1.94%	86.24%	4.30%	1.08%	1.08%
A	4.73%	0.00%	0.43%	0.00%	2.80%	90.54%	1.29%	0.22%
R	2.80%	7.10%	4.30%	1.51%	1.94%	4.95%	68.39%	9.03%
E	3.44%	4.73%	10.54%	0.00%	1.29%	2.15%	14.41%	63.44%

**Example 5**

In this example (Table 8), the results for a bigger input vector are given. We have made use of 40 out of 80 calculated minimal eigenvalues (10 per view). As we can see, the attained effectiveness is greater than that in the first example (Table 1) – 87.21% of correctly recognized letters. It means that the

increase of the input vector elements may lead to improvement of effectiveness. Of course, it also affects the performance as it leads to its decreasing. In case we need high precision and do not care about the computing time, we may even use all of the 120 calculated minimal eigenvalues.

**Table 8. Recognition rate for the case of 40 of 80 minimal eigenvalues**

character	S	O	F	T	W	A	R	E
S	96.71%	1.41%	0.47%	0.24%	0.00%	0.00%	0.47%	0.71%
O	1.65%	90.59%	0.24%	0.00%	0.47%	0.00%	2.59%	4.47%
F	0.00%	0.00%	88.00%	2.12%	0.94%	0.94%	4.71%	3.29%
T	0.00%	0.00%	3.06%	95.53%	0.71%	0.24%	0.47%	0.00%
W	1.65%	2.59%	1.88%	2.59%	87.53%	1.18%	1.65%	0.94%
A	3.06%	1.18%	0.71%	0.00%	1.88%	89.88%	3.29%	0.00%
R	2.12%	2.35%	6.12%	3.29%	1.41%	3.06%	74.82%	6.82%
E	3.29%	4.00%	7.53%	0.00%	0.94%	0.47%	9.18%	74.59%

**7. WORD RECOGNITION**

Experiments were made on word recognition making use of the same letters in the same word SOFTWARE but written in several styles (125 different ones). The percentage of correctly recognized words (words with each of 8 letters correctly recognized) is still a bit low and hence the approach is still under modification.

**8. CONCLUSIONS AND FUTURE WORK**

In this work, a trial has been made to apply a hybrid system for Handwritten Latin script recognition and identification. The obtained results have shown that the system is as successful as it is when applied to printed scripts [13]. It only needs some more experimenting on word recognition in order to generalize the approach. Although the word recognition is not the topic of this paper, the current work has shown promising results. The methods of script identification forming the hybrid system presented in this work have proved their high performance in image recognition. The best results

are obtained when using both of the minimal eigenvalues and the Neural Network approaches. However, the approach of minimal eigenvalues of Toeplitz matrices requires more experiments to find better ways of utilizing its advantages, when applied in this approach, in a similar way of using it in the known methods of classification and recognition of other images [2, 3, 4, 8, 9]. Future work will definitely concern *word recognition* written in Latin and other alphabet scripts. This will cover both printed and handwritten words. In fact, the authors have already started working on word recognition, as mentioned in Section 7 above, and the results are promising to extend. The rate of correct recognition is still not very high. This comes from the fact that the rate of recognition is calculated in such a way that if one of the scripts of a word is not recognized, then the whole word is said to be misclassified and hence unrecognized even though the rest of the word letters are fully recognized. Therefore, under such assumptions, a 70% rate of word recognition, for example, is actually very high. It means we are dealing with more than 90% rate of letter

recognition. Word Recognition is one of the important topics in Language Recognition in both cases of spoken and written expressions. The authors and their research group have already published a number of papers on these aspects, but in a limited kind of words. One work is a trial on written Arabic word recognition without segmentation [16] and the other is on spoken digits, letters and color names in two languages, Polish and English [17]. Spoken Arabic language, however, is more difficult and is still under experimenting. It needs some more modifications of the tools applied in the stage of image description. Therefore, the modification of our mathematical tools in both neural and minimal eigenvalues approaches will always strengthen the mathematical models we are using and hence they will definitely lead to better results. Some new modifications have already led to better results, which have been sent for publication [18].

### ACKNOWLEDGEMENT

This work is an extension with modification of about 60% of the paper presented in ICNNAI'03 – Third International Conference on Neural Networks and Artificial Intelligence conducted in Minsk in November 2003.

The work was supported by the Rector of Bialystok University of Technology (grant number W/II/3/01).

### 9. REFERENCES

- [1] B. Widrow. R. Winter. R. Baxter. *Layered Neural Nets for Pattern Recognition*. IEEE Trans. ASSP, 1988, Vol. 36, pp. 1109-1118.
- [2] K. Saeed. *Computer Graphics Analysis: A Method for Arbitrary Image Shape Description*. Volume 10, Number 2, 2001, pp. 185-194, MGV - International Journal on Machine Graphics and Vision, Institute of Computer Science. Polish Academy of Sciences. Warsaw 2001.
- [3] K. Saeed. M. Tabedzki. M. Adamski. A New Approach for Object-Feature Extract and Recognition. *9<sup>th</sup> International Conference on Advanced Computer Systems – ACS'02*. Miedzyzdroje, 23-25 October 2002. pp. 389-397.
- [4] K. Saeed. Object Classification and Recognition Using Toeplitz Matrices. *Artificial Intelligence and Security in Computing Systems*. Edited by Jerzy Soldek and Leszek Drobiazgiewicz. The Kluwer International Series in Engineering and Computer Science, Volume 752. September 2003.
- [5] R. C. Gonzalez and R. E Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992. pp. 7-9, 413-414.
- [6] K. Bowyer and N. Ahuja. *Advances in Image Understanding*. IEEE Computer Society Press, 1996. p. 65.
- [7] V. F. Leavers. *Shape Detection in Computer Vision Using the Hough Transform*. Springer-Verlag, 1992.
- [8] M. Rybnik. A. Chebira. K. Madani. K. Saeed. M. Tabedzki. M. Adamski. A Hybrid Neural-Based Information-Processing Approach Combining a View-Based Feature Extractor and a Treelike Intelligent Classifier. *CISIM – Computer Information Systems and Industrial Management Applications*. WSFiZ Press. Bialystok, 2003.
- [9] K. Saeed. A New Approach in Image Classification. *Proc. 5<sup>th</sup> International Conference on Digital Signal Processing and its Applications - DSPA'03*. Moscow 2003. Vol. 1, pp. 49-52.
- [10] S. I. Amari. *Mathematical theory of neural learning*. *New Generation Computing*. 1991. Vol. 8, pp. 281-294.
- [11] S. Haykin. *Neural Networks, a Comprehensive Foundation*. Macmillan College Publishing Company. New York 1994.
- [12] J. Hertz. A. Krogh. R. G. Palmer. *Introduction to Theory of Neural Computing*. WNT, Warsaw 1995.
- [13] K. Saeed, M. Tabedzki, "Hybrid System for Letter Recognition," Third International Conference on Neural Networks and Artificial Intelligence - ICNNAI'03, pp. 44 - 48, 12-14 November, Mińsk 2003.
- [14] N. Vapnik V. *Principle of risk minimization for learning theory*. NIPS4. Eds. J. Moody, S. Hanson, R. Lippmann. Morgan Kaufmann, San Mateo 1992, pp. 831-838.
- [15] R. Hecht-Nielsen. *Neurocomputing*. Addison Wesley. Amsterdam, 1991.
- [16] K. Saeed et al, "Language Processing: Word Recognition without Segmentation," JASIST - Journal of the American Society for Information Science and Technology, Volume 52, Issue 14, 2001, pp. 1275-1279, John Wiley and Sons, New York 2001.
- [17] K. Saeed, M. Kozłowski, "An Image-Based System for Spoken-Letter Recognition," 10<sup>th</sup> Int. Conference CAIP'03 on Computer Analysis of Images and Patterns, August 2003, Groningen. Proceedings published in: *Lecture Notes in Computer Science*, Petkov and Westenberg (Eds.), pp. 494-502, LNCS 2756, Springer-Verlag Heidelberg: Berlin 2003.

- [18] K. Saeed, M. Nammous, "Spoken Arabic-Digit Recognition," Submitted to ACS-CISIM'04, *WSFiZ Press*.
- 



**Khalid Saeed** received his B.Sc. degree in Electrical Engineering from University of Baghdad in 1976, his M.Sc. degree in Electronics in 1978 and Ph.D. (distinction) in Telecommunications in 1981. Both of his M.Sc. and Ph.D. he received from Wroclaw University of Technology. He is presently with Bialystok Technical University at Department of Computer Engineering (head 1994-2000), Faculty of Computer Science (Vice-Dean 1996-1999). His research interests include Information Processing Systems and their Hybrid Algorithms and applications in Image Analysis and Description

**Marek Tabedzki** received his M.Sc. degree (distinction) in Computer Science from Bialystok Technical University (Poland) in 2001. He is presently with Bialystok Technical University at Faculty of Computer Science. His research interests include Information Processing Systems, particularly Digital Image Processing and Pattern Recognition.

