ARTIFICIAL NEURAL NETWORKS IN PATTERN RECOGNITION

Mohammadreza Yadollahi, Aleš Procházka

Institute of Chemical Technology, Department of Computing and Control Engineering

Abstract

Pre-processing stages such as filtering, segmentation and feature extraction form important stages before the use of artificial neural networks in pattern recognition or feature vectors classification. The paper is devoted to analysis of pre-processing stages before the application of artificial neural networks in pattern recognition by Kohonen's method and to numerical comparison of results of classification. All algorithms proposed are applied for a biomedical image processing in the MATLAB environment.

1 Introduction

Nowadays, artificial neural networks (ANN) are applicable in many scientific fields such as medicine, computer science, neuroscience and engineering. It is believed that they will receive extensive application to biomedical systems and engineering. They can be also applied in information processing system to solve pattern analysis problems or feature vectors classification [1] for example. This paper describes pre-processing stages before the use of artificial neural networks that include (i) filtering tools to reject noise, (ii) image segmentation to divide the image area into separate regions of interest, and (iii) the use of mathematical method for extraction of feature vectors from each image segment. The set of feature vectors form finally the pattern matrix as the input to the artificial neural networks for their classification. This process is followed by the application of ANN for classification of feature vectors by the unsupervised learning process based on the Kohonen's method. All algorithms used were designed in the Matlab environment.

2 Mathematical Background

The mathematical methods used include the discrete Fourier transform for signal analysis, application of difference equations for digital filtering and optimization methods for training of artificial neural networks.

2.1 Discrete Fourier Transform in Signal Analysis

Discrete Fourier Transform (DFT) is a method used almost in all branches of engineering and science [2]. It is a basic operation for transform of the ordered sequence of data samples from the time domain into the frequency domain to distinguish its frequency components [3]. Discrete Fourier transform algorithms include both the direct and the inverse discrete Fourier transform, which converts the result of direct transform back into the time domain. The discrete Fourier transform of the signal $\{x(n)\}$, $n = 0, 1, 2, ..., N-1$, is defined [4] by relation

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\,k\,n\frac{2\,\pi}{N}} \tag{1}$$

where $k = 0, 1, ..., N-1$. The inverse discrete Fourier transform of a sequence $\{X(k)\}$, $k = 0, 1, 2, ..., N-1$ is defined [4] by formula

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\,k\,n\frac{2\,\pi}{N}} \tag{2}$$

where $n = 0, 1, ..., N-1$.

2.2 Digital Filters

Filtering is the most common form of signal processing used in many applications including telecommunications, speech processing, biomedical systems, image processing, etc. to remove the frequency components in certain regions and to improve the magnitude, phase, or group delay in other signal spectrum ranges [5].

Finite Impulse Response (FIR) filters [5] or non-recursive filters do not have feed back and input-output relation is defined by equation

$$y(n) = \sum_{k=0}^{M} b_k \; x(n-k) \tag{3}$$

where $\{y(n)\}$ and $\{x(n)\}$ are output and input sequences respectively and M is the filter order. Fig. 1 displays the given sequence, output of FIR filter and both the frequency response of the FIR filter and frequency spectrum of the given sequence.

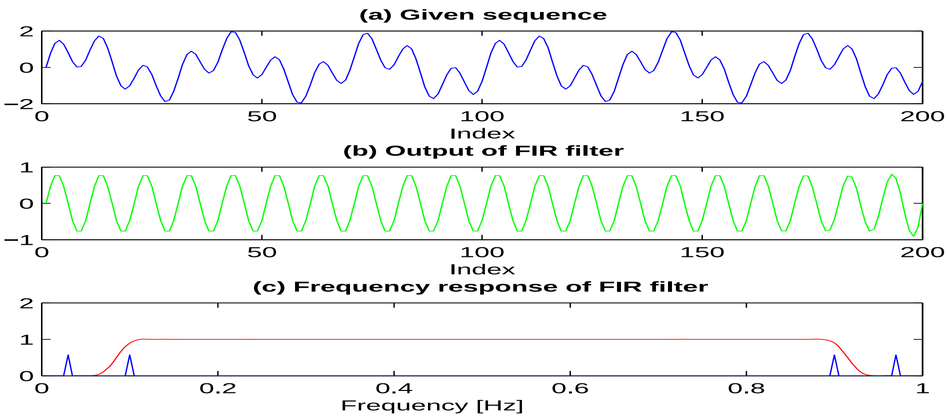

Figure 1: FIR filtering **(a)** the given simulated sequence formed by the sum of two harmonic functions, **(b)** result of its processing by the FIR filter of order $M = 60$, and **(c)** frequency spectrum of the given sequence together with the frequency response of the FIR filter

Infinite Impulse Response (IIR) filters [4] or recursive filters have feedback from the output to input. They can be described by the following general equation

$$y(n) = -\sum_{k=1}^{M} a_k \; y(n-k) + \sum_{k=0}^{M} b_k \; x(n-k) \tag{4}$$

Owing to the lower number of necessary coefficients to obtain similar accuracy the IIR filters are more efficient and faster requiring less storage comparing to FIR filters. On the other hand while FIR filters are always stable the IIR filters can become unstable [4].

2.3 Principle of Neural Networks

Following subsections are devoted to mathematical description of ANN and to the selected problems of unsupervised learning by the Kohonen's method for classification of feature vectors.

2.3.1 Mathematical Description of Neural Networks

An artificial neural network is an adaptive mathematical model or a computational structure that is designed to simulate a system of biological neurons to transfer an information from its input to output in a desired way [6]. In the following description small (*italic*) letters are used for scalars, small (**bold**, nonitalic) letters for vectors and capital (**BOLD**, nonitalic) letters for matrices. Basic neuron structures [7] include

- **Single-Input Neuron** (Fig. 2(a)): The scalar input p is multiplied by the scalar weight w. The bias b is then added to the product $w\,p$ by the summing junction and the transfer function f is then applied for $n{=}wp{+}b$. This transfer function is typically a step function or a sigmoidal function shifted by value of b producing the output a.

- **Multiple-Input Neuron** (Fig. 2(b)): Individual inputs forming a vector $\mathbf{p}{=}[p_1, p_2, ..., p_R]'$ are multiplied by weights $\mathbf{w}{=}[w_{1,1}, w_{1,2}, ..., w_{1,R}]$ and then fed to the summing junction. The bias b is then added to the product $\mathbf{w}\,\mathbf{p}$ forming value

$$n = \mathbf{w}\,\mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + ... + w_{1,R}p_R + b \tag{5}$$

  which can be written in the following matrix form

$$n = \mathbf{w}\,\mathbf{p} + b \tag{6}$$

  This value is then processed by the non-linear transfer function to form the neuron output

$$a = f(\mathbf{w}\,\mathbf{p} + b) \tag{7}$$

- **Layer of Neurons** (Fig. 2(c)): Each element of the input vector $\mathbf{p}$ is connected to each neuron using coefficients of the weight matrix $\mathbf{W}$.

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \dots & \dots & \dots & \dots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

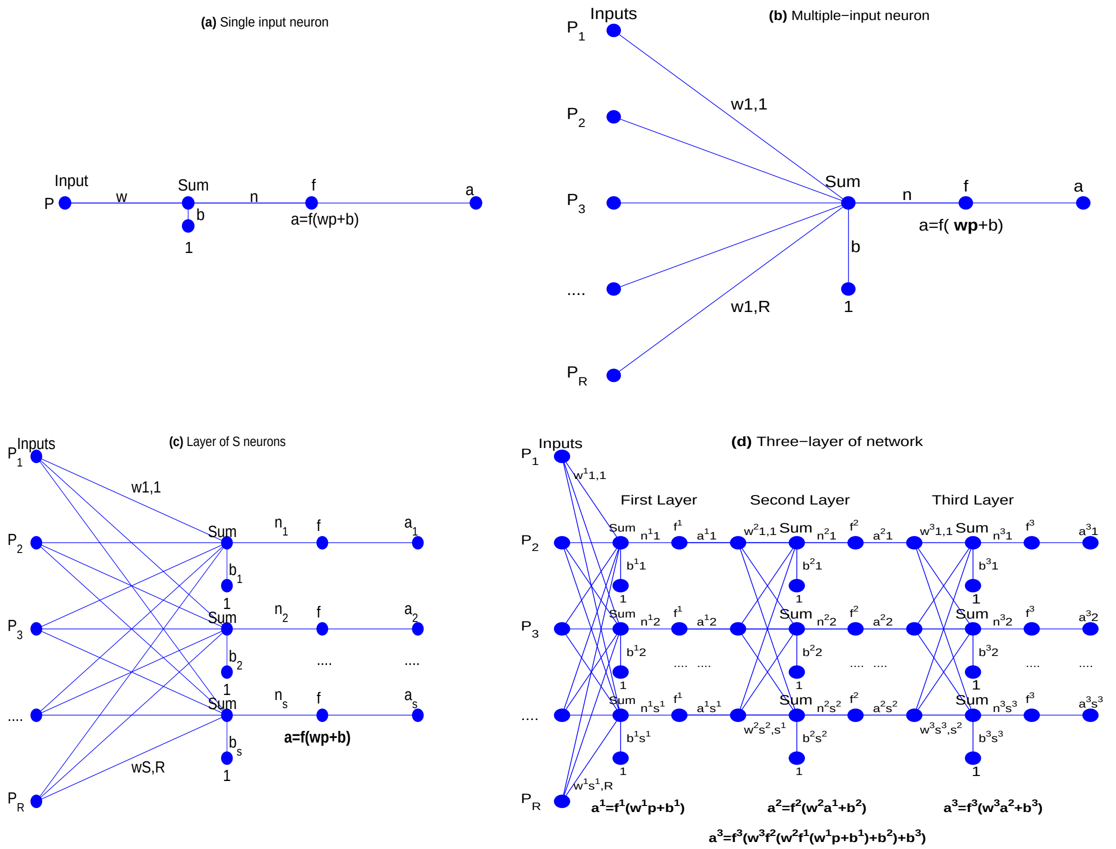

Figure 2: Different structures of neural networks presenting **(a)** a single-input neuron with a bias, **(b)** a multiple-input neuron with biases, **(c)** a layer of neurons, and **(d)** multiple layers of neurons

The *i-th* neuron has a summer that gathers its weighted inputs and bias to form its own scalar output $n(i)$ of an S-element net vector $\mathbf{n}$

$$\mathbf{n} = \mathbf{W}\,\mathbf{p} + b \tag{8}$$

This value is then processed by the non-linear transfer function to form the neuron output

$$\mathbf{a} = f(\mathbf{W}\,\mathbf{p} + b) \tag{9}$$

The row indices on the elements of matrix $\mathbf{W}$ indicate the destination neuron of the weight and the column indices indicate which source is the input for that weight [7].

- **Multiple Layers of Neurons** (Fig. 2(d)): Each layer has its own weight matrix $\mathbf{W}$, its own bias vector $\mathbf{b}$, a net input vector $\mathbf{n}$ and an output vector $\mathbf{a}$. The weight matrix for the first layer can be denoted as $\mathbf{W}^{(1)}$, the weight matrix for the second layer as $\mathbf{W}^{(2)}$, etc. The network has R inputs, $S^{(1)}$ neurons in the first layer, $S^{(2)}$ neurons in the second layer, etc. The outputs of layers one and two are the inputs for layers two and three. Thus layer 2 can be viewed as a one-layer network with $R = S^{(1)}$ inputs, $S = S^{(2)}$ neurons and weight matrix $\mathbf{W}^{(2)}$ of size $S^{(1)}{\times}S^{(2)}$. The layer 2 has input $\mathbf{a}^{(1)}$ and output $\mathbf{a}^{(2)}$. The layers of a multilayer network play different roles. A layer that produces the network output is called an output layer while other layers are called hidden layers. The three layer network has one output layer (layer 3) and two hidden layers (layer 1 and 2) [7].

2.3.2 Unsupervised Learning

Data clustering forms a basic application of unsupervised learning called also the learning without any external teacher and target vectors using one-layer network having R inputs and S outputs. Each input column vector $\mathbf{p}{=}[p_1, p_2, ..., p_R]'$ forming the pattern matrix $\mathbf{P}$ is classified into the suitable cluster using the Kohonen's method and the weight matrix $\mathbf{W}_{S,R}$. The competitive learning algorithm [8] includes the following basic steps

- Initialization of weights to small random values
- Evaluation of distances D_s of the pattern vector $\mathbf{p}$ to the *s-th* node

$$D_s = \sqrt{\sum_{j=1}^{R} (p_j - W_{s,j})^2} \tag{10}$$

  for $s = 1, 2, ..., S$
- Selection of the minimum distance node or in other words the winner node index i node
- Update of the weights of the winner node i

$$w_{i,j}^{new} = w_{i,j}^{old} + \eta(p_j - w_{i,j}^{old}) \tag{11}$$

  where η is the learning rate

3 Image Segmentation

Image segmentation [9] forms an important step to image regions classification. The Watershed transform (WT) can be used to contribute to this task as it can find the watershed ridges in an image. It assigns zero to all ridges detected and it assigns the same unique labels to all points inside separate catchment basins (regions) using the gray-scale image as a topological surface. The Distance transform (DT) forms the fundamental part of the Watershed transform used for the binary image [9, 10] and based upon the calculation of Euclidean distance of each pixel to the nearest pixel with the value 1. Result of this transformation [10] is formed by

$$b_{i,j} = \begin{cases} 0 & \text{for } a_{i,j} = 1 \\ \min\limits_{\forall k,l,a_{k,l}=1} \left(\sqrt{(i-k)^2 + (j-l)^2}\right) & \text{for } a_{i,j} = 0 \end{cases} \tag{12}$$

where $a_{i,j}$ is each pixel and $a_{k,l}$ is nearest pixel for $i = 1, 2, ..., M$ and $j = 1, 2, ..., N$. Application of defined formula by Eq. (12) is showed in following example of matrix

$$I_{bw} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{13}$$

Its distance transform imply the matrix

$$I_{dt} = \begin{pmatrix} 0 & 0 & 1.0000 & 2.0000 & 3.0000 \\ 0 & 0 & 1.0000 & 2.0000 & 2.2361 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.4142 \\ 1.0000 & 0 & 0 & 0 & 1.0000 \\ 1.4142 & 1.0000 & 1.0000 & 1.0000 & 1.4142 \end{pmatrix} \tag{14}$$

Fig. 3 displays a simulated gray-scale image, its conversion to the binary image, results of the application of the distance transform and the final image presenting image edges obtained by the watershed transform.

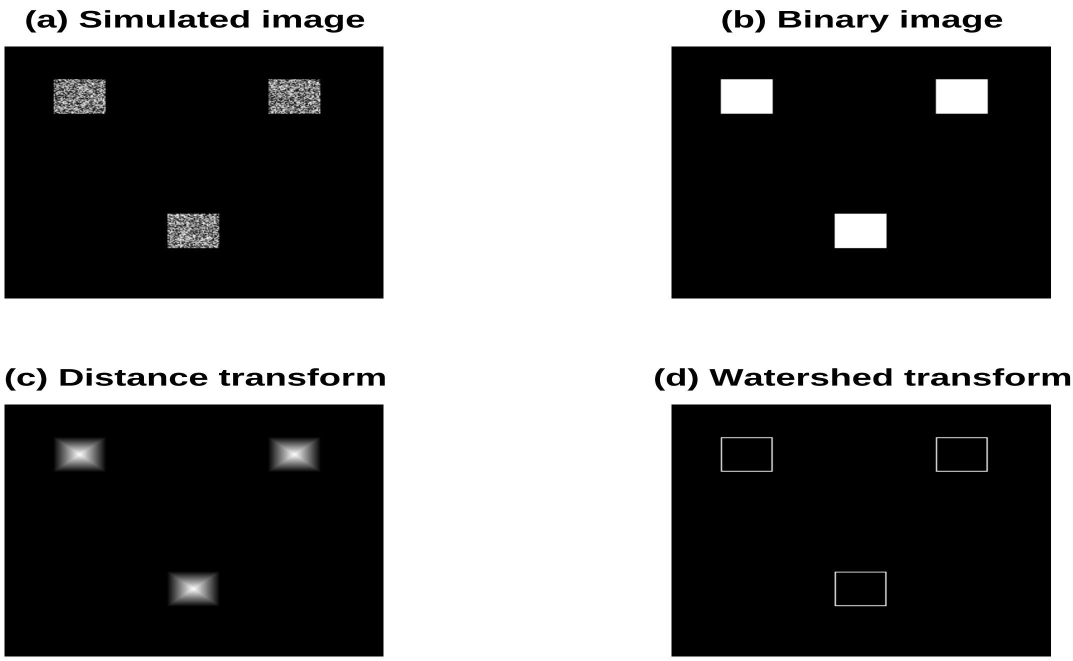

Figure 3: Process of segmentation including **(a)** simulated gray-scale image, **(b)** its conversion to the binary image, **(c)** results of the evaluation of the distance transform of complementary binary image, and **(d)** watershed transform

4 Results

The pre-processing stages of image filtering, segmentation, and feature extraction were applied for real biomedical images of size 128 × 128 to enable the following classification by artificial neural networks discussed. Fig. 4 presents the set of all these processes applied for a biomedical image **(a)** and results of its median filtering **(b)**, segmentation by the Watershed transfom **(c)**, image segments DFT features **(d)** and results of classification using artificial neural networks and Kohonen's learning rule **(e)**.

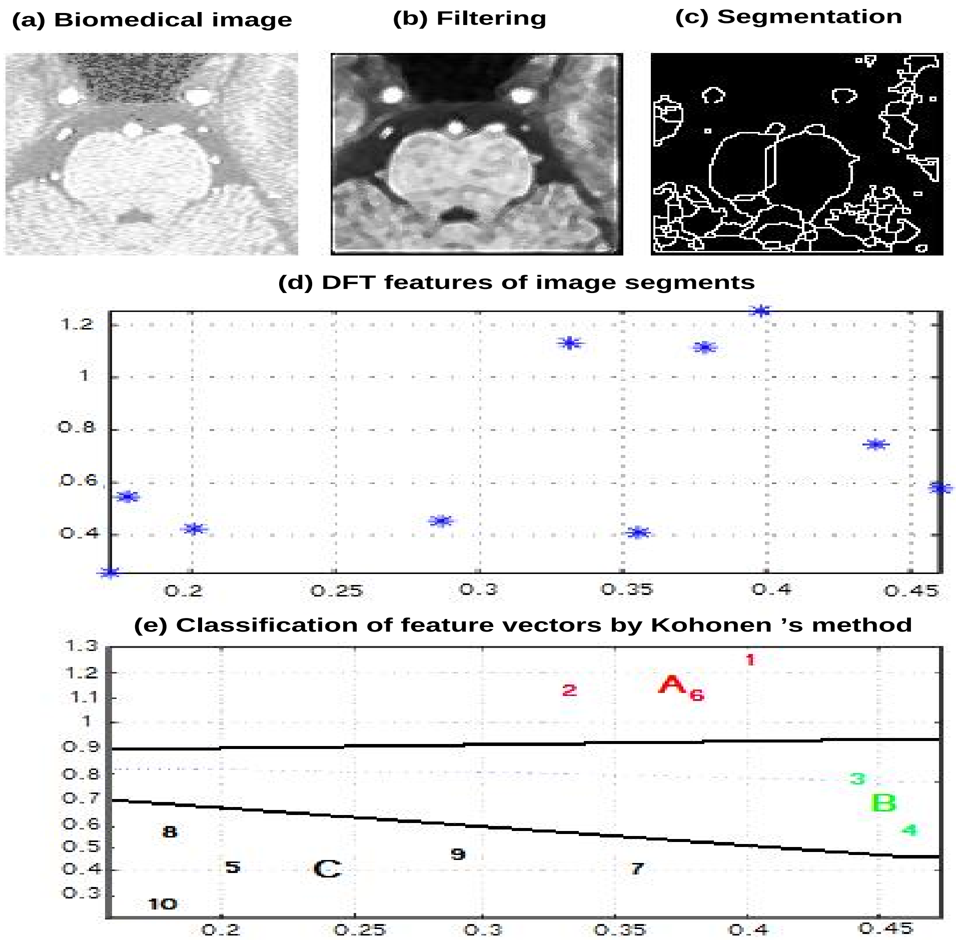

Figure 4: Processing stages of a selected biomedical image presenting **(a)** the given image, **(b)** results of its median filtering, **(c)** evaluation of the Watershed transfom, **(d)** image segments DFT features plot, and **(e)** results of classification using artificial neural networks by the Kohonen's learning rule with class boundaries.

In the present study the median filtering has been used and results compared with other methods. A special attention has been paid to estimation of image regions properties. Both DFT and selected statistical characteristics have been used to form feature vectors of all segments of the image. To allow visualization of results only two features were selected to form feature matrix

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,k} & \cdots & p_{1,Q} \\ p_{2,1} & \cdots & p_{2,k} & \cdots & p_{2,Q} \end{bmatrix}$$

Then the matrix $\mathbf{W}_{S,2}$ was used to classify image features into S classes.

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} \\ \vdots & \vdots \\ w_{k,1} & w_{k,2} \\ \vdots & \vdots \\ w_{S,1} & w_{S,2} \end{bmatrix}$$

The learning process has been based upon the Kohonen's method to classify feature vectors with a selected result on Fig. 4(e). The calculation of a typical structure is done in every cluster by

Table 1: Evaluation of classification.

No.of segment	Class	Mean distance	Typical cluster	Criteria value
1	A	0.0650	6	0.16107
2	A			
6	A			
3	B	0.0850	3	
4	B			
5	C	0.1060	5	
7	C			
8	C			
9	C			
10	C			

computing distance of each pattern to weights of the corresponding neuron using relation

$$d(i) = \sqrt{(p_{1,j} - w_{i,1})^2 + (p_{2,j} - w_{i,2})^2} \tag{15}$$

The feature vector with the least distance in the cluster points to the typical pattern. Evaluating the mean distance for every cluster by relation

$$MSE = \frac{1}{N} \sum_{i=1}^{N} d(i) \tag{16}$$

where N stands for the number of pattern values in the selected class it is possible to detect the cluster compactness.

The quality of the separation process is then evaluated using the mean distances of patterns in all classes and distances of their gravity centers.

5 Conclusion

Further studies will be devoted to the use of different mathematical methods for obtaining compact cluster allowing the efficient classification.

6 Acknowledgements

The work has been supported by the research grant of the Faculty of Chemical Engineering of the Institute of Chemical Technology, Prague No. MSM 6046137306

References

[1] R.Nayak, L.C.Jain and B.K.H. Ting. *Artificial Neural Networks in Biomedical Engineering: A Review.* In Proceedings Asia-Pacific Conference on Advance Computation.2001.QUT Digital Repository :http://eprints.qut.edu.au/1480/.

[2] Shripad Kulkarni. *Discrete Fourier Transform(DFT) by using Vedic Mathematics.* October 2008.http://www.vedicmathsindia.org/Vedic%20Math %20IT/5.DFT%20Using%20 Vedic%20Math%20Kulkarni.pdf.

[3] Paulo Roberto, Nunes de Souza, Mára Regina and Labuto Fragoso da Silva. *Fourier Transform Graphical Analysis: an Approach for Digital Image Processing.* Universidade Federal do Espírito Santo.2009. http://w3.impa.br/ lvelho/sibgrapi2005/html/p13048 /p13048.pdf.

[4] Saeed V. Vaseghi. *Multimedia Signal Processing: Theory and Applications in Speech, Music and Communications* . John Wiley and Sons, 2007. ISBN: 9780470062012.

[5] Belle A. Shenoi. *Introduction to Digital Signal Processing and Filter Design.* John Wiley and Sons, 2005. ISBN: 9780471464822.

[6] Eeglossary. *Neural Networks: Definition and Recommended Links* . 1 May 2009.http://www.eeglossary.com/neural-networks.htm.

[7] Martin T. Hagan, Howard B . Demuth and Mark H. Beale. *Neural Network Design.* PWS Pub., 1996. ISBN: 9780534943325.

[8] Unsupervised learning. *Lecture 8.Artificial Neural Networks.* Neg nevitsky, Pearson Education, 2002.www.neiu.edu/ mosztain/cs335/lecture08.ppt.

[9] R. C. Gonzales, R. E. Woods and S. L. Eddins. *Digital Image Processing using MATLAB.* Prentice Hall, 2004. ISBN-13: 978-0130085191.

[10] Andrea Gavlasová, Aleš Procházka and Jaroslav Pož ivil. *Functional Transforms in MR Image Segmentation.* The 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP), Malta, 12-14 March 2008.http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4537437.

Ing. Mohammadreza Yadollahi, Prof. Aleš Procházka
Institute of Chemical Technology, Praha
Department of Computing and Control Engineering
Technická 1905, 166 28 Praha 6
Phone: +420 220 444 170, +420 220 444 198 * Fax: +420 220 445 053
E-mail:Mohammadreza1.Yadollahi@vscht.cz, A.Prochazka@ieee.org