Limited Receptive Area neural classifier for texture recognition of metal surfaces

Oleksandr Makeyev[1], Tatiana Baidyk[2] and Anabel Martín[2]
1 National Taras Shevchenko University of Kyiv,
64, Volodymyrska Str., 01033, Kiev, Ukraine
mckehev@hotmail.com
2 Center of Applied Sciences and Technological Development, National Autonomous University of Mexico, Cd. Universitaria, Circuito Exterior s/n, Coyoacán, 04510, México, D.F., Mexico
tbaidyk@aleph.cinstrum.unam.mx; anabelmartin@lycos.com

Abstract. The Limited Receptive Area (LIRA) neural classifier is proposed for texture recognition of mechanically treated metal surfaces. It can be used in systems that have to recognize position and orientation of complex work pieces in the task of assembly of micromechanical devices. 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 99.7% was obtained.

1 Introduction

The main approaches to microdevice production are the technology of micro electromechanical systems (MEMS) [1, 2] and microequipment technology (MET) [3-6]. To get the best of these technologies it is important to have advanced image recognition systems.

Texture recognition systems are widely used in industrial inspection, for example, in textile industry for detection of fabric defects [7], in electronic industry for inspection of the surfaces of magnetic disks [8], in decorative and construction industry for inspection of polished granite and ceramic titles [9], etc.

Numerous approaches were developed to solve the texture recognition problem. Many statistical texture descriptors are based on a generation of co-occurrence matrices. In [8] the texture co-occurrence of n-th rank was proposed. The matrix contains statistics of the pixel under investigation and its surrounding. Another approach was proposed in [9]. The authors proposed the coordinated cluster representation (CCR) as a technique of texture feature extraction. The underlying principle of the CCR is to extract a 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 prove this approach to be very promising. Leung et al. [10] proposed textons (representative texture elements) for texture description and recognition. The vocabulary of textons corresponds to the characteristic features of the image. There are many works on applying neural networks in texture recognition problem [11, 12].

In this paper we propose the LIRA neural classifier [4] for metal surface texture recognition. Four types of metal surfaces after mechanical treatment were used to test the proposed 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 metal surface images that it is necessary to recognize in industry have all these problems and what is more there are some problems specific for industrial environment, for example, metal surface can have dust on it.

The reason to choose a system based on neural network architecture for the current task was that such systems have already proved their efficacy in texture recognition due to significant properties of adaptability and robustness to texture variety [13].

We have chosen the LIRA neural classifier because we have already applied it in the flat image recognition problem in microdevice assembly and the results were very promising [4]. We have also tested it in handwritten digit recognition task and its recognition rate on the MNIST database was 0.55% [4] that is among the best results obtained on this database.

2 Metal surface texture recognition

The task of metal surface texture recognition is important to automate the assembly processes in micromechanics [3]. To assembly a device it is necessary to recognize the position and orientation of the work pieces to be assembled [4]. It is useful to identify the surface of a work piece to recognize its position and orientation. For example, let the shaft have two polished cylinder surfaces for bearings, one of them milled with grooves for dowel joint, and the other one turned with the lathe. It will be easier to obtain the orientation of the shaft if we can recognize both types of the surface textures.

There are works on fast detection and classification of defects on treated metal surfaces using a back propagation neural network [14], but we do not know any on texture recognition of metal surfaces after mechanical treatment.

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). It can be seen that different lighting conditions affect greatly the grayscale properties of the images. The textures may also be arbitrarily oriented and not centered perfectly. Metal surfaces may have minor defects and dust on it. All this image properties correspond to the conditions of the real industrial environment and make the texture recognition task more complicated. Two out of four texture classes that correspond to polishing with sandpaper and to polishing with file sometimes can be hardly distinguished with the naked eye (Fig. 1, columns *b* and *d*).

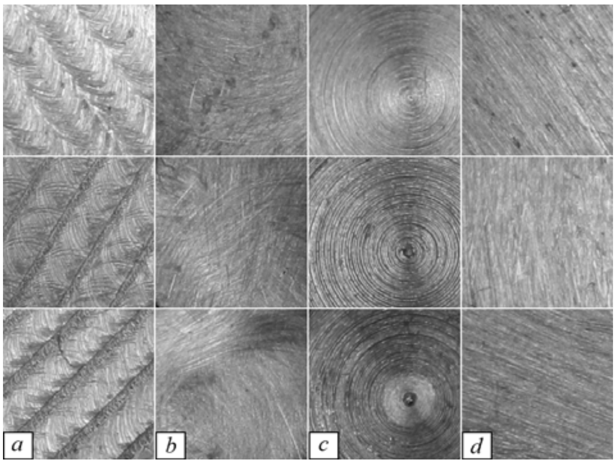

Fig. 1. Examples of metal surfaces after (columns): a) milling, b) polishing with sandpaper, c) turning with lathe, d) polishing with file

3 The LIRA neural classifier

The LIRA neural classifier [4] was developed on the basis of the Rosenblatt perceptron [15]. 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 the 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 2-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.

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

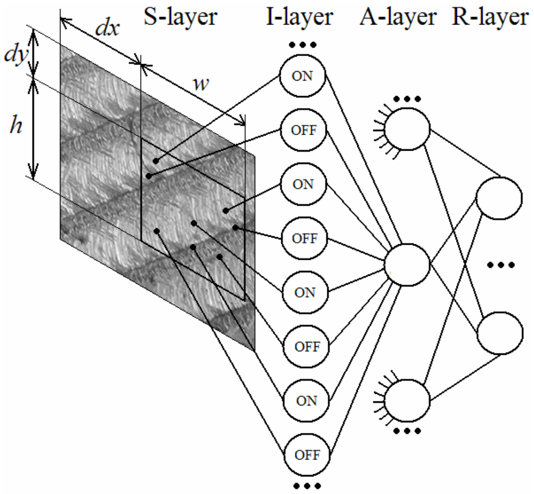

Fig. 2. The structure of the LIRA_grayscale neural classifier

The one-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 necessary to transform the initial parametric space represented by pixel brightness to the parametric space of larger dimension. In our case the connections between the *S*-layer and the *A*-layer transform initial ($W_S \cdot H_S$)-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 the linear separability. The coding procedure used in the LIRA classifier is the following.

3.1 Image coding

Each input image defines the 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 the input image into the 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 the window $h \cdot w$ 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 $h \cdot w$. The procedure of random selection of connections is used to design the mask of *A*-layer neuron. This procedure starts with the selection of the upper left corner of the window $h \cdot w$ in which all connections of the associative neuron are located.

The following formulas are used:

$dx_i = random_i\,(W_S - w)$,

$dy_i = random_i\,(H_S - h)$,

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 $h \cdot w$ is defined by the pair of numbers:

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

$y_{ij} = random_{ij}\,(h)$,

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

Absolute coordinates of the connection on the retina are defined by the pair of the numbers:

$X_{ij} = x_{ij} + dx_i$,

$Y_{ij} = y_{ij} + dy_i$.

To adapt the LIRA neural classifier for grayscale image recognition we have added the additional 2-state neuron layer between the *S*-layer and the *A*-layer. We term it the *I*-layer (intermediate layer, see Fig. 2).

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. There are two types of *I*-layer neurons: ON-neurons and OFF-neurons. The output of the ON-neuron i is equal to 1 when its input value is larger than the threshold θ_i and it is equal to 0 in opposite case. The output of the OFF-neuron j is equal to 1 when its input value is smaller than the threshold θ_j and it is equal to 0 in opposite case. For example, in Fig. 2, the group of eight *I*-layer neurons, four ON-neurons and four OFF-neurons, corresponds to one *A*-layer neuron. The thresholds θ_i and θ_j are selected randomly from the range [0, $\eta \cdot b_{max}$], where b_{max} is maximal brightness of the image pixels, η is the parameter selected experimentally from the range [0, 1]. The i-th 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 opposite case.

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

A = 00010000100000010000.

The corresponding list of the numbers of active neurons will be 4, 9, and 16. Such compact representation of code vector permits faster calculations in training procedure. Thus, after execution of the coding procedure every image has a corresponding list of numbers of active neurons.

3.2 Training procedure

Before starting the training procedure the weights of all connections between neurons of the *A*-layer and the *R*-layer are set to 0. As distinct from the Rosenblatt perceptron our LIRA neural classifier has only non-negative connections between the *A*-layer and the *R*-layer.

The first stage. The training procedure starts with the presentation of the first 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:

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

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

The second stage. 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:

$$E_c* = E_c \cdot (1 - T_E),$$

where $0 \leq T_E \leq 1$ 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.

After that we select the neuron with the largest excitation. This winner neuron represents the recognized class.

The third stage. 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 nothing is to be done. If $j \neq c$ then following modification of weights is to be done:

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

$$w_{ij}(t+1) = w_{ij}(t) - a_i, \text{ if } (w_{ij}(t+1) < 0) \text{ then } w_{ij}(t+1) = 0,$$

where $w_{ij}(t)$ and $w_{ij}(t + 1)$ are the weights of the connection between the *i*-th neuron of the *A*-layer and the *j*-th neuron of the *R*-layer before and after modification, a_i is the output signal (0 or 1) of the *i*-th 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 one percent 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 more than a predetermined value.

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

It is known [16] that the performance of the recognition systems can be improved with implementation of distortions of the input image during the training process. In our experiments we used different combinations of horizontal, vertical and bias image shifts, skewing and rotation.

3.3 Recognition procedure

In our LIRA neural classifier we use image distortions not only in training but also in recognition process. 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 rule of decision-making in order to be able to make a decision about a class of the image under recognition based on the mutual information about this image and all its distortions. The rule of decision-making that we have used consists in calculation of the *R*-layer neuron excitations for all the distortions sequentially:

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

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

After that we select the neuron with the largest excitation. This winner neuron represents the recognized class.

4 Results

To test our texture recognition system we created our own image database of mechanically treated metal surfaces (see Section 2 for details). We work with four texture classes that correspond to metal surfaces after: milling, polishing with sandpaper, turning with lathe and polishing with file. 20 grayscale images of 220x220 pixels were taken for each class. We randomly divide these 20 images into the training and test sets for the LIRA_grayscale neural classifier. The number of images in training set varied from 2 to 10 images for each class.

All experiments were performed on a Pentium 4, 3.06 GHz computer with 1.00 GB RAM.

We carried out a large amount of preliminary experiments first to estimate the performance of our classifier and to tune the parameter values. On the basis of these preliminary experiments we selected the best set of parameter values and carried out final experiments to obtain the maximal recognition rate. In preliminary experiments the following parameter values were set: window $h \cdot w$ width $w = 10$, height $h = 10$, parameter that determines the reserve of excitation the neuron that corresponds to the correct class must have $T_E = 0.3$. The following distortions were chosen for the final experiments: 8 distortions for training including 1 pixel horizontal, vertical and bias image shifts and 4 distortions for recognition including 1 pixel horizontal and vertical image shifts. 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 in order to keep the ratio between the number of active neurons *K* and the total number of associative neurons *N* within the limits of $K = c \cdot \sqrt{N}$, where *c* is the constant selected experimentally from the range [1, 5]. This ratio corresponds to neurophysiological data. The number of active neurons in the cerebral cortex is hundreds times less than the total number of neurons. For example, for the total number of associative neurons *N* = 512,000 we selected three ON-neurons and five OFF-neurons.

In each experiment we performed 50 runs to obtain statistically reliable results. That is, the total number of recognized images was calculated as number of images in test set for one run multiplied by 50. New mask of connections between the *S*-layer and the *A*-layer and new division into the training and test sets were created for the each run.

In the first stage of final experiments we changed the total number of associative neurons *N* from 64,000 to 512,000. The results are presented in Table 1. Taking into account that the amount of time needed for 50 runs of coding and classifier's training and recognition with *N* = 512,000 is approximately 3 h and 20 min we can conclude that such computational time is justified by the increase in the recognition rate. That is why we used *N* = 512,000 in all the posterior experiments.

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

Total number of associative neurons	Number of errors / Total number of recognized images	% of correct recognition
64,000	20 / 2000	99
128,000	13 / 2000	99.35
256,000	8 / 2000	99.6
512,000	6 / 2000	99.7

In the second stage of final experiments 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 8 distortions for training allows to increase the size of training set 9 times. Distortions used in recognition process also have significant positive impact on the recognition rate.

Table 2. Dependency of the recognition rate on the distortions

Distortions		Number of errors / Total number of recognized images	% of correct recognition
Training	Recognition		
-	-	1299 / 2000	35.05
-	+	1273 / 2000	36.35
+	-	14 / 2000	99.3
+	+	6 / 2000	99.7

In the third stage of final experiments we performed experiments with different numbers of images in the training and test sets. The results are presented in Table 3. The note tr./t. reflects how many images were used for training (tr.) and how many for testing (t.). It can be seen that even in case of using only 2 images for training and 18 for recognition the LIRA_grayscale neural classifier gives a good recognition rate of 83.39%.

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

tr./t.	Number of errors / Total number of recognized images	% of correct recognition
2/18	598 / 3600	83.39
4/16	174 / 3200	94.56
6/14	34 / 2800	98.78
8/12	8 / 2400	99.67
10/10	6 / 2000	99.7

5 Discussion

The LIRA neural classifier was tested in the task of texture recognition of mechanically treated metal surfaces. This 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 very promising recognition rate of 99.7% were obtained on the specially created image database (see Section 2 for details). 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 or position or lighting conditions of the test image is changed with respect to the one in the training set the same methods will perform poorly. The usefulness of methods that are not robust to such changes is very limited and that is the reason for developing of our texture classification system that works well independently of the particular orientation, position and lighting conditions. In this regard the results obtained in experiments are very promising.

6 Conclusion

This paper continues the series of works on automation of micro assembly processes [3, 4].

The LIRA neural classifier is proposed for texture recognition of mechanically treated metal surfaces. It can be used 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 99.7% was obtained.

Acknowledgment

The authors gratefully acknowledge Dr. Ernst Kussul, National Autonomous University of Mexico, for constructive discussions and helpful comments.

This work was supported by projects PAPIIT 1112102, PAPIIT IN116306-3, PAPIIT IN108606-3, NSF-CONACYT 39395-A.

References

1. W.S. Trimmer (ed.), Micromechanics and MEMS. Classical and Seminal Papers to 1990 (IEEE Press, New York, 1997).
2. A.M. Madni, L.A. Wan, Micro Electro Mechanical Systems (MEMS): an Overview of Current State-of-the Art, Aerospace Conference **1**, 421-427 (1998).
3. E. Kussul, T. Baidyk, L. Ruiz-Huerta, A. Caballero, G. Velasco, L. Kasatkina, Development of Micromachine Tool Prototypes for Microfactories, J. Micromech. Microeng. **12**, 795-813 (2002).
4. 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), 107-118 (2004).
5. C.R. Friedrich, M.J. Vasile, Development of the Micromilling Process for High- aspect-ratio Micro Structures, J. Microelectromech. S. **5**, 33-38 (1996).
6. Naotake Ooyama, Shigeru Kokaji, Makoto Tanaka, et al, Desktop Machining Microfactory, Proc. of the 2-nd International Workshop on Microfactories, 14-17 (2000).
7. Chi-ho Chan, Grantham K.H. Pang, Fabric Defect Detection by Fourier Analysis, IEEE T. Ind. Appl. **36**(5), 1267-1276 (2000).
8. L. Hepplewhite, T.J. Stonham, Surface Inspection Using Texture Recognition, Proc. of the 12th IAPR International Conference on Pattern Recognition **1**, 589-591 (1994).
9. R. Sanchez-Yanez, E. Kurmyshev, A. Fernandez, One-class Texture Classifier in the CCR Feature Space, Pattern Recogn. Lett. **24**, 1503-1511 (2003).
10. T. Leung, J. Malik, Representing and Recognizing the Visual Appearance of Materials Using Three-dimensional Textons, Int. J. Comput. Vision **43**(1), 29-44 (2001).
11. M.A. Mayorga, L.C. Ludeman, Shift and Rotation Invariant Texture Recognition with Neural Nets, Proc. of the IEEE International Conference on Neural Networks **6**, 4078-4083 (1994).
12. Woobeom Lee, Wookhyun Kim, Self-organization Neural Networks for Multiple Texture Image Segmentation, Proc. of the IEEE Region 10 Conference TENCON 99 **1**, 730-733 (1999).
13. M.A. Kraaijveld, An Experimental Comparison of Nonparametric Classifiers for Time-constrained Classification Tasks, Proc. of the Fourteenth International Conference on Pattern Recognition **1**, 428-435 (1998).
14. C. Neubauer, Fast Detection and Classification of Defects on Treated Metal Surfaces Using a Back Propagation Neural Network, Proc. of the IEEE International Joint Conference on Neural Networks **2**, 1148-1153 (1991).
15. F. Rosenblatt, Principles of Neurodynamics (Spartan books, New York, 1962).
16. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based Learning Applied to Document Recognition, P. IEEE **86**(11), 2278-2344 (1998).