Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 22 (2004) 971–981

www.elsevier.com/locate/imavis

Improved method of handwritten digit recognition tested on MNIST database

Ernst Kussul*, Tatiana Baidyk

UNAM, CCADET (Centro de Instrumentos), Cd. Universitaria, A.P. 70-186, C.P. 04510 Mexico D.F., Mexico

Received 31 July 2003; received in revised form 26 November 2003; accepted 22 March 2004

Abstract

We have developed a novel neural classifier LImited Receptive Area (LIRA) for the image recognition. The classifier LIRA contains three neuron layers: sensor, associative and output layers. The sensor layer is connected with the associative layer with no modifiable random connections and the associative layer is connected with the output layer with trainable connections. The training process converges sufficiently fast. This classifier does not use floating point and multiplication operations. The classifier was tested on two image databases. The first database is the MNIST database. It contains 60,000 handwritten digit images for the classifier training and 10,000 handwritten digit images for the classifier testing. The second database contains 441 images of the assembly microdevice. The problem under investigation is to recognize the position of the pin relatively to the hole. A random procedure was used for partition of the database to training and testing subsets. There are many results for the MNIST database in the literature. In the best cases, the error rates are 0.7, 0.63 and 0.42%. The classifier LIRA gives error rate of 0.61% as a mean value of three trials. In task of the pin–hole position estimation the classifier LIRA also shows sufficiently good results.

Keywords: Handwritten digit recognition; LImited Receptive Area neural classifier; MNIST database; Microdevice assembly

1. Introduction

The neural networks are widely used for image recognition [7,13,15]. We distinguish two categories of neural network classifiers. The first one uses the gradient-based training process. As a rule such classifiers have various layers of neurons and all the neurons have differentiable characteristics [13,15]. During the training process the synaptic weights of the connections among all neurons are modified. The second category of neural networks also has some neuron layers but only two ultimate layers are connected with the modifiable connections [5,7,11]. The initial layers contain the binary neurons connected with no modifiable connections. The synaptic weights of these connections are determined using a random procedure. The first category of neural networks is good for the recognition of general properties of the objects. For example, if we have to distinguish the letter 'O' from the letter 'L' such networks give very good results. The second category of neural networks is better for recognition of local properties of the objects. For example, it is good if it is necessary to recognize the letter 'O' from the letter 'Q'. For such tasks we propose the neural classifier LImited Receptive Area (LIRA), which belongs to the second category of neural networks.

The classifier LIRA is based on Rosenblatt's perceptron principles. We proposed two variants of LIRA classifier: LIRA_binary and LIRA_grayscale. The first one is used for recognition of binary (black-and-white) images and the second one is used for recognition of grayscale images. We made some changes in perceptron structure, training and recognition algorithms.

Rosenblatt's perceptron contains three layers of neurons. The first S-layer corresponds to the retina. In technical terms it corresponds to the input image. The second A-layer (associative layer) corresponds to the feature extraction subsystem. The third R-layer corresponds to the system output. Each neuron of this layer corresponds to one of the output classes. In the handwritten digit recognition task this layer contains 10 neurons corresponding to digits 0,…,9.

* Corresponding author. Tel.: +52-55-5622-8602; fax: +52-55-5550-0654.

E-mail addresses: ekussul@servidor.unam.mx (E. Kussul), tbaidyk@aleph.cinstrum.unam.mx (T. Baidyk).

0262-8856/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2004.03.008

Table 1
Recognition rate of different classifiers

Methods	% of error rate	References
Reduced set SVM poly 5	1.0	[12]
LeNet-5 (neural net)	0.95	[12]
Virtual SVM poly 9 [distortions]	0.8	[12]
LeNet-5 [distortions] (neural net)	0.8	[12]
Boosted LeNet-4 [distortions] (neural net)	0.7	[12]
Shape matching + 3-NN	0.63	[2,3]
Proposed classifier LIRA_grayscale (neural net)	*0.61*	*New*
SVC-rbf_grayscale	0.42	[4]

The connections between the layers S and A are established using a random procedure and cannot be changed during the perceptron training. They have the weights -1 or 1.

Each neuron of the A-layer has connections with all neurons of the R-layer. Initially, the connection weights are set to 0. The weights are modified during the perceptron training. The rule of weights modification corresponds to the training algorithm. We used the training algorithm slightly different from Rosenblatt's one. We also modified a random procedure of the S-connections arrangement. Our latest modifications are related to the rule of the winner selection in the output R-layer, and adaptation of the classifier for grayscale image recognition.

We tested the classifier LIRA in two applications: digital recognition and microdevice assembly. There are many applications, for example, bank checks, custom declaration automatic reading, etc. which need to recognize handwritten digits.

Various methods were proposed to solve this problem [2,3,6,8,9,12]. For estimation of the method effectiveness the most important parameter is error rate. This parameter shows which proportion of samples in test database is recognized incorrectly.

The MNIST database contains 60,000 handwritten digits in the training set and 10,000 handwritten digits in the test set. Different classifiers proved on this database [2,4,9,12] had shown error rate from 1.00 to 0.42% (Table 1).

For testing LIRA in assembly application we used the second database which contains 441 images of an assembly microdevice. The problem under investigation is to recognize the position of the pin relatively to the hole. A random procedure was used for partition of the database to training and testing subsets.

2. Rosenblatt perceptron and RSC classifier

The classifier LIRA was developed on the base of the Rosenblatt perceptron [14] and the Random Subspace Classifier (RSC) classifier [11]. The 3-layer Rosenblatt perceptron contains the sensor S-layer, the associative A-layer and the reaction R-layer. Many investigations were dedicated to perceptrons with one neuron in the R-layer [14]. Such perceptron can recognize only two classes. If output of the R neuron is higher than the predetermined threshold T, the input image belongs to class 1. If it is lower than T, the input image belongs to class 2. The sensor S-layer contains two-state elements $\{-1,1\}$. The element is set to 1 if it belongs to the object and set to -1, if it belongs to the background.

The associative A-layer contains neurons with 2-state outputs $\{0,1\}$. The inputs of these neurons are connected with the outputs of the S-layer neurons with no modifiable connections. Each connection may have the weight 1 (positive connection); or the weight -1 (negative connection). If the threshold of such neuron equals to number of its input connections, this neuron is active only in the case if all positive connections correspond to the object and negative connections correspond to the background.

The neuron R is connected with all neurons of the A-layer. The weights of these connections are changed during the perceptron training. The most popular training rule is increasing the weights between active neurons of the A-layer and the neuron R if the object belongs to class 1. If the object belongs to class 2 the corresponding weights are decreasing. It is known that such perceptron has fast convergence and can form nonlinear discriminating surfaces. The complexity of discriminating surface depends on the number of A-layer neurons.

The RSC was developed for general classification problem in parameter spaces of limited dimensions. The structure of this classifier is presented in Fig. 1.

This classifier contains four neural layers:

the input layer $X = x_1, x_2, \ldots, x_K$
the intermediate layer $GROUP = group_1, group_2, \ldots, group_N$
the associative layer $A = a_1, a_2, \ldots, a_N$
the output layer $Y = y_1, y_2, \ldots, y_m$.

Each neuron x_i of the input layer corresponds to the component of the input vector to be classified. Each $group_i$ of the intermediate layer contains some quantity P of neuron pairs p_{ij}. Each pair p_{ij} contains one ON-neuron and one OFF-neuron (Fig. 1). The ON-neuron is active if $x_r > T_{ONij}$. Each OFF-neuron is active if $x_r < T_{OFFij}$, where T_{OFFij} is the threshold of the OFF-neuron, T_{ONij} is the threshold of the ON-neuron. Each pair p_{ij} is connected with a randomly selected neuron of the input layer X. All neuron thresholds of the layer $GROUP$ are selected randomly under condition $T_{ONij} < T_{OFFij}$ in each pair. All the neurons of $group_i$ are connected with one neuron a_i of the associative layer A. A neuron a_i is active if and only if all the neurons of $group_i$ are active. The output of

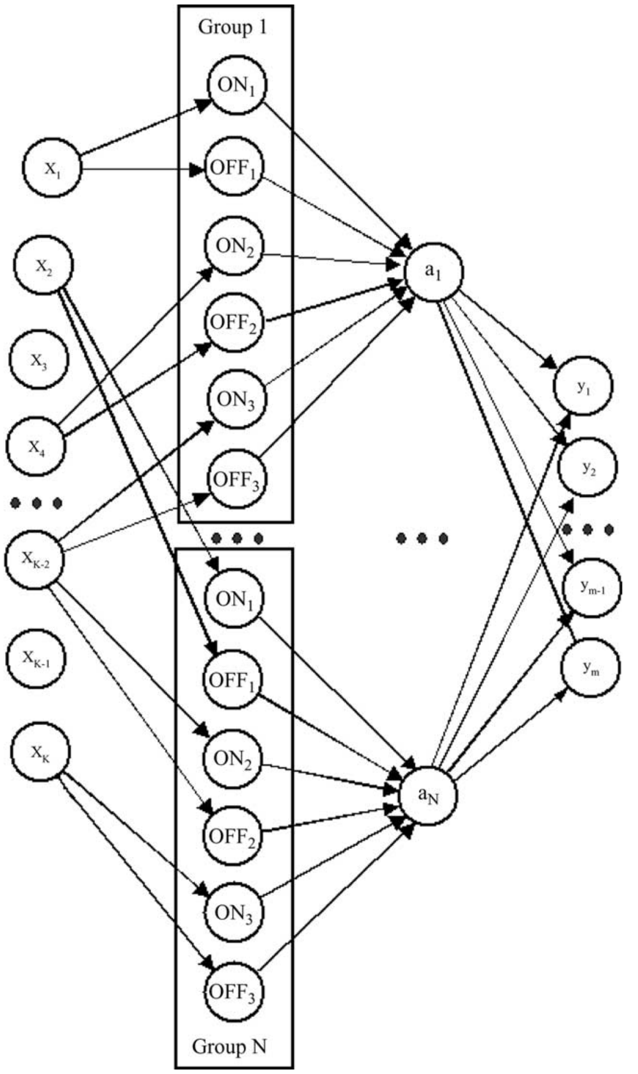

Fig. 1. The structure of the Random Subspace Classifier.

active neuron equals to 1. If the neuron is not active, its output equals to 0. Each neuron of the *A*-layer is connected with all neurons of the output layer *Y*. The training process changes the weights of these connections. The training and the winner selection rules are the same as in the classifier LIRA.

The main difference of the classifier LIRA from the RSC classifier is the absence of the group layer. Instead of each pair of the neurons in the layer *GROUP*, the classifier LIRA uses one connection ON-type or OFF-type which could be active or inactive. This modification permits to increase the classification speed. Other difference is related with the way of applications of this classifiers. We applied the RSC classifier for the texture recognition and other problems where the activities of the input neurons were calculated with a special algorithm of the feature extraction. The classifier LIRA is applied directly to a raw image.

3. Description of the Rosenblatt perceptron modifications for the LIRA-binary

We proposed several changes to the perceptron structure to create the neural classifiers for handwritten digit recognition. To examine them we used the MNIST database [12], which was created from the NIST database, composed from black and white images. The original black and white digit images were size normalized to fit in a 20 × 20 pixels box. The resulting images contain gray levels as a result of the antialiasing (image interpolation) technique used by the normalization algorithm. The images were centered in a 28 × 28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28 × 28 field [12].

The binary image is obtained from a gray-level image by the following procedure. The threshold *th* is computed as

$$th = \frac{2\left(\sum_{i=1}^{W_S}\sum_{j=1}^{H_S} b_{ij}\right)}{W_S H_S}, \tag{1}$$

where H_S is the number of rows of the image; W_S is the number of columns of the image; b_{ij} is the pixel brightness of a grayscale image; s_{ij} is the pixel brightness of the resulting binary image:

$$s_{ij} = \begin{cases} 1, & \text{if } b_{ij} > th, \\ -1, & \text{if } b_{ij} \le th. \end{cases} \tag{2}$$

For the MNIST database $H_S = W_S = 28$.

For the first modification of the Rosenblatt perceptron 10 neurons were included into the *R*-layer. In this case it is necessary to introduce the rule of winner selection. In the first series of experiments we used the simplest rule of winner selection. The neuron from the *R*-layer having the highest excitation determines the class under recognition. Using this rule we obtained error rate of 0.79%.

After that we modified the winner selection rule and achieved the error rate of 0.63%. We will describe this selection rule later.

The second modification was made in the training process. Let the neuron-winner have excitation E_w, its nearest competitor has excitation E_c. If

$$(E_w - E_c)/E_w < T_E \tag{3}$$

the competitor is considered as a winner. Here, T_E is the superfluous excitation of the neuron-winner.

The third modification is concerned with connections. As distinct from the Rosenblatt perceptron our neural classifier has only positive connections between the *A*-layer and the *R*-layer. In this case the training procedure is the following:

1. Let *j* correspond to the correct class under recognition. During the recognition process we obtain excitations of *R*-layer neurons. The excitation of neuron R_j

corresponding to the correct class is decreased by the factor $(1 - T_E)$. After this the neuron having maximum excitation R_k is selected as winner.

2. If $j = k$, nothing to be done.
3. If j does not equal k

$$w_{ij}(t+1) = w_{ij}(t) + a_i, \tag{4}$$

where $w_{ij}(t)$ is the weight of the connection between the i-neuron of the A-layer and the j-neuron of the R-layer before modification, $w_{ij}(t+1)$ is the weight after modification, a_i is the output signal (0 or 1) of the i-neuron of the A-layer

$$\begin{aligned} w_{ik}(t+1) &= w_{ik}(t) - a_i, \quad \text{if } (w_{ik}(t) > 0), \\ w_{ik}(t+1) &= 0, \qquad\qquad\quad \text{if } (w_{ik}(t) = 0), \end{aligned} \tag{5}$$

where $w_{ik}(t)$ is the weight of the connection between the i-neuron of the A-layer and the k-neuron of the R-layer before modification, $w_{ik}(t+1)$ is the weight after modification. More detailed description of the training procedure will be done further.

The perceptron with these changes is termed the LImited Receptive Area classifier for binary images (LIRA_binary) (Fig. 2). More general case of such classifier was developed and termed RSC [11] and was based on the Random Threshold Classifier [7,10].

Each A-layer neuron of the LIRA classifier has random connections with the S-layer. To install these connections it is necessary to enumerate all elements of the S-layer. Let the number of these elements equal to N_S. To determine the connections of the A-layer neuron we select the random number from the range $[1, N_S]$. This number determines the S-layer neuron, which will be connected with the mentioned A-layer neuron. This rule is used to determine the connections between all A-layer neurons and the S-layer neurons. Frank Rosenblatt proposed this rule [14]. Our experience shows that it is possible to improve the perceptron performance due to the modification of this rule.

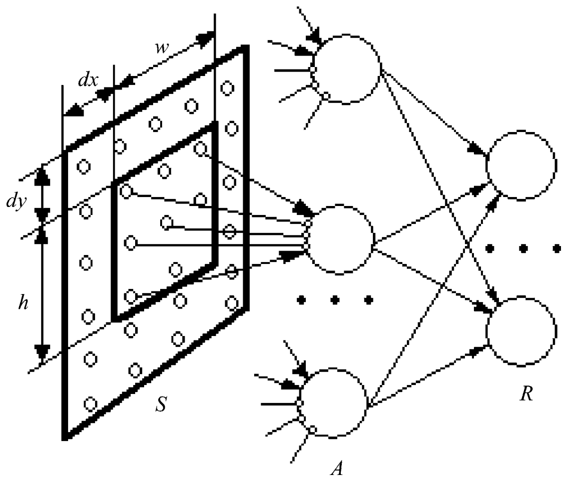

Fig. 2. LImited Receptive Area (LIRA) classifier.

The fourth modification is the following. We connect the A-layer neurons with the S-layer neurons randomly selected not from the whole S-layer but from the rectangle (hw), which is located in the S-layer (Fig. 2).

The distances dx and dy are random numbers selected from the ranges: dx from $[0, W_S - w]$ and dy from $[0, H_S - h]$, where W_S, H_S stand for width and height of the S-layer.

3.1. Mask design

The associative neuron mask is a set of the positive and negative connections of the A-layer neuron with the retina. To design the mask the procedure of random selection of connections is used. This procedure begins from choice of the upper left corner of the rectangle in which all positive and negative connections of the associative neuron are located. The following formulas are used

$$\begin{aligned} dx_i &= random_i(W_S - w), \\ dy_i &= random_i(H_S - h), \end{aligned} \tag{6}$$

where i is the neuron position in the associative layer A; $random_i(z)$ is the random number from the range $[0, z]$. After that each positive and negative connection position within the rectangle is defined by couple of numbers

$$\begin{aligned} x_{ij} &= random_{ij}(w), \\ y_{ij} &= random_{ij}(h), \end{aligned} \tag{7}$$

where j is the number of the ith neuron connection with retina.

The absolute coordinates of the connection on the retina are defined by a couple of numbers:

$$\begin{aligned} X_{ij} &= x_{ij} + dx_i, \\ Y_{ij} &= y_{ij} + dy_i. \end{aligned} \tag{8}$$

3.2. Image coding

Any input image defines the activities of the A-layer neurons in one-to-one correspondence. The binary vector which corresponds to the associative neuron activities is termed the image binary code $\mathbf{A} = a_1, \ldots, a_n$ (where n is the number of the A-layer neurons). The procedure, which transforms the input image to the binary vector A, is termed the image coding.

In our system the ith neuron of the A-layer is active only if all the positive connections with retina correspond to the object and all negative connections correspond to the background. In this case $a_i = 1$, in opposite case $a_i = 0$. From the experience of the work with such systems it is known that the active neuron number m in the A-layer must be many times less than whole neuron number n of this layer. In our work we usually use the following expression $m = c\sqrt{n}$, where c is the constant, which belongs to the range from

1 to 5. This ratio corresponds to neurophysiological facts. The active neurons number in the cerebral cortex is hundreds times less than the total number of neurons.

Taking into account the little number of active neurons it is convenient to represent the binary vector **A** not explicitly but as a list of active neuron numbers. Let, for example, the vector **A** be:

$$\mathbf{A} = 00010000100000010000.$$

The corresponding list of the active neuron numbers will be 4, 9, and 16. This list is used to save the image codes in compact form and for fast calculation of the neuron activities of the output layer. Thus, after the coding procedure execution every image has corresponding list of active neuron numbers.

3.3. Training procedure

Before the training all weights of the connections between the neurons of the A-layer and the R-layer are set to zero.

1. The training procedure begins from the input of the first image to the perceptron. The image is coded and the R-layer neuron excitation E_i is computed. The excitation E_i is defined as

$$E_i = \sum_{j=1}^{n} a_j w_{ji} \tag{9}$$

   where E_i is the excitation of the ith neuron of the R-layer; a_j is the excitation of the jth neuron of the A-layer; w_{ji} is the connection weights between the jth neuron of the A-layer and the ith neuron of the R-layer.
2. We want the recognition to be robust. After the calculation of all neuron excitations of the R-layer the correct name of input image is read from the mark file of the MNIST database. The excitation E of the corresponding neuron is recalculated according to the formula:

$$E_k^* = E_k(1 - T_E). \tag{10}$$

   After that we find the neuron (winner) with the maximal activity. This neuron presents the recognized handwritten digit.
3. We denote the neuron-winner number as i_w, and the number of neuron, which really corresponds to the input image, as i_c. If $i_w = i_c$ nothing to be done. If $i_w \neq i_c$

$$\begin{aligned}&(\forall j)(w_{ji_c}(t+1) = w_{ji_c}(t) + a_j)\\ &(\forall j)(w_{ji_w}(t+1) = w_{ji_w}(t) - a_j)\\ &\text{if } (w_{ji_w}(t+1) < 0)\ \ w_{ji_w}(t+1) = 0,\end{aligned} \tag{11}$$

   where $w_{ji}(t)$ is the weight of the connection between the j-neuron of the A-layer and the i-neuron of the R-layer before modification, $w_{ji}(t+1)$ is the weight after modification.

The training process is carried out iteratively. After the input of all images from the training subset the total number of training errors is calculated. If this number is higher than 1% of the total number of images then the next training cycle is doing. If the error number is less than 1% the training process is stopped. The training process is also stopped when the cycle number is more than preestablished value. In previous experiments this value was 10 cycles, and in final ones—40 cycles.

It is obvious that with every new training cycle the image coding procedure is repeated and gives the same results as in previous cycles. Therefore in final experiments we performed the image coding process only once and recorded the lists of active neuron numbers for each image on hard drive. Later for all cycles we used not the images but the corresponding lists of the active neurons. Due to this procedure the training process was accelerated approximately by an order of magnitude.

It is known [12] that the handwritten symbol recognition rate may be increased essentially if during the training cycle the images are input not only in initial state but also with shifting and with changing the image inclination (so called distortions). In final experiments in addition to the initial image we used 16 variants of each image, i.e. 16 distortions.

The distortions can be used to increase the effective size of a data set without collecting more data. We used 16 distortions (Fig. 3): 12 shifts and 4 skewing. The skewing angles selected were −26, −13, 13 and 26°.

3.4. Recognition procedure

To examine the recognition rate the test set of the MNIST database was used. This test set contains 10,000 images. The coding and calculation of neuron activity were made with the same rules as during the training process, but the value T_E (reserve of robustness) was 0.

The recognition process for the new classifier differs from the previous ones. In this version we use distortions in recognition process too. There is the difference between implementation of distortions during the training session and the recognition session. In the training session each new position of initial image produced by distortions is considered as a new image, which is independent of other image distortions. In the recognition session it is necessary to introduce a rule of the decision-making. All results of one image and its distortions recognition must be used for obtaining one result, which gives the class name of the image under recognition. We have developed two rules of the decision-making.

Rule 1. According to this rule all excitations of the R-layer neurons are sum for all the distortions

$$E_i = \sum_{k=1}^{d} \sum_{j=1}^{n} a_{kj} w_{ji}, \tag{12}$$

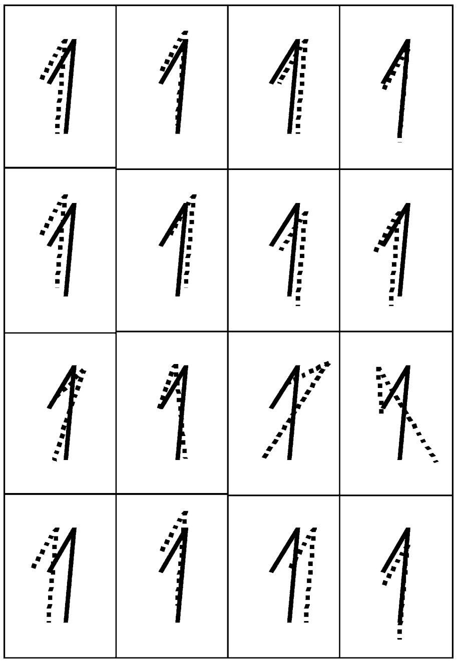

Fig. 3. The scheme of 16 distortions.

where E_i is the excitation of the ith neuron of the R-layer; a_{kj} is the excitation of the jth neuron of the A-layer in the kth distortion; w_{ji} is the connection weight between the jth neuron of the A-layer and the ith neuron of the R-layer. And after that the neuron-winner is selected as a recognition result.

Rule 2. The second rule consists in calculations of the R-layer neurons excitations and selection of the neuron-winner and its nearest competitor for each distortion. For the kth distortion the ratio r_k of the neuron-winner excitation E_{wk} to its nearest competitor excitation E_{ck} is calculated

$$r_k = \frac{E_{wk}}{E_{ck}}. \tag{13}$$

After that we select the distortion with the maximal r_k. The neuron-winner of this distortion is considered to be the recognition result.

4. Description of the Rosenblatt perceptron modifications for the LIRA-grayscale

To adapt the LIRA classifier for grayscale image recognition we have added the additional neuron layer between the S-layer and the A-layer. We termed it the I-layer (intermediate layer, see Fig. 4).

Each input of the I-layer neuron has one connection with the S-layer. Each output of this neuron is connected with the input of one neuron of the A-layer. All the I-layer neurons connected to one A-layer neuron form the group of this A-layer neuron. The number of neurons in one group corresponds to the number of positive and negative connections between one neuron of the A-layer and the retina in the LIRA_binary structure. The I-layer neurons could be ON-neurons or OFF-neurons. The output of the ON-neuron i is '1' when its input is higher than the threshold θ_i and in opposite case it equals to '0'. The OFF-neuron j output is '1' when its input is less than the threshold θ_j and in opposite case it equals to '0'.

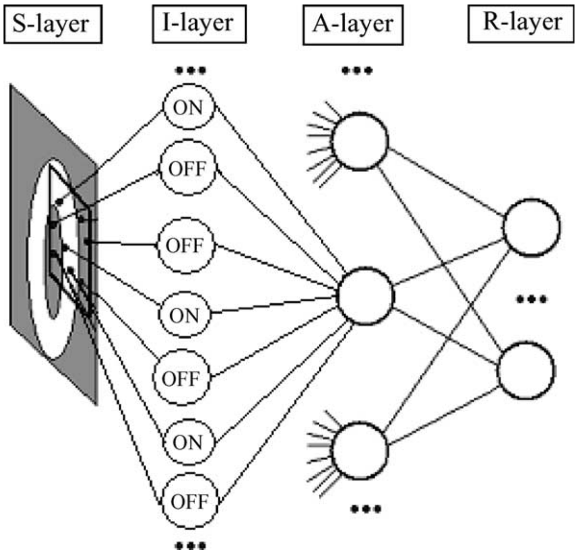

Fig. 4. LIRA_grayscale scheme.

The ON-neuron number in each group corresponds to the number of the positive connections of one A-layer neuron in the LIRA_binary structure. The OFF-neuron number in each group corresponds to the number of the negative connections. In our case we selected three ON-neurons and five OFF-neurons. The rule of connection arrangement between the retina and one group of the I-layer is the same as the rule of mask design for one A-layer neuron in the LIRA_binary.

The thresholds θ_i and θ_j are selected randomly from the range $[0, \eta b_{\max}]$, where $b_{\max}$ is maximal brightness of the image pixels; η is the parameter from [0,1], which is selected experimentally.

The output of the A-layer neuron is '1' if all outputs of its I-layer group are '1'. If any neuron of this group has the output '0' the A-layer neuron has the output '0'.

5. Handwritten digit recognition results for LIRA-binary

We carried out preliminary experiments to estimate the performance of our classifiers. On the basis of these preliminary experiments we selected the best classifiers and carried out final experiments to obtain the maximal recognition rate. In the preliminary experiments we changed the A-layer neuron number from 1000 to 128,000 (Table 2). These experiments showed that recognition error number has been decreased approximately by the factor 8 with

Table 2
The recognition rates of classifier in the preliminary experiments

A-layer neuron number	Error number				
	$p = 0.2$	$p = 0.4$	$p = 0.6$	$p = 0.8$	$p = 1$
1000	3461	1333	1297	1355	1864
2000	1705	772	772	827	1027
4000	828	452	491	532	622
8000	482	338	335	388	451
16,000	330	249	247	288	337
32,000	245	205	207	246	270
64,000	218	186	171	190	217
128,000	207	170	168	190	195

increasing of the A-layer neuron number. The main disadvantages of the large A-layer are the increasing of train and recognition time and memory capacity.

We also changed the ratio $p = w/W_S = h/H_S$ from 0.2 to 0.8. The parameter T_E was 0.1. In these experiments we did not use the distortions in either training or recognition sessions.

For each set of parameters we made 10 training cycles on the MNIST training set. After that we estimated the recognition rate on the MNIST test set. The recognition rates obtained in the preliminary experiments are presented in Table 2.

In the preliminary experiments we generated 3 positive and 3 negative connections for each A-layer neuron. In the final experiments we generated 3 positive and 5 negative connections. The number of the A-layer neurons was 256,000. The windows parameters were $w = 10$ and $h = 10$ and the retina size was 28×28. The training cycle number was 40.

The coding time was 20 h and the training time was 45 h. The recognition time (for 10,000 samples) was 30 min without distortions, 60 min for 4 distortions and 120 min for 8 distortions. We made different experiments with different number of distortions during the recognition session (4 and 8). We created distortions only with shifting (the first four or eight cases in Fig. 3). For comparison we made experiments without distortions in recognition session.

For statistical comparison purposes we made three experiments for each set of parameters. The difference between the experiments consists in using different random structure of connections between the S-layer and the A-layer. In Table 3 each row corresponds to one of such experiments. The column title ‘Rule 1’ corresponds to the first rule of the winner selection (Formula 12) and ‘Rule 2’ corresponds to the second rule of the winner selection (Formula 13).

Table 3
Error numbers in the final experiments

Number of experiments	Without distortions	Rule 1		Rule 2	
		4 dist.	8 dist.	4 dist.	8 dist.
1	72	68	63	75	75
2	86	65	65	72	67
3	83	66	62	68	72
Mean value	80	66	*63*	72	71

The error number 63 corresponds to 99.37% of recognition rate. Serge Belongie et al. [2] has the same result on the MNIST database.

Table 4
Error numbers in the final experiments

Number of experiments	1	2	3
Number of errors	60	59	64
Mean number of errors		61.33	

6. Handwritten digit recognition results for the LIRA_grayscale

The recognition rates obtained in the experiments with the LIRA_grayscale are presented in Table 4. In this case we also made three experiments for statistical comparison purposes. The difference between the experiments consists in the use of the different random structures of connections between the S-layer and the A-layer. In Table 4 each column is marked by the number of concrete experiment.

We obtained the minimal number of errors, 59. To our knowledge at present this result is one of the best from known results.

7. Application of the LIRA classifier for flat image recognition in the process of microdevice assembly

One of the assembly tasks is to install the pin into the hole (Fig. 5). For this purpose it is necessary to know the displacements (dx, dy, dz) of the pin tip relative to the hole. It is possible to evaluate these displacements with stereovision system, which resolves 3D problems. Stereovision system demands two TV cameras. To simplify the control system we propose to transform 3D into 2D images, preserving all the information about mutual location of the pin and the hole. This approach makes it possible to use only one TV camera [1].

To perform 3D → 2D transformation we use the shadows of the pin. Four light sources are used to obtain pin shadows.

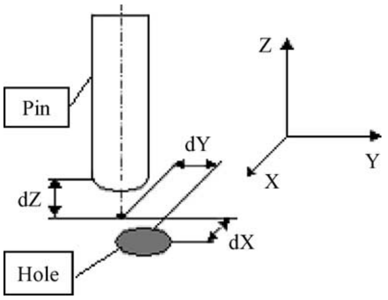

Fig. 5. Mutual location of the pin and hole.

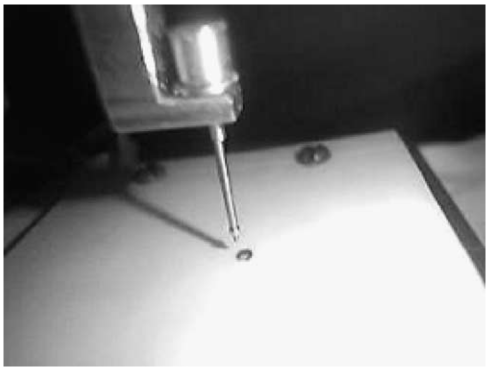

Fig. 6. Example of real image.

Mutual location of these shadows and the hole contains all the information about displacements of the pin relative to the hole. The displacements in the horizontal plane (dx, dy) could be obtained directly by displacements of shadows center points relative to the hole center. Vertical displacement of the pin may be obtained from the distance between the shadows. To calculate the displacements it is necessary to have all the shadows in one image. We capture four images corresponding to each light source sequentially. After this it is necessary to extract contours and superpose contour images.

We can consider the resulting image as a 'symbol' which can be interpreted and treated as 'letter' or 'digit'.

The example of real image with one of the light sources is shown in Fig. 6. They were made with the system presented in Fig. 7.

The contour extraction procedure was applied to these images and after that the corresponding contour images were superimposed. The results are shown in Fig. 8.

The different relative pin–hole positions are shown in Fig. 8. In Fig. 8a the pin is elevated relative to the hole. And the distance between shadows is large. In Fig. 8b the pin is put down to the hole and the shadow contours are connected to the contours of the pin and the hole. In Fig. 8c the pin is shifted to the left from the hole. In Fig. 8d the pin is centered and slightly elevated.

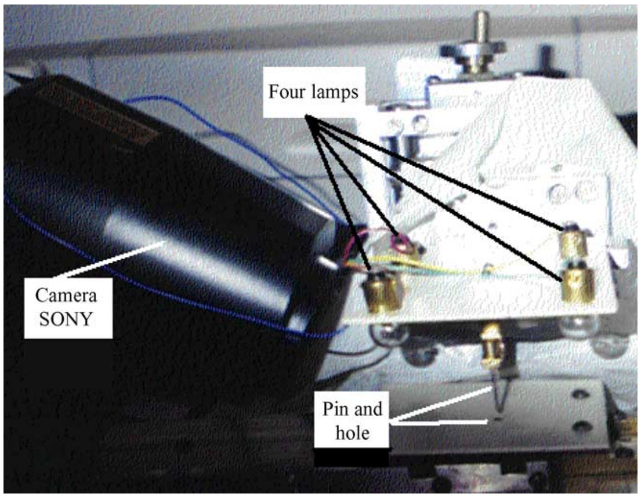

Fig. 7. System with camera 'Sony'.

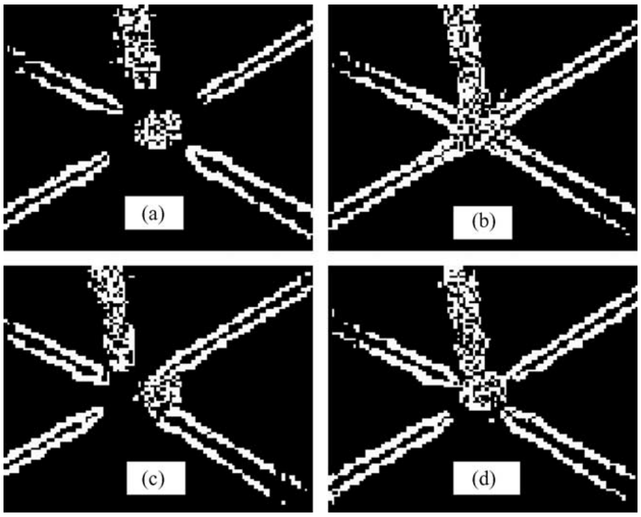

Fig. 8. Superimposed contour image (binary image).

Two data sets were made with different positions of the pin. The first dataset contains 23 images corresponding to the displacements of the pin along the X and Y-axis with the step of 0.5 mm.

The second dataset contains 441 images corresponding to X, Y-displacements of the pin. The step of the displacements was 0.1 mm. The third coordinate Z of the pin was constant in the both datasets.

For the first dataset two classifiers were created to recognize the pin positions. The first classifier has three outputs corresponding to the X-displacements: $X_{pin} < X_{hole}$; $X_{pin} = X_{hole}$; $X_{pin} > X_{hole}$. The second classifier has three outputs corresponding to the Y-displacements: $Y_{pin} < Y_{hole}$; $Y_{pin} = Y_{hole}$; $Y_{pin} > Y_{hole}$. Twelve randomly selected images from the first dataset were used for the classifiers training and the residuary 11 images—for the classifier testing. After the training both the classifier recognized all images correctly (error number = 0).

At present we use the second dataset for more detailed recognition of the pin displacements. The classifier outputs contain more classes than previous classifiers: 21 classes for the X-coordinates and 21 classes for the Y-coordinates (i.e. one R-layer has 21 neurons for the X and the other R-layer has 21 neurons for the Y). Every neuron from the R-layer has 64,000 connections with the previous layer (i.e. the A-layer has 64,000 neurons).

It is known [12] that the recognition rate may be increased if during the training cycle the images are represented not only in the initial state but also with shifted image positions (so called distortions). In case of the microassembly we did four experiment sets. The first set had no distortions. The second set had the image shifts by 1 pixel to the right, to the left, up and down. The third set of experiments had additional diagonal shifts to 1 pixel.

Table 5
The results of the classifier investigation

Distortions	X (%)	Y (%)	$X \pm 0.1$ mm (%)	$Y \pm 0.1$ mm (%)
1	112 (50.7)	44 (19.9)	213 (96.2)	171 (77.5)
5	145 (65.9)	40 (18.4)	219 (99.2)	195 (88.2)
9	149 (67.7)	42 (19.5)	220 (99.5)	199 (89.9)
13	138 (62.8)	57 (26.4)	219 (99.3)	192 (86.9)

The fourth set has additional shifts to the right, to the left, up and down by 2 pixels.

From 441 images the training and the test sets were formed with a random procedure. In Table 5 the results of the classifier investigation are presented.

We investigate the classifier with the different number of distortions: 1, 5, 9, and 13. Number of correct recognitions is presented in the first column for the coordinates X and Y. The recognition rate (%) is in the second column. These recognition rates are not high. But in our case it is necessary to analyze the recognition errors in more detail. The error of position recognition is proportional to the difference between the numbers of the correct class and recognized class. If we take into account the recognition with the precision 0.1 mm we obtain a rather high quality of recognition (the right part of Table 5).

When we determine the relative pin–hole position it is necessary to take into account the image discretization. For small objects comparable with 1 pixel it is impossible to determine their positions with tolerances less than ±0.5 pixel. But the position of larger objects could be obtained with smaller tolerances. It is possible to illustrate this fact on the following example. Let us consider the part of the object presented in Fig. 9.

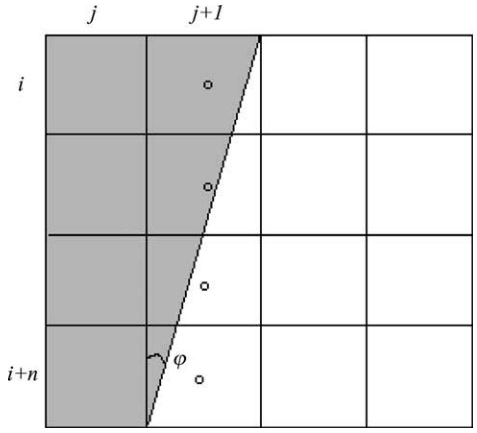

Fig. 9. Example of the object part in the image.

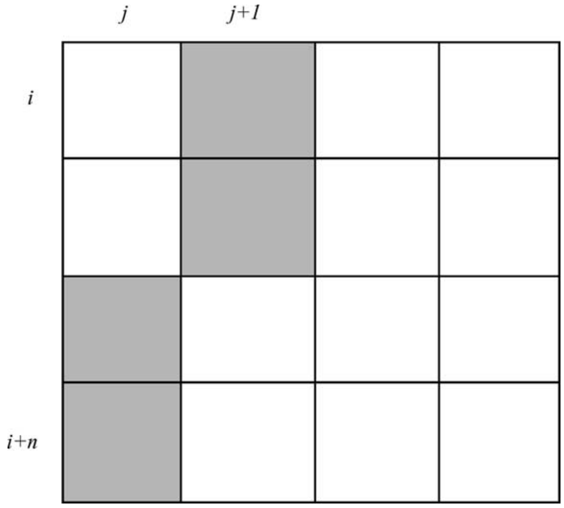

Fig. 10. The contours of the image.

The object has the brightness $br = 1$ (right part of the image), and the background has the brightness $br = 0$ (left part of the image). We suppose that the brightness of the pixel p_{ij} partially occupied by the object has the brightness proportional to the occupied area of the pixel p_{ij} (this assumption is realistic for TV cameras). In our example the object has the edge with the angle f relatively to vertical axis of the raster grid. Let $tg(f)$ be $1/n$. Let the recognition system extract the contour using the following rule

$$c_{ij} = \begin{cases} 1, & \text{if } (br_{ij+1} - br_{ij}) > 0.5 \\ 0 & \text{if } (br_{ij+1} - br_{ij}) \leq 0.5 \end{cases} \tag{14}$$

where $c_{ij} = 1$ corresponds to the contour pixel; br_{ij} ($1 \geq br_{ij} \geq 0$) is the brightness of the pixel p_{ij}. The pixel p_{ij} belongs to the contour if the object edge is located to the left of the central point of the pixel p_{ij} and crosses it. The pixel p_{ij+1} belongs to the contour if the object edge is located to the right of the central point of the pixel p_{ij} and crosses it (Fig. 10).

Let the real position dx^* of the edge corresponds to the central point of this edge. In Fig. 9, $dx^* = j + 0.5$. Let the position of the gravity center of contour pixels be x_c. We determine the recognized edge position as $dx = x_c + 0.5$. In Fig. 10, $dx = j + 0.5$. If the edge is displaced from the initial position showed in Fig. 9 to the position showed in Fig. 11

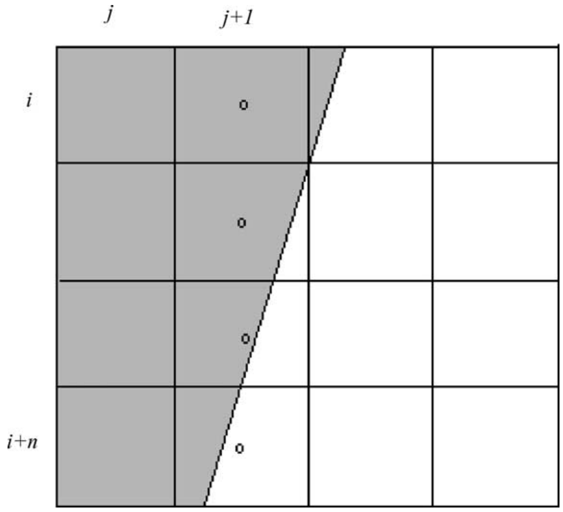

Fig. 11. Displacement of the initial image.

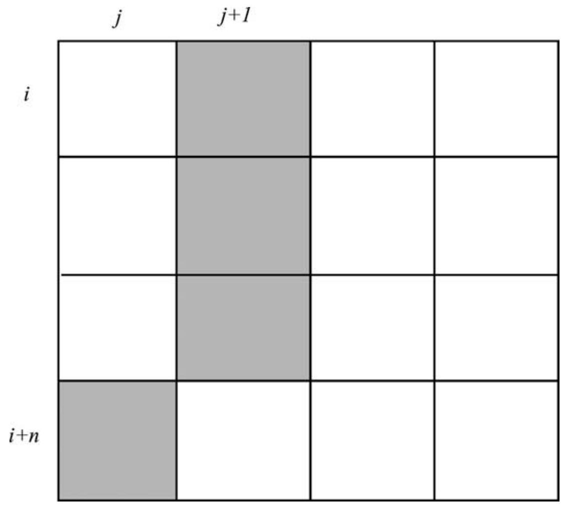

Fig. 12. The contours of the image after displacement.

the real position of the edge will be $j + 0.5 + 1/n$. The recognized position (see Fig. 12) will be $dx = j + 0.5 + 1/n$. Thus in these cases the error will be 0.

Inside this interval the error will change in accordance with the curve shown in Fig. 13.

This means that the large object position could be determined with the tolerances smaller than 0.5 pixel.

In our case, the tolerances of pin–hole displacement was 0.1 mm which corresponds to 1.8 pixels in the X-axis and 1 pixel in the Y-axis and the image objects were much more large than 1 pixel. So we had a good reserve for the pin–hole displacements recognition.

The neural classifier permits us to recognize the pin–hole relative displacement with 2 pixels tolerance. The absolute values of detectable displacements depend on the optical channel resolution. In our case, 2 pixels correspond approximately to 0.1 mm for X-axis and 0.2 mm for Y-axis. This precision is sufficient for many cases of the assembly processes.

8. Discussion

The novel neural classifier LIRA was developed. The classifier LIRA contains three neuron layers: sensor, associative and output layers. The sensor layer is connected with the associative layer with no modifiable random connections and the associative layer is connected with the output layer with trainable connections. The training process converges sufficiently fast. This classifier does not use floating point and multiplication operations. This property in combination with the parallel structure of classifier permits to implement it in low cost, high-speed electronic devices. The classifier LIRA shows good recognition rate. It was tested on two image databases. The first database is the MNIST database. It contains 60,000 handwritten digit images for the classifier training and 10,000 handwritten digit images for the classifier testing. The second database contains 441 images of an assembly microdevice. The problem under investigation is to recognize the pin–hole relative position. A random procedure was used for partition of the database to the training and testing subsets.

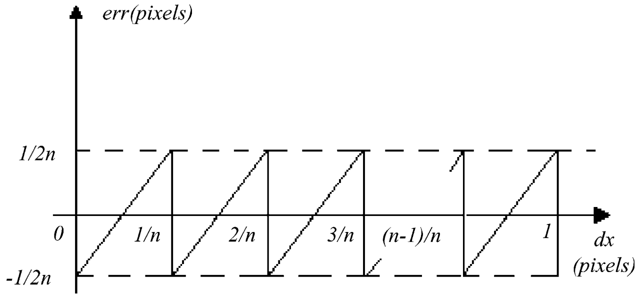

Fig. 13. The error changing.

The results which were obtained on the MNIST database seem to be sufficiently good for applications. But there are many tasks of handwritten number recognition. If the number contains, for example, 10 digits and the recognition rate of one digit is 0.994 (in our case) the whole number recognition rate could be $0.994^{10} = 0.942 = 94.2\%$. This recognition rate is insufficient for many applications. For this reason additional investigations are needed to improve a handwritten digit recognition rate.

The recognition time of each handwritten digit is also an important parameter. In many cases to estimate the recognition time the authors of different methods give the number of multiply accumulate operations for one symbol recognition. For example, for RS-SVM method it equals 650,000, LeNet-5 is about 60% less expensive [12]. It is difficult to compare our classifier using this parameter because our classifier does not use neither multiply operations nor floating point operations. For one digit recognition our classifier needs approximately 50,000 fixed point add operations. It seems to be very fast but it is not the case. For one image coding it needs approximately $10 \times 256{,}000$ readings from memory and logical operations. We must code during recognition not only initial image but also, for example, 4 distortions. All this process demands near 10 million operations for each digit recognition which is difficult to compare with the number of floating point operations. In general our classifier has lower recognition speed than methods by LeCun and SVM.

The other method of recognition time comparison is the classifiers testing on the similar computer. Belongie [2] gives the time of the shape matching as 200 ms on the computer Pentium III, 500 MHz workstation. Using regular nearest neighbor method it is necessary to make N matching for each digit recognition, where N is the size of training set (Belongie used from 15,000 to 20,000 images). In this case, the recognition time of one digit could be from 3000 to 4000 s. We tested our latest version of the classifier also on computer Pentium III, 500 MHz and obtained the recognition time of one digit 0.5 s.

The third important parameter for classifier comparison is the training time. Belongie used the nearest neighbor classifiers which practically need no training time [2]. The training time of LeNet was 2–3 days of CPU to train LeNet-5 on a Silicon Graphics Origin 2000 server using a single 200 MHz on R10000 processor. Our training time is 55 h on the computer Pentium III, 500 MHz.

The experiments with microdevice assembly system showed that the LIRA classifier could estimate the relative pin–hole position with the tolerances of 1–2 pixels. In principle it is possible to make such estimation with subpixel tolerances. This case demands additional investigations which should be made in the future.

9. Conclusions

The novel neural classifier LIRA for an image recognition task was developed. The classifier was tested in a handwritten recognition problem and a pin–hole relative position estimation for an automatic microassembly problem and showed good results. For a handwritten recognition the experiments with the MNIST database showed that this classifier has one of the best recognition rate (99.41%) among other classifiers proved on this database. In the microassembly problem the relative pin–hole position was estimated with 1–2 pixels tolerances. The recognition time for handwritten digits was 0.5 s on the Pentium III, 500 MHz. The training time (55 h) is reasonably good. The main drawback of this classifier is a relatively low recognition speed. To eliminate this drawback it is possible to implement the classifier LIRA in special parallel electronic device. This device could have the low cost because the classifier LIRA does not use floating point and multiplication operations.

The classifier could be used for handwriting recognition and for robotic vision.

Acknowledgements

We are grateful to Yan LeCun for his database MNIST, which can be used by anyone who wants to test his classifier.

This work was funded by the projects CONACYT 33944-U, DGAPA PAPIIT IN-118799, PAPIIT 112102, NSF-CONACYT 39395-A.

References

[1] T. Baidyk, E. Kussul, Application of neural classifier for flat image recognition in the process of microdevice assembly, Proceedings of the International Joint Conference on Neural Networks, Hawaii, USA 1 (2002) 160–164.

[2] S. Belongie, J. Malik, J. Puzicha, Matching shapes, Proceedings of Eighth IEEE International Conference on Computer Vision ICCV 1 (2001) 454–461.

[3] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.

[4] C.-K. Kiu, K. Nakashima, H. Saco, H. Fujisawa, Handwritten digit recognition using state-of-the-art techniques, Proceedings of the IEEE Eighth International Workshop on Frontiers in Handwritten Recognition, Ontario, Canada (2002) 320–325.

[5] T.J.W. Clarke, R.W. Prager, F. Fallside, The modified Kanerva model: theory and results for real-time word recognition, IEE Proceedings-F 138 (1) (1991) 25–31.

[6] M.S. Hoque, M.C. Fairhurst, A moving window classifier for off-line character recognition, Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition (2000) 595–600.

[7] E.M. Kussul, T.N. Baidyk, Neural random threshold classifier in OCR application, Proceedings of the Second All-Ukrainian International Conference UkrOBRAZ'94, Ukraine (1994) 154–157.

[8] E.M. Kussul, T.N. Baidyk, Improved method of handwritten digit recognition, Proceedings of the 15th International Conference on Vision Interface, Canada (2002) 192–197.

[9] E.M. Kussul, T.N. Baidyk, et al., Rosenblatt perceptrons for handwritten digit recognition, Proceedings of International Joint Conference on Neural Networks IJCNN 2 (2001) 1516–1520.

[10] E.M. Kussul, T.N. Baidyk, V.V. Lukovitch, D.A. Rachkovskij, Adaptive high performance classifier based on random threshold neurons, in: Cybernetics and Systems'94, World Scientific, Singapore, 1994, pp. 1687–1695.

[11] E.M. Kussul, D. Rachkovskij, D. Wunsch, The random subspace coarse coding scheme for real-valued vectors, Proceedings of IEEE International Joint Conference on Neural Networks, Washington (1999) 450–455.

[12] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2344.

[13] G. Ososkov, Effective neural network approach to image recognition and control, Proceedings of International Conference on Physics and Control 1 (2003) 242–246.

[14] F. Rosenblatt, Principles of Neurodynamics, Spartan Books, New York, 1962.

[15] S. Vitabile, A. Gentile, F. Sorbello, A neural network based automatic road signs recognizer, Proceedings of the 2002 International Joint Conference on Neural Networks 3 (2002) 2315–2320.