# Texture Recognition with Random Subspace Neural Classifier

BAIDYK T., KUSSUL E., MAKEYEV O.*,
Center of Applied Science and Technological Development (CCADET),
National Autonomous University of Mexico (UNAM),
Cd. Universitaria,
04510, Mexico, D.F.
MEXICO

* Kyiv National Taras Shevchenko University,
64, Volodymyrska str.,
01033 Kiev,
UKRAINE

*Abstract:* - The Random Subspace Neural Classifier (RSC) for the texture recognition is proposed. This system was developed and used for image recognition in micromechanics. It permits us to recognize different types of metal surfaces after mechanical processing. At the first stage the different samples of milling, turning and polishing with sandpaper surfaces were used to test the developed system.

*Key-words :* - Texture recognition, Random Subspace Neural Classifier (RSC), micromechanics.

## 1 Introduction

The main approaches to microdevices production are the technology of micro electromechanical systems (MEMS) [1], [2] and microequipment technology (MET) [3]-[7]. To get the better of these technologies it is important to have advanced image recognition systems.

The task of classification in recognition systems is more important issue than clustering or unsupervised segmentation in a vast majority of applications [8]. The texture classification plays an important role in outdoor scene images recognition. Despite its potential importance, there does not exist a formal definition of texture due to an infinite diversity of texture samples. There exist a large number of texture analysis methods in the literature.

On the base of the texture classification Castano et al. obtained satisfactory results for real-world images relevant to navigation on cross-country terrain [8]. They had four classes: soil, trees, bushes/grass, and sky. This task was elected by Pietikäinen et al. to test their system for texture recognition [9]. In this case new database was done. Five texture classes were defined: sky, trees, grass, road and buildings. Due to perceptible changes of illumination, the following sub-classes were used: trees in the sun, grass in the sun, road in the sun, and buildings in the sun. They achieved a very good accuracy of 85.43%. In 1991 we solved the same task [10], [11]. We worked with five textures (sky, trees/crown, road, transport means, and post/trunk). The images were taken in the streets of the city. We used brightness, contrast and contour orientation histograms as input to our system (74 features). We used associative-projective neural networks for recognition [10], [11]. The recognition rate was 79.9 %. In Fig.1 two examples of such images are presented. In 1996 Goltsev A. developed an assembly neural network for texture segmentation [12], [13] and used it for real scene analysis. Texture recognition algorithms are used in different areas, for example, in textile industry for detection of fabric defects [14]. In electronic industry texture recognition is important to characterize the microstructure of metal films deposited on flat substrates [15], in the task of automation of visual inspection of magnetic disks as a quality control [16]. Texture recognition is used for foreign object detection (for example, contaminants in food, such as pieces of stone, fragments of glass, etc.) [17]. Aerial texture classification is applied to resolve difficult figure-ground separation problem [18]. In this work we propose the neural classifier RSC for metal surface texture recognition. Examples of such metal surfaces are presented in Fig.2.

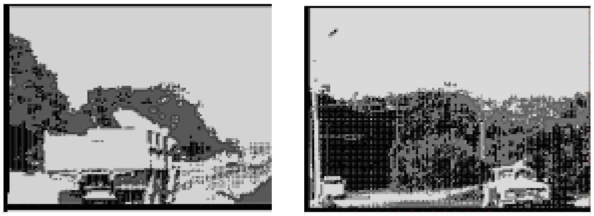

Fig. 1. Samples of real-world images

Due to the changes in viewpoint and illumination, the visual appearance of different surfaces can vary greatly, which makes their recognition very difficult [9]. Different lighting conditions and viewing angles greatly affect the gray scale properties of an image due to such effects as shading, shadowing or local occlusions. The real surface images which it is necessary to recognize in industrial environment have all these problems and more, for example, sometimes the surface can have dust on it.

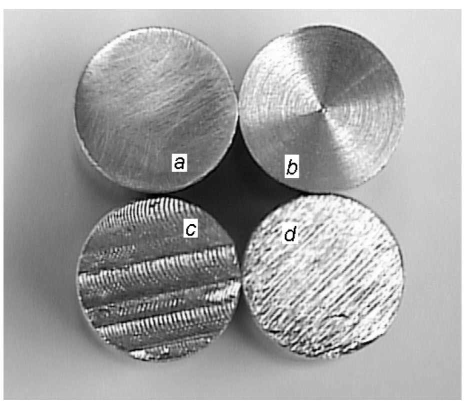


Fig. 2. Samples of metal surface after:
a) sandpaper polishing, b) turning,
c) milling, d) polishing with file

Different approaches were developed to solve the problem of texture recognition. Leung et al. [20] proposed textons (representative texture elements) for texture describing and recognition. The vocabulary of textons corresponds to the characteristic features of the image. They tested this algorithm on Columbia-Utrecht Reflectance and Textures (CUReT) database [21], [22]. This approach has disadvantages: it needs many parameters to be set manually and it is computationally complex.

Other approach is based on the micro-textons which are extracted by means of multiresolution local binary pattern operator (LBP). LBP is a gray-scale invariant primitive statistic of texture. This method was tested on CUReT database and performed well both in experiments and analysis of outdoor scene images.

Many statistical texture descriptors were based on a generation of co-occurrence matrices. In [16] the texture co-occurrence of *n*-th order rank was proposed. This matrix contains statistics of the pixel under investigation and its surrounding pixels. Co-occurrence operator can be used to map the binary image too. For example, in [23], the method to extract texture features in terms of the occurrence of *n* conjoint pixel values was combined with a single layer neural network. There are many investigations in application of neural network for texture recognition [24], [25]. To test the developed system [25] texture images from [26] were used.

The reasons to choose a system based on neural network architecture are its significant properties of adaptiveness and robustness to texture variety.

## 2 Metal surface texture recognition

The task of metal surface texture recognition is important to automate the assembly processes in micromechanics [3]. To assembly any device it is necessary to recognize the position and orientation of the work pieces to be assembled [4]. It is useful to identify 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 by the lathe. It will be easier to obtain the orientation of the shaft if we can recognize both types of the surface textures.

Our texture recognition system has the following structure (Fig. 3).

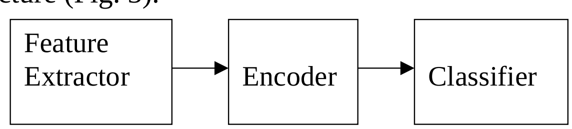


Fig.3. Structure of RSC Neural Classifier

The texture image serves as input data to the feature extractor. The extracted features are presented to the input of encoder. The encoder produces the output binary vector of large dimension, which is presented to the input of one-layer neural classifier. The output of the classifier gives the recognized class. Further, we will describe all these blocks in detail.

To test our neural classifier RSC we created our own test set of metal surface images. We work with three texture classes. Every class contains 20 images. From these 20 images we randomly selected a part for training of our neural classifier RSC and the rest of images we used to test our system. Number of images in training set varied from 3 to 10.

The first texture class corresponds to metal surface after milling (Fig. 4), the second texture class corresponds to metal surface after polishing with sandpaper (Fig. 5), and the third texture class corresponds to metal surface after turning with lathe (Fig. 6). You can see that different lighting conditions affect greatly the gray-scale properties of an image. The texture may also be arbitrarily oriented which makes the texture recognition task more complicated.

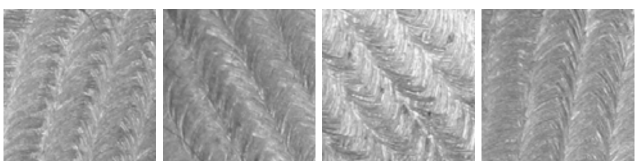

Fig. 4. Images of metal surface after milling

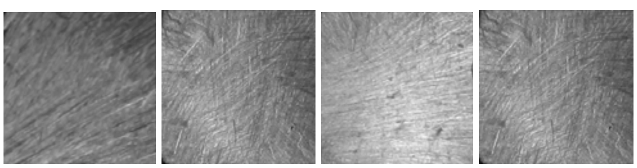

Fig. 5. Images of metal surface after polishing with sandpaper

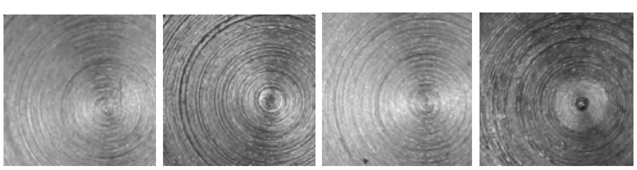

Fig. 6. Images of metal surface after turning with lathe

There are works on fast detection and classification of defects on treated metal surfaces using a back propagation neural network [27], but we do not know any on texture recognition of metal surfaces after different mechanical treatments.

Solving this problem can help us to recognize the positions and orientations of complex mechanically processed work pieces.

RSC neural classifier that was used in our system is based on the Random Threshold Classifier (RTC) developed earlier [19].

## 3 Random Threshold Neural Classifier

RTC neural classifier was developed and tested in 1994 [19]. The architecture of RTC is shown in Fig.7

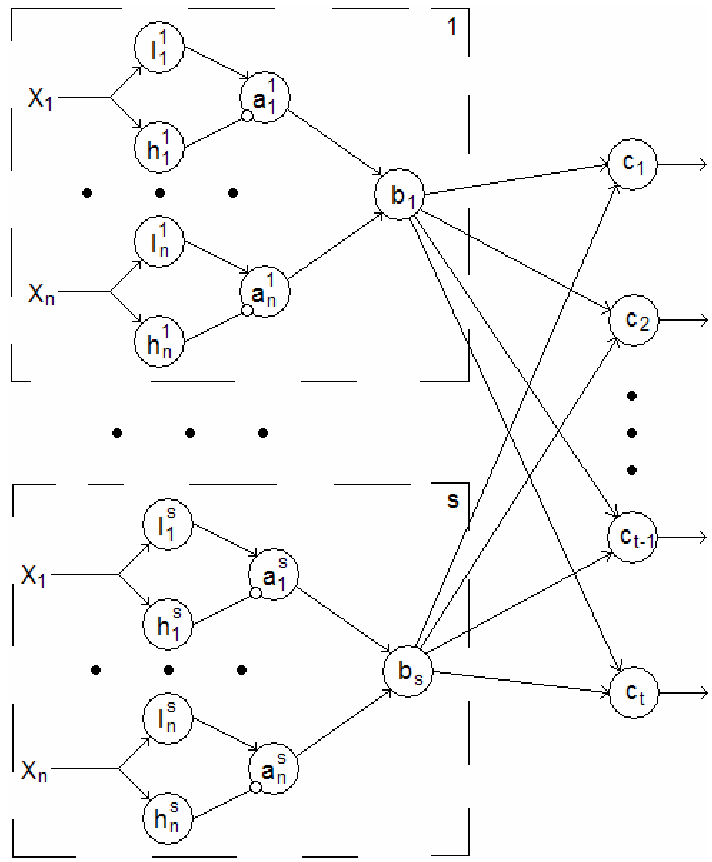


Fig. 7. Structure of RTC neural classifier

The neural network structure consists of *s* blocks, each block with one output neuron ($b_1$, ..., $b_s$). The set of features ($X_1$, ..., $X_n$) input to every block. Every feature $X_i$ input to two neurons $h_i^j$ and $l_i^j$, where $i$ ($i$ =1, 2, ..., $n$) represents the number of features, and $j$ ($j$ = 1, 2, ..., $s$) represents the number of neural blocks. The threshold of $l_i^j$ is less than the threshold of $h_i^j$. The values of thresholds are randomly selected once and fixed. The output of neuron $l_i^j$ is connected with the excitatory input of the neuron $a_i^j$, and the output of the neuron $h_i^j$ is connected with the inhibitory input of the neuron $a_i^j$. In the output of the neuron $a_i^j$, the signal appears only if input signal from $l_i^j$ is equal to 1, and input signal from $h_i^j$ is equal to 0. All outputs from neurons $a_i^j$, in one block $j$, are inputs of the neuron $b_j$, which presents the output of the whole neuron block.

The output of the neuron $b_j$ is 1 only if all neurons $a_i^j$ in the block $j$ are excited. The output of every neuron block is connected with trainable connections to all the inputs of output layer of classifier ($c_1$, ..., $c_t$) where $t$ is a number of classes. The classifier works in two regimes: training and recognition. We use the perceptron rule to change the connection weights during the training process.

The geometrical interpretation can help us to explain the discussed principles. Let us consider the case with two features $X_1$ and $X_2$ (Fig. 8).

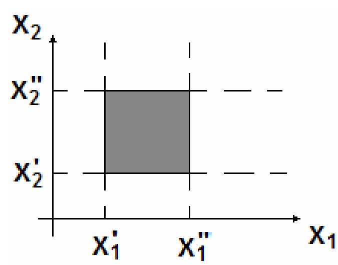


Fig. 8. Geometrical interpretation of the neuron

The neuron $b_1$ will be active when input feature point is located inside the rectangle shown in Fig. 8.

Since there are many blocks (1, …, $s$), the whole feature space will be covered by many rectangles of different size and location (Fig. 9).

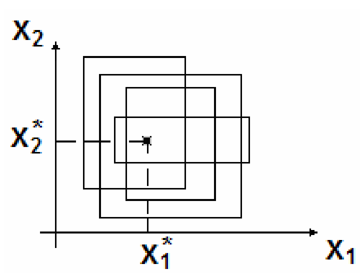


Fig. 9. Geometrical interpretation of the neural classifier

In a multidimensional space, instead of rectangles we will get multidimensional parallelepipeds. Each parallelepiped corresponds to the active state of one $b_j$ neuron (Fig.7).

For example, if we want to recognize new point ($X_1^*$, $X_2^*$) in the space of two features ($X_1$ and $X_2$) and two classes (1 and 2) (Fig. 10), we will obtain the neuron responses which are related to the rectangles that cover new point.

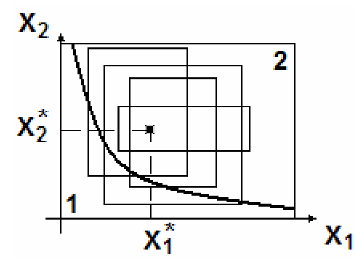


Fig. 10. New point recognition with RTC

During training process connections between active neurons that correspond to point ($X_1^*$, $X_2^*$) and output neuron that correspond to the second class will be stronger than those with output neuron that correspond to the first class because major part of these rectangles is covered by second class area. Therefore, this point will be recognized as the point of the second class.

## 4 Random Subspace Classifier

When the dimension of input space $n$ (Fig.7) increases it is necessary to increase the gap between the thresholds of neurons $h_i^j$ and $l_i^j$, so for large $n$ many thresholds of neurons $h_i^j$ achieve the higher limit of variable $X_i$ and thresholds of $l_i^j$ achieves lower limit of variable $X_i$. In this case the corresponding neuron $a_i^j$ always has output 1 and gives no information about the input data. Only the small part of neurons $a_i^j$ can change the outputs. To save the calculation time we have modified RTC classifier including to each block $j$ only a small number of neurons $a_i^j$, i.e. we calculate the output of $a_i^j$ only for a small number of input features, which we select randomly from the input vector. This small number of chosen components of input vector we term random subspace of the input space. For each block $j$ we select different random subspaces. Thus we represent our input space by a multitude of random subspaces.

## 5 Feature extraction

Our image database consists of 60 gray-scale images with resolution of 220x220 pixels, 20 images for each of three classes. The procedure of image processing is organized as scanning across the initial image by moving a window of 40x40 pixels with step of 20 pixels. This overlapping smoothes out transitions from one texture region to another. It is important to select window size appropriately for all textures within the set to be classified because it gives us opportunity to obtain local characteristics of the texture under recognition.

For every window three histograms of brightness, contrast and contour orientation were calculated. Every histogram contains 16 components, so at all we have 48 components which we use as features. These 48 features form the input vector for our RSC classifier.

## 6 Encoder of features

The task of encoder is to codify the feature vector ($X_1$, ..., $X_n$) into binary form in order to present it to the input of one-layer classifier.

To create the encoder structure we have to select the subspaces for each neuron block. For example, if subspace size is 3, in each neuron block $j$ we will use only three input parameters whose numbers we select randomly from the range 1,…, $n$ (where $n$ is dimension of the input space, in our case $n$=48). After that, we calculate the thresholds for each pair of neurons $l_i^j$ and $h_i^j$ of three selected neurons $a_i^j$ of the block $j$. For this purpose we select the point $x_i^j$ randomly from the range of the 0,…, $X_i$. After that we select random number $y_i^j$ uniformly distributed in the range 0,…, GAP, where GAP is the parameter of the encoder structure. Then we calculate the thresholds of neurons $l_i^j$ and $h_i^j$ in accordance with formulas:

$$\begin{aligned} &Trl_i^j = x_i^j - y_i^j; \\ &if\ \left(Trl_i^j < X_i \min\right) then\ Trl_i^j = X_i \min; \end{aligned} \qquad (1)$$

$$\begin{aligned} &Trh_i^j = x_i^j + y_i^j; \\ &if\ \left(Trh_i^j > X_i \max\right) then\ Trh_i^j = X_i \max; \end{aligned} \qquad (2)$$

where $Trl_i^j$ and $Trh_i^j$ are the thresholds of neurons $l_i^j$ and $h_i^j$ correspondingly, $X_i min$ and $X_i max$ are the minimum and maximum possible values for a component $X_i$ of the input vector ($X_1$, ..., $X_n$).

Then encoder forms binary vector ($b_1$,…, $b_s$) for each feature vector. This vector is presented to the input of one-layer classifier. The training rule of our one-layer classifier is the same as one of a one-layer perceptron.

## 7 Results of texture recognition

We have made experiments with different number of images in training/test sets. The larger number of images we use for training the better results we obtain in recognition and, for example, selecting for each output class only 3 images for training set and 17 images for recognition set we obtained as a result 80% of correct recognition. The parameters of the RSC neural classifier were the following: number of neurons – 30000, number of training cycles – 500.

## 8 Discussion

There are quite a few methods that work well when the features used for the recognition and classification are obtained from a database sample that has the same orientation and position as the test sample; but as soon as the orientation and/or position of the test image is changed with respect to the one in the database the same methods will perform poorly. The usefulness of methods that are not robust to the changes in the orientation is very limited and that is the reason of developing of our texture classification system that works well independently of the particular position and orientation of the texture. In this sense the results obtained in experiments are sufficiently promising.

We train our RSC neural classifier with patterns in all the expected orientations and positions in such way that the neural network becomes insensitive to those specific orientations and changes in positions.

## 9 Conclusion

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

The neural network classifier is developed for recognition of the textures of mechanically treated surfaces. This classifier can be used in recognition systems that have to recognize position and orientation of complex work pieces in the task of assembly of micromechanical devices. The performance of the developed classifier was tested in recognition of three texture types obtained after milling, turning and polishing of metal surfaces. The obtained recognition rate is 80%. In the future we want to improve this result.

Acknowledgement

This work was supported in part by the projects PAPIIT 112102, 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*, Vol.1, 1998, pp. 421-427.

[3] E. Kussul, T. Baidyk, L. Ruiz-Huerta, A. Caballero, G. Velasco, L. Kasatkina, Development of micromachine tool prototypes for microfactories, *J. of Micromech. and Microeng.*, Vol.12, 2002, pp. 795-813.

[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 Recognition Letters*, Vol.25/1, 2004, pp. 107-118.

[5] C.R. Friedrich, M.J. Vasile, Development of the micromilling process for high- aspect- ratio micro structures, *J. Microelectromechanical Systems*, Vol.5, 1996, pp. 33-38.

[6] Naotake Ooyama, Shigeru Kokaji, Makoto Tanaka, et al, Desktop machining microfactory, *Proceedings of the 2-nd International Workshop on Microfactories*, Switzerland, 2000, pp.14-17.

[7] T. Baidyk, E. Kussul, Application of neural classifier for flat image recognition in the process of microdevice assembly, *IEEE Intern. Joint Conf. on Neural Networks*, Hawaii, USA, Vol. 1, 2002, pp. 160-164.

[8] R. Castano, R.Manduchi, J. Fox, Classification experiments on real-world texture, *Proceedings of the Third Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, Hawaii, 2001, pp. 3-20.

[9] Matti Pietikäinen, Tomi Nurmela, Topi Mäenpää, Markus Turtinen, View-based recognition of real-world textures, *Pattern Recognition*, Vol.37, 2004, pp. 313-323.

[10] T. Baidyk, Comparison of texture recognition with potential functions and neural networks, *Neural Networks and Neurocomputers*, 1991, pp.52-61.

[11] E. Kussul, D. Rachkovskij, T. Baidyk, On image texture recognition by associative-projective neurocomputer, *Proceedings of the International Conference on Intelligent Engineering Systems through Artificial Neural Networks*, 1991, pp.453-458.

[12] A. Goltsev, An assembly neural network for texture segmentation, *Neural Networks*, Vol.9, No.4, 1996, pp. 643-653.

[13] A. Goltsev, D. Wunsch, Inhibitory connections in the assembly neural network for texture segmentation, *Neural Networks*, Vol.11, 1998, pp. 951-962.

[14] Chi-ho Chan, Grantham K.H. Pang, Fabric defect detection by Fourier analysis, *IEEE Transactions on Industry Applications*, Vol.36/5, 2000, pp. 1267-1276.

[15] E. Zschech, P. Besser. Microstructure characterization of metal interconnects and barrier layers: status and future. *Proc. of the IEEE International Interconnect Technology Conference*, 2000, pp. 233-235.

[16] L. Hepplewhite, T.J. Stonham, Surface inspection using texture recognition, *Proc. of the 12th IAPR International Conference on Pattern Recognition*, Vol.1, 1994, pp. 589-591.

[17] D.Patel, Hannah I, E.R. Davies, Foreign object detection via texture analysis, *Proc. of the 12th IAPR International Conference on Pattern Recognition*, 1994, Vol.1, pp. 586-588.

[18] A. Hoogs, R. Collins, R. Kaicic, J. Mundy, A common set of perceptual observables for grouping, figure-ground discrimination, and texture classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.25/4, 2003, pp. 458-474.

[19] E. Kussul, T. Baidyk, V. Lukovitch, D. Rachkovskij, Adaptive high performance classifier based on random threshold neurons, *Proc. of Twelfth European Meeting on Cybernetics and Systems Research (EMCSR-94*), Austria, Vienna, 1994, pp. 1687-1695.

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

[21] K.J. Dana, B. van Ginneken, S.K. Nayar, J.J. Koendrink, Reflectance and texture of real world surfaces, *ACM Trans. Graphics*, Vol.18 (1), 1999, pp. 1-34.

[22] http://ww1.cs.columbia.edu/CAVE/curet/

[23] D. Patel, T.J. Stonham, A single layer neural network for texture discrimination, *IEEE International Symposium on Circuits and Systems*, Vol.5, 1991, pp.2656-2660.

[24] M.A. Mayorga, L.C. Ludeman, Shift and rotation invariant texture recognition with neural nets, *Proc .of the IEEE International Conference on Neural Networks*, Vol.6, 1994, pp. 4078-4083.

[25] Woobeom Lee, Wookhyun Kim, Self-organization neural networks for multiple texture image segmentation, Proc. *of the IEEE Region 10 Conference TENCON 99*, Vol.1, 1999, pp. 730-733.

[26] P. Brodatz, *Texture: A Photographic Album for Artists and Designers*, Dover Publications, New York, 1966.

[27] 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*, Vol.2, 1991, pp. 1148-1153.