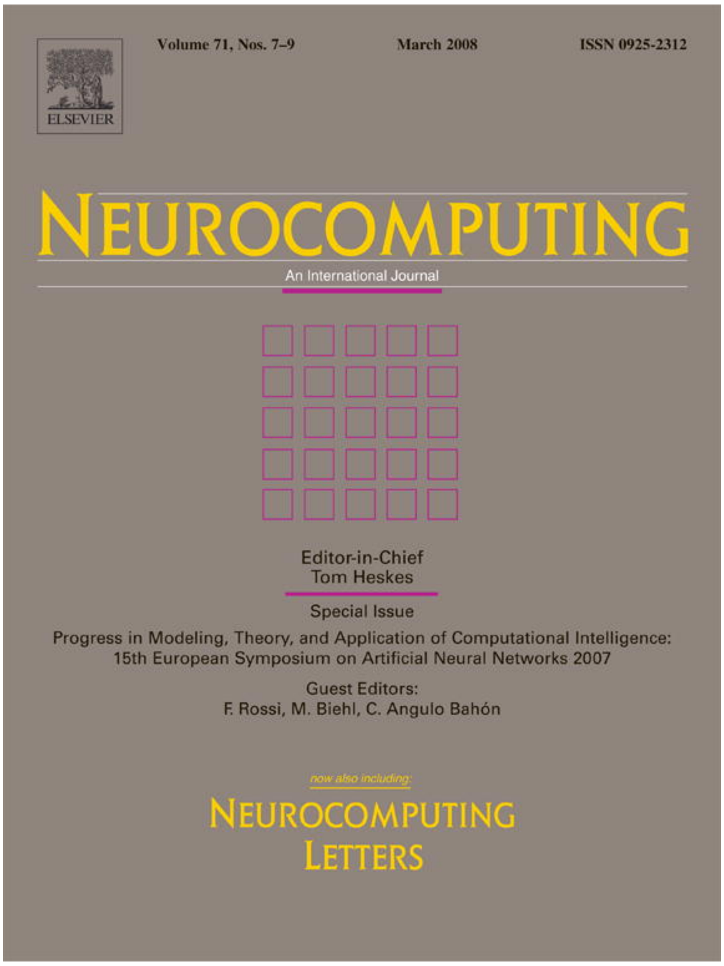

Volume 71, Nos. 7–9
March 2008
ISSN 0925-2312
ELSEVIER
NEUROCOMPUTING
An International Journal
Editor-in-Chief
Tom Heskes
Special Issue
Progress in Modeling, Theory, and Application of Computational Intelligence:
15th European Symposium on Artificial Neural Networks 2007
Guest Editors:
F. Rossi, M. Biehl, C. Angulo Bahón
now also including:
NEUROCOMPUTING
LETTERS

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Neurocomputing 71 (2008) 1413–1421
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Limited receptive area neural classifier for texture recognition of mechanically treated metal surfaces

O. Makeyev[a,*], E. Sazonov[a], T. Baidyk[b], A. Martín[b]

[a]Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13699, USA
[b]Lab of Micromechanics and Mechatronics, CCADET, National Autonomous University of Mexico (UNAM), Cd. Universitaria, Mexico, D.F., 04510, Mexico

Received 24 January 2006; received in revised form 20 April 2007; accepted 22 May 2007
Communicated by D.P. Casasent
Available online 8 June 2007

Abstract

The limited receptive area (LIRA) neural classifier is proposed for texture recognition of mechanically treated metal surfaces. It may be applied in systems that have to recognize position and orientation of complex work pieces during micromechanical device assembly as well as in surface quality inspection systems. The performance of the proposed classifier was tested on a specially created image database with four texture types corresponding to metal surfaces after milling, polishing with sandpaper, turning with lathe and polishing with file. The promising recognition rate of 99.8% was obtained.

Keywords: Metal surface; Texture recognition; Limited receptive area; Neural classifier; Micromechanics

1. Introduction

The main approaches to microdevice production are the technology of microelectromechanical systems (MEMS) [14,20] and microequipment technology (MET) [1,5,8,16]. To get the best of these technologies, it is important to have advanced image recognition systems because feedback based on computer vision can be used to increase the precision and quality of the microcomponent manufacturing [8] and microassembly [1] without considerable increase in the cost of equipment.

Texture recognition systems are widely used for industrial inspection in cases when the texture of a surface defines its quality and therefore affects the durability of the product, for example, in textile industry for inspection of fabric [4], in electronic industry for inspection of the surfaces of magnetic disks [6], etc. Texture recognition is also used when it is necessary to distinguish automatically different types of textures, for example, in decorative and construction industry for classification of polished granite and ceramic titles [18].

Numerous approaches have been developed to solve the texture recognition problem. Many statistical texture descriptors are based on generation of co-occurrence matrices. In Ref. [6], the texture co-occurrence of the nth rank was proposed. The matrix contains statistics of the pixel under investigation and its surrounding. The authors of Ref. [18] proposed the coordinated cluster representation (CCR) as a technique for texture feature extraction. The underlying principle of the CCR is to extract spatial correlation between pixel intensities using the distribution function of the occurrence of texture units. Experiments with one-layer texture classifier in the CCR feature space proved this approach to be effective. Representative texture elements (textons) for texture description and recognition were proposed in Ref. [13]. The vocabulary of textons corresponds to the characteristic features of the image. Several publications describe application of neural networks to texture recognition problem [15,21].

In this paper, we propose a texture recognition system based on the limited receptive area (LIRA) [1] neural

*Corresponding author. Tel.: +52 55 55737253.
E-mail addresses: mckehev@cias.clarkson.edu (O. Makeyev), esazonov@cias.clarkson.edu (E. Sazonov), tbaidyk@aleph.cinstrum.unam.mx (T. Baidyk), anabel@uxmcc2.iimas.unam.mx (A. Martín).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2007.05.004

classifier for recognition of mechanically treated metal surfaces. The proposed texture recognition system may be applied in systems that have to recognize position and orientation of complex work pieces in the task of assembly of micromechanical devices as well as in surface quality inspection systems. Four types of metal surfaces after mechanical treatment were used to test the texture recognition system.

Different lighting conditions and viewing angles affect the grayscale properties of an image due to such effects as shading, shadowing, local occlusions, etc. The real images of metal surfaces obtained in industrial applications have all these problems. Moreover, industrial environments pose some additional problems. For example, a metal surface can have dust on it.

The reason to choose a system based on neural network architecture for the problem at hand was that such systems have already proved their efficacy in texture recognition due to significant properties of adaptability and robustness to texture variety [7].

We have chosen LIRA neural classifier because we have already applied it in the flat image recognition problem in microdevice assembly with good results [1]. LIRA has been successfully used in the task of handwritten digit recognition and in the task of shape recognition of microcomponents [10]. LIRA neural classifier was developed as a multipurpose classifier for image recognition problems and its high efficiency inspires us to apply it in different tasks including micromechanics.

2. Metal surface texture recognition

Texture recognition of metal surfaces provides an important tool for automation of micromechanical device assembly [8]. The assembly process requires recognition of the position and orientation of the components to be assembled [1]. It is useful to identify the surface texture of a component to recognize its position and orientation. For example, a shaft may have two polished cylinder surfaces for bearings, one of them milled with grooves for a dowel joint, and another surface turned with the lathe. It is easier to obtain the orientation of the shaft if both types of the surface textures can be recognized automatically.

There are several existing publications on fast detection and classification of defects of mechanically treated metal surfaces [15,21]. A method for surface inspection of rolling bearing metal rings based on a backpropagation neural network is proposed in Ref. [15]. In Ref. [21], a genetic algorithm-based approach for detection and quantitative evaluation of minor flaws of metal surfaces of a multi-plate clutch is proposed. Both methods require preprocessing of the initial images of metal surfaces. A brightness histogram serves as an input to the neural network in Ref. [15]. A model-based matching technique is used in Ref. [21] and the correlation between the initial image and the surface-strip model is used as a fitness function for the genetic algorithm.

The only work on texture classification of mechanically treated metal surfaces known to us is Ref. [3]. The authors propose to use a vibration-induced tactile sensor that they call dynamic touch sensor (DTS) in combination with one-layer Rosenblatt perceptron [17]. The DTS produces signals based on the vibration induced by a sensor needle sliding across a metal surface with fixed velocity and pressure. The motion path of the sensor is an arc of approximately 100°. Such motion path permits to capture information about surface in two dimensions in one sweep; however, the system is very sensitive to the changes in texture position and orientation. Spectral energy of the sensor was used as an input to the neural classifier. Metal surfaces were characterized by two characteristics: surface type and surface roughness. Surface roughness is a measure of the average height of the surface irregularities given in microinches. Six types of surfaces and six values of surface roughness were used in testing. Obtained recognition rate varied from 74.16% in recognition of two types of metal surfaces with roughness of 8 microinches to 100% in recognition of three types of metal surfaces with roughness of 250 microinches. In our experiments we achieved the recognition rate of 99.8% in recognition of four types of metal surfaces with roughness of the order of 1 microinch. In addition, our approach does not require a complex mechanical sensor and is robust to changes in texture position and orientation.

To test our texture recognition system, we created our own image database of metal surface images. Four texture classes correspond to metal surfaces after milling, polishing with sandpaper, turning with lathe and polishing with file (Fig. 1). Twenty grayscale images with resolution of 220×220 pixels were taken for each class. We randomly divided these 20 images into the training and validation sets. Fig. 1 illustrates the fact that different lighting conditions greatly affect the grayscale properties of images. The textures may also be arbitrarily oriented and not centered perfectly. Metal surfaces may have minor defects and be covered with dust. All these image properties correspond to the conditions of a real industrial environment and make the texture recognition task more complicated. Two out of four texture classes that correspond to polishing with sandpaper and polishing with file sometimes can be hardly distinguished with the naked eye (Fig. 1, columns b and d).

Our initial experiments with this image database involved a texture recognition system based on the random subspace (RSC) neural classifier. Results of the experiments were presented in Ref. [2]. The best recognition rate of 80% was obtained in recognition of three classes of metal surfaces—the class that corresponds to a metal surface after polishing with file was excluded because of its similarity with the class that corresponds to a metal surface after polishing with sandpaper. The number of images in training and validation sets were 3 and 17, respectively.

The LIRA-based texture recognition system presented in this paper gives the recognition rate of 84.42% in

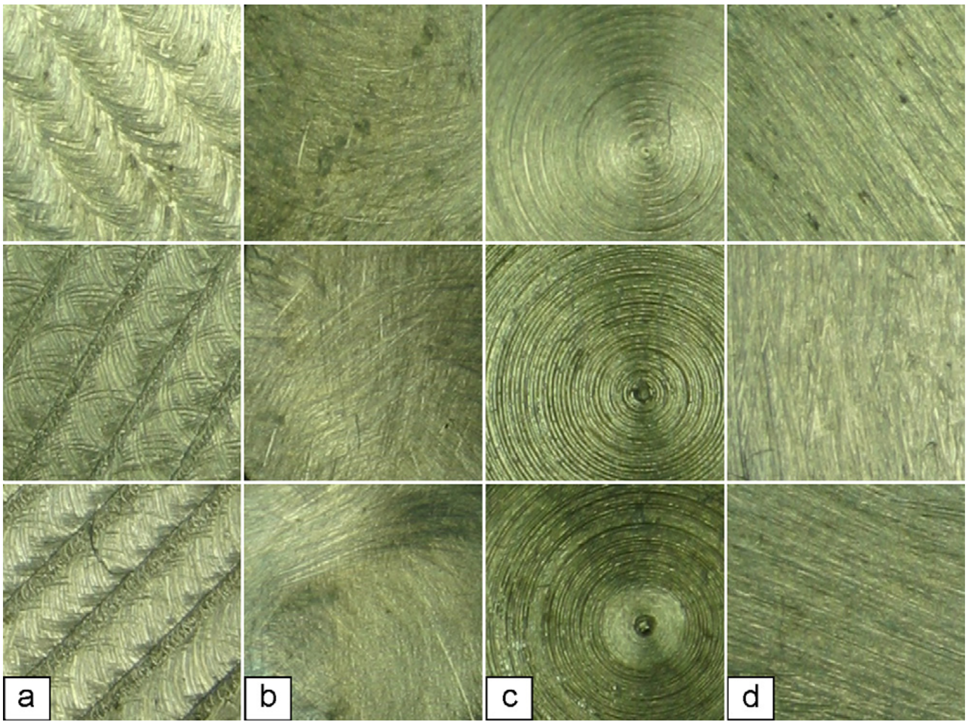


Fig. 1. Examples of metal surfaces after (columns): (a) milling, (b) polishing with sandpaper, (c) turning with lathe, and (d) polishing with file.

recognition of four classes of metal surfaces with only two images in training set and 18 images in the validation set, clearly demonstrating its superiority over the RSC classifier.

3. LIRA neural classifier

LIRA [1] neural classifier was developed on the basis of the Rosenblatt perceptron [17]. The three-layer Rosenblatt perceptron consists of the sensor S-layer, associative A-layer and the reaction R-layer. The first S-layer corresponds to the retina. In technical terms it corresponds to the input image. The second A-layer corresponds to the feature extraction subsystem. The third R-layer represents the system's output. Each neuron of this layer corresponds to one of the output classes.

The associative layer A is connected to the sensor layer S with randomly selected, non-trainable connections. The weights of these connections can be equal either to 1 (positive connection) or to -1 (negative connection). The set of these connections can be considered as a feature extractor.

A-layer consists of two-state neurons; their outputs can be equal either to 1 (active state) or to 0 (non-active state). Each neuron of the A-layer is connected to all the neurons of the R-layer. The weights of these connections are modified during the perceptron training.

We have made four major modifications in the original perceptron structure. These modifications concern random procedure of arrangement of the S-layer connections, the adaptation of the classifier to grayscale image recognition, the training procedure and the rule of winner selection.

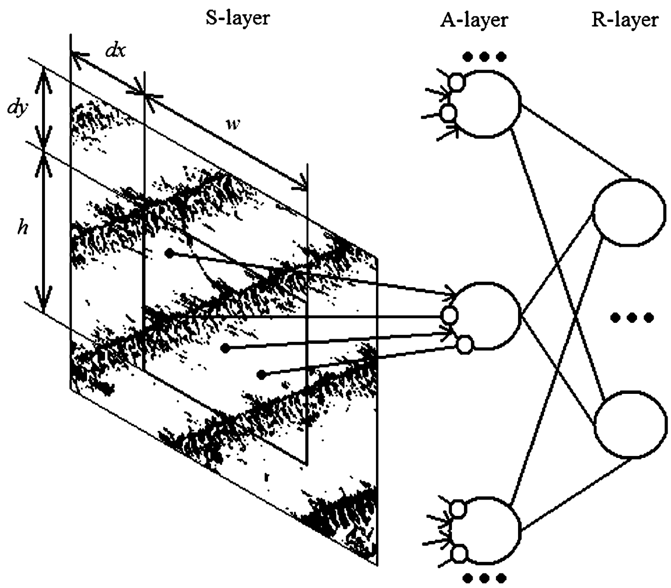


Fig. 2. Structure of the LIRA_binary neural classifier.

We propose two variants of the LIRA neural classifier: LIRA_binary and LIRA_grayscale. LIRA_binary is designed for the recognition of binary (black and white) images and LIRA_grayscale for the recognition of grayscale images. The structure of the LIRA_binary neural classifier is presented in Fig. 2 and the structure of the LIRA_grayscale in Fig. 3.

The three-layer perceptron has very good convergence, but it demands the linear separability of the classes in the parametric space. To obtain linear separability, it is

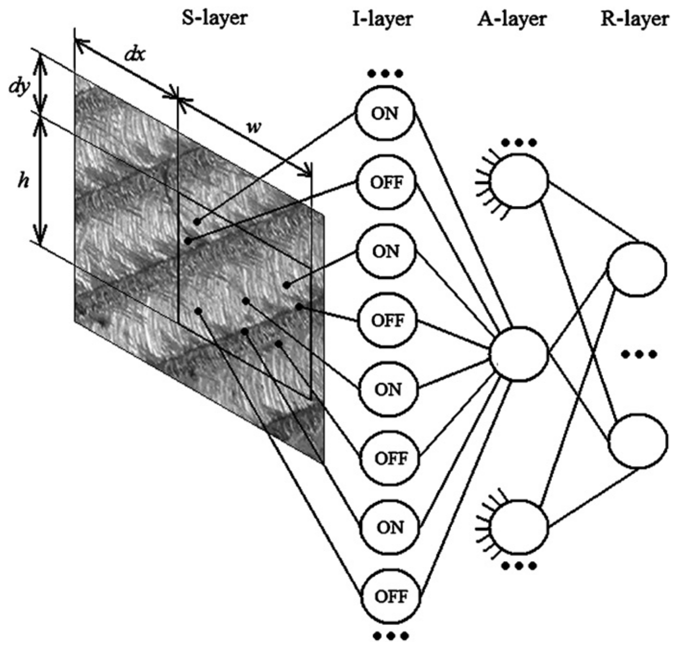


Fig. 3. Structure of the LIRA_grayscale neural classifier.

necessary to transform initial parametric space represented by pixel brightness to a parametric space of larger dimension. In our case, the connections between the S-layer and the A-layer transform initial $(W_S \times H_S)$-D space (W_S and H_S stand for width and height of the S-layer) into N-dimension space represented by binary code vector. In our experiments, $W_S = H_S = 220$ and N varied from 64,000 to 512,000. Such transformation improves linear separability.

3.1. Image coding

Each input image defines activities of the A-layer neurons in one-to-one correspondence. The binary vector that corresponds to the associative neuron activities is termed the image binary code $A = (a_1, \ldots, a_N)$, where N is the number of the A-layer neurons. The procedure that transforms an input image into corresponding binary vector A is termed the image coding.

We connect each A-layer neuron to S-layer neurons randomly selected not from the entire S-layer, but from a window $h \times w$ that 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 $[0, H_S - h]$. We create the associative neuron masks that represent the positions of connections of each A-layer neuron with neurons of the window $h \times w$. The procedure of random selection of connections is used to design the mask of A-layer neurons. This procedure starts with the selection of the upper left corner of the window $h \times w$ in which all connections of the associative neuron are located.

The following formulas are used:

$$\mathrm{d}x_i = \mathrm{random}_i(W_S - w),$$

$$\mathrm{d}y_i = \mathrm{random}_i(H_S - h),$$

where i is the position of a neuron in associative layer A, random$_i(z)$ is a random number that is uniformly distributed in the range $[0, z]$. After that position of each connection within the window $h \times w$ is defined by the pair of numbers:

$$x_{ij} = \mathrm{random}_{ij}(w),$$

$$y_{ij} = \mathrm{random}_{ij}(h),$$

where j is the number of the connection with the S-layer.

Absolute coordinates of a connection to the S-layer are defined as

$$X_{ij} = x_{ij} + \mathrm{d}x_i,$$

$$Y_{ij} = y_{ij} + \mathrm{d}y_i.$$

Random weight assignment assures good description of an image without prior knowledge about the image content. Large total number of random features assures that there will be a sufficient number of significant features that will allow separation of classes for each particular class of images. Features that do not provide useful information for separation of classes will not obtain significant weights during training.

In case of the LIRA_binary neural classifier, A-layer neurons are connected to S-layer neurons directly (Fig. 2). In this case, S-layer consists of two-state neurons; their outputs can be equal either to 1 (active state) or to 0 (non-active state). The weights of connections between the S-layer and the A-layer are randomly selected and are equal either to 1 (positive connection) or to -1 (negative connection). For example, in Fig. 2, four connections, two positive (marked with arrows) and two negative (marked with circles), are chosen within the window $h \times w$. The ith neuron of the A-layer is active ($a_i = 1$) only if all the positive connections with S-layer correspond to active neurons of the S-layer and all negative connections correspond to non-active neurons and is non-active ($a_i = 0$) in the opposite case.

To adapt the LIRA neural classifier for grayscale image recognition, we have added an additional two-state neuron layer between the S-layer and the A-layer. We term it the I-layer (intermediate layer). The structure of the LIRA_grayscale neural classifier is presented in Fig. 3.

The input of each I-layer neuron is connected to one neuron of the S-layer and the output is connected to 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. There are two types of I-layer neurons: ON-neurons and OFF-neurons. The output of an ON-neuron i is equal to 1 if its input value is larger than the threshold θ_i and is equal to 0 in the opposite case. The output of an OFF-neuron j is equal to 1 if its input value is smaller than the threshold θ_j and is equal to 0 in the opposite case. The number of ON-neurons in each group corresponds to the number of the positive connections of one A-layer neuron in the LIRA_binary structure. The number of OFF-neurons in each group corresponds to the number of the negative connections. For example, in Fig. 3, the group of eight I-layer neurons, four ON-neurons and four OFF-neurons, corresponds to one A-layer neuron. The formulas for setting connections between the S-layer and a group on the I-layer are the same as the formulas of mask design for one A-layer neuron in the LIRA_binary structure. The thresholds θ_i and θ_j are selected randomly from the range $[0, b_{\max}]$, where $b_{\max}$ is maximal brightness of the image pixels. The ith neuron of the A-layer is active ($a_i = 1$) only if outputs of all the neurons of its I-layer group are equal to 1 and is non-active ($a_i = 0$) in the opposite case.

Taking into account small number of neurons that can be active at the same time, it is convenient to represent the binary code vector not explicitly but as a set of indexes of active neurons. Let, for example, the vector A be $A = 00010000100000010000$. The corresponding set of the indexes will be $\{4, 9, 16\}$. Such compact representation of code vector permits faster calculations in the training procedure. After execution of the coding procedure, every image has an associated set of active neurons indexes.

The LIRA_binary neural classifier can potentially be used for the recognition of grayscale images after transforming them into a binary form, for example, by the following procedure:

$$\theta = \frac{2\sum_{i=1}^{W_S}\sum_{j=1}^{H_S} b_{ij}}{W_S \times H_S},$$

$$s_{ij} = \begin{cases} 1 & \text{if } b_{ij} > \theta, \\ 0 & \text{if } b_{ij} \leqslant \theta, \end{cases}$$

where θ is the threshold value, b_{ij} is the brightness of the grayscale image pixel with coordinates (i,j), s_{ij} is the brightness of the resulting binary image pixel with coordinates (i,j). However, experiments performed on the MNIST grayscale image database [1] have shown that the recognition rate obtained with the LIRA_grayscale neural classifier on original grayscale images was better than the one obtained with the LIRA_binary neural classifier on the transformed binary images. This fact justifies utilization of LIRA_grayscale on the grayscale image database of surface textures.

An image recognition system that could potentially be applied to texture recognition and was based on an approach somewhat similar to LIRA was proposed in Ref. [11]. The original images were also presented directly to the input of the neural classifier. The feature extraction subsystem was based on several layers, each composed of groups of units arranged as independent feature maps. Each feature map performed a nonlinear subsampled convolution with the kernel of a fixed size. The fundamental difference between the system in Ref. [11] and the LIRA neural classifier is that in former classifier all the connections are adaptive although heavily constrained. The feature extraction subsystem is to be trained on every particular database. In the latter classifier, only connections between the layers A and R are trainable and generality of feature extraction is achieved through the large number of random features.

3.2. Training procedure

The training procedure is identical both for LIRA_binary and LIRA_grayscale neural classifiers. Prior to training, the weights of all connections between neurons of the A-layer and the R-layer are set to 0. Then training is performed using the following algorithm:

Step 1. Present an image to the LIRA neural classifier. The image is coded and the R-layer neuron excitations E_i are computed. E_i is defined as

$$E_i = \sum_{j=1}^{N} a_j \times w_{ji},$$

where E_i is the excitation of the ith neuron of the R-layer, a_j is the output signal (0 or 1) of the jth neuron of the A-layer, w_{ji} is the weight of the connection between the jth neuron of the A-layer and the ith neuron of the R-layer.

Step 2. Robustness of the recognition is one of the important requirements the classifier must satisfy. After calculation of the neuron excitations of the R-layer, the correct class c of the image under recognition is read. The excitation E_c of the corresponding neuron of the R-layer is recalculated according to the formula:

$$E_c^* = E_c \times (1 - T_E),$$

where $0 \leqslant T_E \leqslant 1$ determines the reserve of excitation the neuron that corresponds to the correct class must have. In our experiments, the value T_E varied from 0.1 to 0.5.

Next, the neuron with the largest excitation is selected. This winner neuron represents the recognized class.

Step 3. Let us denote the winner neuron number as j keeping the number of the neuron that corresponds to the correct class denoted as c. If $j = c$, then no modification of weights is to be done. If $j \neq c$, then following modification of weights is performed:

$$w_{ic}(t+1) = w_{ic}(t) + a_i,$$

$$w_{ij}(t+1) = w_{ij}(t) - a_i,$$

where $w_{ij}(t)$ and $w_{ij}(t+1)$ are the weights of the connection between the ith neuron of the A-layer and the jth neuron of the R-layer before and after modification, a_i is the output signal (0 or 1) of the ith neuron of the A-layer.

The training process is carried out iteratively. After all, the images from the training set have been presented the total number of training errors is calculated. If this number is larger than 1% of the total number of images then the next training cycle is performed, otherwise training process is stopped. The training process is also stopped if the number of performed training cycles is larger than a predetermined value.

It is obvious that in every new training cycle the image coding procedure is repeated with the same results as in previous cycles. Therefore, in our experiments we performed the coding procedure only once and saved the sets of active neuron numbers for each image on the hard drive. During the training procedure, we used not the images, but the corresponding sets of active neurons. This approach accelerated the training process approximately by an order of magnitude.

According to Refs. [9,12], performance of a recognition system can be improved with implementation of distortions of input images during training [12] and recognition [9]. The decision whether to use distortions and selection of a particular type of distortions depends on specificity of the image database and available computational resources. In our experiments we used different combinations of horizontal, vertical and bias image translations, skewing and rotation. Since the images in our database are not centered perfectly, image translations resulted more advantageous than skewing and rotation. On the other hand, the use of image translations together with skewing and/or rotation resulted not in the higher recognition rate but in the redundant increase of the training set and corresponding increase in computational burden. We have not used scaling or squeezing distortions because all the images in the database are of the same scale.

3.3. Recognition procedure

Similar to the training procedure, the recognition procedure is identical both for LIRA_binary and LIRA_grayscale classifiers. Image distortions have been used both for training and recognition. There is an essential difference between implementation of distortions for training and recognition. In the training process, each distortion of the initial image is considered as an independent new image. In the recognition process, it is necessary to introduce a decision-making rule in order to be able to make a decision about a class of the image under recognition based on both the image itself and all of its distortions. Decision-making rule that we have used consists in calculation of the R-layer neuron excitations for all the distortions sequentially:

$$E_i = \sum_{k=0}^{d} \sum_{j=1}^{N} a_{kj} \times w_{ji},$$

where E_i is the excitation of the ith neuron of the R-layer, a_{kj} is the output signal (0 or 1) of the jth neuron of the A-layer for the kth distortion of the initial image, w_{ji} is the weight of the connection between the jth neuron of the A-layer and the ith neuron of the R-layer, d is the number of applied distortions (case $k = 0$ corresponds to the initial image).

The neuron with the largest excitation (winner neuron) represents the recognized class.

4. Results

Testing was performed on the grayscale image database described in Section 2. The database contained four texture classes that corresponded to metal surfaces after milling, polishing with sandpaper, turning with lathe and polishing with file. Images in the database were randomly divided into the training and validation sets. The number of images in training set varied from 2 to 10. All experiments were performed on the computer equipped with AMD Athlon 64 X2 4400+ dual core processor and 2.00 GB of RAM.

Parameters of the LIRA neural classifier should be tuned experimentally for specific characteristics of a given set of images; therefore, the first round of experiments was conducted to find appropriate parameter values. The most common approaches to parameter tuning include division of the database into train, test and validation sets and cross-validation techniques. In our experiments, we used holdout cross-validation, i.e. the validation set for each class was chosen randomly from the database and the rest of the database was used for training. In each experiment we performed 50 runs of the holdout cross-validation to obtain statistically reliable results. New mask of connections between the S-layer and the A-layer and new division into the training and validation sets were created for each run. The following statistics were calculated for the obtained sample of 50 error numbers: mean, standard deviation and 95% confidence interval for the mean [19]. Mean recognition rate was calculated based on the mean number of errors for one run and the total number of images in the validation set for one run. Total number of images in the validation set for one run is equal to the number of images in the validation set for each class multiplied by the number of classes (four).

The first round of experiments established the following parameter values: window $h \times w$ width $w = 10$, height $h = 10$, reserve of excitation $T_E = 0.3$. The following distortions were chosen: eight distortions for training including ± 1 pixel horizontal, vertical and bias image translations and four distortions for recognition including ± 1 pixel horizontal and vertical image translations. The number of training cycles was equal to 30.

The numbers of ON-neurons and OFF-neurons in the I-layer neuron group that corresponded to one A-layer neuron were chosen such that the ratio between the number of active neurons K and the total number of associative neurons N remains within the limits of $K = c\sqrt{N}$, where c is the constant selected experimentally from the range [1,5]. For example, for the total number of associative neurons $N = 512{,}000$ we selected three ON-neurons and five OFF-neurons.

The second round of experiments tested classification accuracy for classifiers with various total number of associative neurons, various combinations of distortions for training and recognition and different number of images in training and validation sets.

First, the total number of associative neurons N was changed from 64,000 to 512,000. The results are presented in Table 1. The amount of time needed for one run of classifier coding, training and recognition with $N = 512{,}000$ is approximately 1 min 40 s (65 s for coding, 34 s for training and 1 s for recognition). Such computational time is justified by the increase in the recognition rate and allows practical utilization of the classifier with highest recognition rate. We used $N = 512{,}000$ in all subsequent experiments.

Next, we performed experiments with different combinations of distortions for training and recognition. The results are presented in Table 2. It can be seen that distortions used in training process have great impact on the recognition rate that is no wonder if to take into account that the use of distortions for training allows to increase the size of training set nine times. Distortions used for recognition also have significant positive impact on the recognition rate.

Finally, we performed experiments with different numbers of images in the training and validation sets. The results are presented in Table 3. It can be seen that even in case of using only 2 images for training and 18 for recognition for each class, the LIRA_grayscale neural classifier achieves a reasonable recognition rate of 84.42%.

5. Discussion

LIRA neural classifier was tested in the task of texture recognition of mechanically treated metal surfaces. The classifier does not use floating point or multiplication operations. This property combined with the classifier's parallel structure allows its implementation in low cost, high-speed electronic devices. Sufficiently fast convergence of the training process and high recognition rate of 99.8% were obtained on the experimental image database. There are quite a few methods that perform well when the features used for the recognition are obtained from a training set image that has the same orientation, position and lighting conditions as the test image, but as soon as orientation, position or lighting conditions of the test image are changed the same methods perform poorly. The usefulness of methods that are not robust to such changes for practical application is very limited. The proposed method works well independently of the particular orientation and position. In this regard the results obtained in experiments are very promising.

Table 1
Dependency of the recognition rate on the total number of associative neurons

Number of associative neurons	Mean number of errors for one run/total number of images in the validation set for one run	S.D.	Confidence interval (95%)	Mean recognition rate (%)
64,000	0.34/40	0.48	[0.23, 0.45]	99.15
128,000	0.28/40	0.45	[0.17, 0.39]	99.3
256,000	0.18/40	0.39	[0.09, 0.27]	99.55
512,000	0.08/40	0.27	[0.02, 0.14]	99.8

Table 3
Dependency of the recognition rate on the number of images in training set

T/V[a]	Mean number of errors for one run/total number of images in the validation set for one run	S.D.	Confidence interval (95%)	Mean recognition rate (%)
2/18	11.22/72	5.43	[9.93, 12.51]	84.42
4/16	2.46/64	3.11	[1.72, 3.2]	96.16
6/14	0.48/56	1.02	[0.24, 0.72]	99.14
8/12	0.1/48	0.36	[0.02, 0.19]	99.79
10/10	0.08/40	0.27	[0.02, 0.14]	99.8

[a] T is the size of the training set for each class, V is the size of the validation set for each class.

Table 2
Dependency of the recognition rate on the distortions

Distortions		Mean number of errors for one run/total number of images in the validation set for one run	S.D.	Confidence interval (95%)	Mean recognition rate (%)
Training	Recognition				
–	–	25.74/40	3.57	[24.89,26.59]	35.65
–	+	24.8/40	3.19	[24.04,25.56]	38.0
+	–	0.26/40	0.44	[0.16, 0.36]	99.35
+	+	0.08/40	0.27	[0.02, 0.14]	99.8

6. Conclusion

This paper continues the series of publications on automation of microassembly processes [1,8].

The LIRA neural classifier is proposed for texture recognition of mechanically treated metal surfaces. The proposed approach does not require the use of a complex mechanical sensor and is robust to changes in texture position and orientation. It may be applied in systems that have to recognize position and orientation of complex work pieces in the task of assembly of micromechanical devices as well as in surface quality inspection systems.

The performance of the proposed classifier was tested on specially created image database in recognition of four texture types that correspond to metal surfaces after milling, polishing with sandpaper, turning with lathe and polishing with file. The promising recognition rate of 99.8% was obtained.

Acknowledgment

The authors gratefully acknowledge E. Kussul, National Autonomous University of Mexico (UNAM), for the constructive discussions and helpful comments.

References

[1] T. Baidyk, E. Kussul, O. Makeyev, A. Caballero, L. Ruiz, G. Carrera, G. Velasco, Flat image recognition in the process of microdevice assembly, Pattern Recogn. Lett. 25 (1) (2004) 107–118.

[2] T. Baidyk, E. Kussul, O. Makeyev, Texture recognition with random subspace neural classifier, WSEAS Trans. Circuits Syst. 4 (4) (2005) 319–325.

[3] D. Brenner, J.C. Principe, K.L. Doty, Neural network classification of metal surface properties using a dynamic touch sensor, in: Proceedings of the IJCNN'91 International Joint Conference on Neural Networks, vol. 1, IEEE, Seattle, 1991, pp. 189–194.

[4] C.-H. Chan, G.K.H. Pang, Fabric defect detection by Fourier analysis, IEEE Trans. Ind. Appl. 36 (5) (2000) 1267–1276.

[5] C.R. Friedrich, M.J. Vasile, Development of the micromilling process for high-aspect-ratio microstructures, J. Microelectromech. Syst. 5 (1996) 33–38.

[6] L. Hepplewhite, T.J. Stonham, Surface inspection using texture recognition, in: Proceedings of the IAPR'94 12th International Conference on Pattern Recognition, vol. 1, IEEE, Jerusalem, 1994, pp. 589–591.

[7] M.A. Kraaijveld, An experimental comparison of nonparametric classifiers for time-constrained classification tasks, in: Proceedings of the ICPR'98 14th International Conference on Pattern Recognition, vol. 1, IEEE, Brisbane, 1998, pp. 428–435.

[8] E. Kussul, T. Baidyk, L. Ruiz-Huerta, A. Caballero, G. Velasco, L. Kasatkina, Development of micromachine tool prototypes for microfactories, J. Micromech. Microeng. 12 (2002) 795–813.

[9] E. Kussul, T. Baidyk, Improved method of handwritten digit recognition tested on MNIST database, Image Vision Comput. 12 (2004) 971–981.

[10] E. Kussul, T. Baidyk, D. Wunsch, O. Makeyev, A. Martin, Image recognition systems based on random local descriptors, in: Proceedings of the IJCNN'06 International Joint Conference on Neural Networks, IEEE, Vancouver, 2006, pp. 4722–4727.

[11] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.J. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (1989) 541–551.

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

[13] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vision 43 (1) (2001) 29–44.

[14] A.M. Madni, L.A. Wan, Microelectromechanical systems (MEMS): an overview of current state-of-the art, in: Proceedings of the IEEE Aerospace Conference, vol. 1, IEEE, Snowmass, 1998, pp. 421–427.

[15] C. Neubauer, Fast detection and classification of defects on treated metal surfaces using a backpropagation neural network, Neural Networks 2 (1991) 1148–1153.

[16] N. Ooyama, S. Kokaji, M. Tanaka, et al, Desktop machining microfactory, in: Proceedings of the IWMF'00 second International Workshop on Microfactories, FSRM, Fribourg, 2000, pp. 14–17.

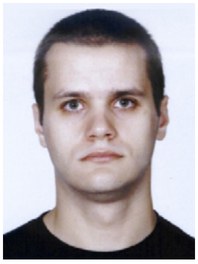
[17] F. Rosenblatt, Principles of Neurodynamics, Spartan Books, New York, 1962.

[18] R. Sanchez-Yanez, E. Kurmyshev, A. Fernandez, One-class texture classifier in the CCR feature space, Pattern Recogn. Lett. 24 (2003) 1503–1511.

[19] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC Press, Boca Raton, 2003.

[20] W.S. Trimmer (Ed.), Micromechanics and MEMS: Classic and Seminal Papers to 1990, IEEE Press, New York, 1997.

[21] J. Zhu, M. Minami, Finding and quantitative evaluation of minute flaws on metal surface using hairline, in: Proceedings of the IECON'04 30th Annual Conference of IEEE Industrial Electronics Society, vol. 2, IEEE, Busan, 2004, pp. 1240–1245.



Oleksandr Makeyev was born in Kiev, Ukraine, in 1983. He received B.S. in mathematics and M.S. in statistics from National Taras Shevchenko University of Kyiv, Kiev, Ukraine, in 2003 and 2005, respectively. He joined the Department of Electrical and Computer Engineering, Clarkson University, Potdsam, NY, USA, in 2006, and is currently working toward his Ph.D. His research interests include bioengineering, neural networks, pattern recognition, evolutionary computation, wavelets and time series analysis.

Edward Sazonov received the Ph.D. degree in computer engineering and the M.S. degree in electrical engineering from West Virginia University, Morgantown, WV, USA in 2002 and 1999, respectively. He received the diploma of Systems Engineer from Khabarovsk State University of Technology, Khabarovsk, Russia, in 1993. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, Clarkson University, NY, USA. His research interests are focused on ambient intelligent systems with applications in bio- and rehabilitation engineering such as self-powered wearable sensors for collection of physiological data; non-invasive measurement of energy intake and expenditure in free living population; sleep state scoring from non-invasive measurements and biofeedback devices for people with impaired proprioceptive abilities.

Tatiana Baidyk was born in March 1954. She received the M.S. in electrical engineering from the Kiev Polytechnic Institute, Ukraine in 1977, and the Ph.D. in control systems for mobile robots in 1983 from the Institute of Cybernetics of the Ukrainian Academy of Sciences. From 1977 to 1995 she was with the Institute of Cybernetics of the Ukrainian Academy of Sciences as a post-graduate student from 1977, as a junior researcher from 1980, and as a senior researcher from 1990. In 1994, T. Baidyk received D.S. in the area of neural networks from the Institute of Cybernetics of Ukrainian Academy of Sciences. From 1995 to 1999 she was coordinator of the International Program of Education and Science in Ukraine, International Renaissance Foundation. From 1998 she was with the International Research and Training Center of UNESCO/IIP of Information Technologies and Systems as a leading researcher. From 2001 she is Researcher level "A" in UNAM, Mexico. She has published over 130 papers, authored and coauthored two books, three patents, two author's rights, and edited eight other books. She participated in international projects INTAS, ISF, NSF-CONACYT, worked in 1992 in the international group in Japan creating neurocomputer B-512. Baidyk's present research interests are neural networks, pattern recognition, control systems, and industrial applications. Dr. Baidyk is a member of the European Council of Artificial Intelligence, Association of Artificial Intelligence (Russia), a member of the Association of Developers and Users of Intelligent Systems (Ukraine), a member of the Executive Committee of the Neural Network Society (RNNS, Russia).

Anabel Martín received the B.S. degree in computer science from the Autonomous University of Yucatán (UADY), Mérida, Yucatán, Mexico, in 2004. She received the M.S. degree in computer science from the National Autonomous University of Mexico (UNAM), Mexico City, Mexico, in 2006. Her primary research interests are computer vision and neural networks.