Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp102-108)

The objects location from images binarized by means of self-learning neural network

VINCENZO NIOLA
Department of Mechanical Engineering for Energetics
University of Naples "Federico II"
Via Claudio, 21 - 80125 Napoli
ITALY

GIUSEPPE QUAREMBA
University of Naples "Federico II"
Via S. Pansini, 5 – 80131 Napoli
ITALY

SERGIO SAVINO
University of Naples "Federico II"
Via Claudio, 21 - 80125 Napoli
ITALY

Abstract: - The present work represents a first approach for the realization of a methodology for objects location and recognition. The choice objects was the starting point of the study; it has been decided to consider nuts and bolts of various dimensions. Then in order to eliminate the noise added to the images a neural network was designed. Once the desired pictures were obtained, an algorithm, based on the theory of the fuzzy sets, was applied, for detecting the position of each objects. The instruments used were low cost: a digital camera, one stand, equipped with four lamps assuring, for each acquisition, a diffused lighting system, one personal computer.

Key-words:- Computer vision, Fuzzy systems, Neural network, Object location.

1 Introduction

The importance of the computer vision is well known for the industrial applications and automation. The development of such a science has produced sophisticated technologies, as well as vision systems concurring to the control of many activities without the employment of dedicated workers. The process consists in the acquisition of images and their interpretation. Several applications of such a science are in the field the advanced technology (e.g., assembling and disassembling of mechanical components, cutting or taking off of metal chips, electrophysical and electrochemical processes, cold or hot moulding processes, CAD, metrology). Other applications are on the process (e.g., control of raw materials, machine tools control, packing or parceling, stocking, quality control, maintenance).
The advantages are multiple and are as follows:

- elimination of the human error, in particular when repetitive or monotone operations are involved, which should cause inattention or stress to the workers;
- production increasing by improving automatic systems controls;
- powerful information control system, through the data acquisition of historical trends for successive elaborations;
- quality control based on objective parameters.

The present work, taking place within that scenery, proposes a new methodology for objects location. The proposed method, belongs to a computer vision problem of *low level*, as it will be demonstrated later. It provides the position of objects randomly placed starting from a noised picture (comparable to a real working condition). To gat this goal a Self Learning Neural Network (SLNN) was designed with the scope of operating a soft and intelligent denoising of the scene without loss of fundamental information [1]. For the identification and the positional calculation of objects, it has been used an algorithm based on the theory of fuzzy systems [2]: an appropriate algorithm examines the binarized image provided by neural network and separates the points, composing the image, into clusters and estimates, for each cluster, the relative centre of gravity. This means that, as final result, all the objects will be identify in their position. The coordinates of the centre of gravity, then, will be passed to a robot in order to pick-up the single object.

2 Multilayer perceptron

The *perceptron* can be thought like a net composed of elementary processors organized in such a way to recreate the biological neural connections [3]. It is able to learn, to recognize and to classify *pattern* in independent way. A model of Multilayer Perceptron (MLP) should be applied when the pattern is not linearly separable. Observe that the nodes of two consecutive levels are connected by one link (or weight) but no connection exists among nodes belonging to the same level. The level where the nodes of input are present is named *input layer*, while the level which shows the output is said *output layer*. The layers which lies between the input and the output layers are named *hidden layers*[4].

The output of nodes of one layer is transmitted to the correspondent nodes of the following layers by means of links (weights) which can amplify, attenuate or inhibit such output through *weighted factors*. With the exception of nodes of the input layer, the total input for each node is the sum of the weighted output of nodes belonging to previous layer. Each node is activated in agreement with the input received from both the other nodes and the activation function. The total input of the i-th node of one layer is:

$$I_i = \sum_j w_{ij} o_j \quad (1)$$

where o_j is the output of j-th neuron of the previous layer and w_{ij} is the weighted link between the i-th node of one layer and j-th node of the previous layer.

The output of the i-th node is:

$$o_i = f(I_i)$$

where $f(\cdot)$ is the activation function. Generally the activation function is sigmoidal as shown in Fig.2.

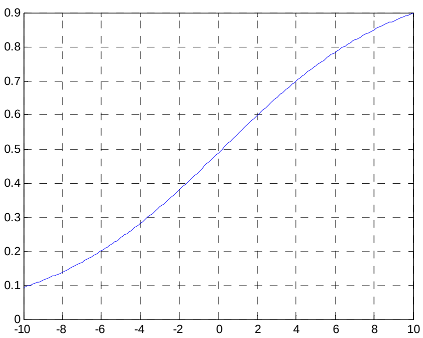

Fig.2 Sigmoidal activation function

The function is symmetrical around θ and θ_0 controls the degree of steepness of the activation function (i.e., value of threshold / bias). During the training set, the pattern X={xi } was submitted as input to the net, where x_i is the i-th component of vector X. In general, the output $\{o_i\}$ is not the same if compared with the target $\{t_i\}$. For a specific pattern p the error can be estimated as:

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

The procedure in order *to learn* the correct set of weights is to vary them in such a way that E is minimized as quickly as possible. From a mathematical point of view this means that gradient of E must be negative [5]:

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} = \eta \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial I_i} \times \frac{\partial I_i}{\partial w_{ji}} = \eta\, \delta_j\, o_i$$

with:

$$\delta_j = -\frac{\partial E}{\partial I_j} = -\frac{\partial E}{\partial o_j} \times \frac{\partial o_j}{\partial I_j} = -\frac{\partial E}{\partial o_j} f'(I_j)$$

Since E can be calculated directly on the output layer, the variation of weights for the links connected to the output layer is:

$$\Delta w_{ji} = \eta \left(-\frac{\partial E}{\partial o_j} \right) \times f'(I_j)\, o_i \,.$$

In particular, if:

$$o_j = \frac{1}{1 + e^{-\left(\left(\sum_i w_{ji}(o_i - o_j)\right)\right)}}$$

then:

$$f'(I_j) = \frac{\partial o_j}{\partial I_j} = o_j(1 - o_j)$$

Finally, we have:

$$\Delta w_{ji} = \begin{cases} \eta\left(-\dfrac{\partial E}{\partial o_j}\right) o_j(1 - o_j) o_i \\ \eta\left(\sum_k \delta_k w_{kj}\right) o_j(1 - o_j) o_i \end{cases}$$

for the output layer and other layers, respectively.
The greater value of η, the quicker is the learning but it should procure strong oscillations in the response. For that reason the last relation can modified as follows [6]:

$$\Delta w_{ji} = (t+1) = \eta \cdot \delta_j \cdot o_i + \alpha \Delta w_{ji}(t)$$

where the term $(t+1)$ is used in order to indicate the time $(t+1)th$ and α is a constant of proportionality.
The neighborhood system N_{ij}^d, for a MxN grid and a generic element, is defined as:

$$N_{ij}^d = \{(i,j) \in L\}$$

such that:

$$(i,j) \notin N_{ij}^d$$

$$(k,l) \in N_{ij}^d \Leftrightarrow (i,j) \in N_{kl}^d$$

For $d = 2$, then N_{ij}^2 can be obtained by considering the 8 pixels like in Fig.3.

			6			
	5	4	3	4	5	
	4	2	1	2	4	
6	3	1	(i,j)	1	3	6
	4	2	1	2	4	
	5	4	3	4	5	
			6			

Fig. 3 Neighborhood system

Note that the pixels of border, since the image has a fixed dimension, will be necessarily less if a periodic grid has not been adopted.

3 Architecture

The SLNN was developed by constructing 3 layers. Each layer has MxN neurons (the image is MxN). Each neuron corresponds to one pixel. Between the input and output layer there exists one layer indeed. The neurons belonging to the same level do not have any link between them. Each neuron of one layer is connected to the correspondent neuron of the previous layer and to its nearest neurons. Moreover, each neuron of the output layer, is connected to the correspondent neuron of the input layer.
The input to a neuron belonging to the input layer is given from a real number in the range $[0,1]$ proportional to the gray level of the correspondent pixel of the image in study. Since we are interested to eliminate the noise and to extract compact regions, all the weights, initially, were put to be equal to 1. The value assigned to θ, used in the activation function, was $\theta = n_l / 2$, where n_l is the number of neighbor neuron.
The input value I_i for each neuron belonging to the i-th layer (except the input layer) was calculated through (1).
The goal is to obtain as output, the greatest number of neurons set to 0 or 1 (optimal extraction of compact regions). Due to the presence of noise the state of such neurons will be, instead, ranged between $[0,1]$. Therefore we will say that the state of the output level can be thought like a *fuzzy set*. The measure of fuzziness of such a set can be considered like the error of instability of the entire system (i.e., neural network). Therefore we can use the fuzziness value as a measure of the error produced by the system and use the back-propagation in order *to adjust* the weights until to eliminate the error (fuzziness). The measure of E can be adopted as a meaningful function of fuzziness index:

$$E = g(I),$$

where I is the measure of fuzziness of a fuzzy set. After a first adjustment of the weights, the output of the neurons belonging to the output layer is used as feedback to the correspondent neurons belonging to the input layer. In the same way the second iteration will proceed. The iteration of weight adjustment shall continue until the net becomes stabilized (i.e., the fuzziness error/index becomes minimum/negligible). When the net shall be stabilized, the state of output of neurons belonging to the output layer shall assume the values 0 or 1. The nodes showing output 0 form one group, the other ones show the value 1, [7].

The mathematical rules for the weight adjustment are the following (weight correction by fuzziness linear index) [8]:

$$\Delta w_{ji} = \begin{cases} -\eta f'(I_j) o_i & \text{se } 0 \le o_j \le 0.5 \\ \eta f'(I_j) o_i & \text{se } 0.5 \le o_j \le 1 \end{cases}$$

Finally, we define the target output t_j as:

$$t_j = \begin{cases} 0 & \text{if } 0 \le o_j \le 0.5 \\ 1 & \text{if } 0.5 \le o_j \le 1 \end{cases},$$

note that if $o_j = 0.5$ the value t_j is not defined. In such a case the algorithm assumes the movement can be made in any direction.

4 Fuzzy C-means

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by [9] and improved by [10]) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2, \quad 1 \le m < \infty$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}}, \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}.$$

This iteration will stop when

$$\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon,$$

where ε is a termination criterion between 0 and 1, whereas k are the iteration steps. The procedure converges to a local minimum or a saddle point of J_m. The data set was clusterized by means a built-in fuzzy c-means clustering routine of Matlab 6.5. It finds a prefixed number of clusters in the assigned data set. The data-set size was M-by-N, where M is the number of data points and N is the number of coordinates for each data point. The coordinates for each cluster center are returned in the rows of the matrix. A membership function matrix U contains the grade of membership of each data point in each cluster. The values 0 and 1 indicate no membership and full membership respectively. Grades between 0 and 1 indicate that the data point has partial membership in a cluster. At each iteration, an objective function is minimized to find the best location for the clusters and its values are returned in an objective function. The clustering process stops when the maximum number of iterations is reached, or when the objective function improvement between two consecutive iterations is less than the minimum amount of improvement specified. Note that the situation of "singularity" (one of the data points is exactly the same as one of the cluster centers) is not checked. However, it hardly occurs in practice [11].

5 The technical equipment

The work was developed at the Mechanical Department for Energetics, University of Naples Federico II, where the technical equipment was assembled for the acquisition of the images, focus of proposed methodology. In order to eliminate the interference due to the external light a well isolated darkroom was prepared. Therefore the lighting source was produced by 4 lamps (4 x 150 W) assembled to the stand by adjustable supports in order to regulate the angle of incidence on the surface of scene (i.e., 40 °). A Nikon Coolpix 5400 was used as camera. The acquisition of images was carried by setting the shutter at $1.60s$ and the optical sensor at $200\,ISO$. All the images have a resolution of $640x480$ pixels.
The camera was placed on the stand in order to form a virtual plan perfectly parallel to the surface of the scene. The distance between the camera and the scene was set to the value of $44.3\,cm$. An example of objects employed is shown in Fig.4

Fig.4 An example of characteristics of nuts and bolts

Note that nuts and bolts are of various dimensions and are galvanized or burnished. In fact, the aim was to test the *robustness* of neural network in order to binarize the images, composed of several objects different for dimension, shape and texture. Other fundamental choice was represented by the background. After several tests with various colors and materials, a black roughness card (i.e., Bristol) was chosen. This solution allows to reduce the problem due to the reflecting contour of pieces with the background. The only condition to be respected was the absence of connection among pieces placed in the scene. For that reason it was possible to place in the scene no more of 6 pieces, the limit due to the dimensional average of objects. All the images were saved in format *jpg*.

6 Image Processing

The algorithms of image processing were implemented for Matlab, version 6.5 [12]. In this paragraph the programs shall be described in more details with reference to results. The starting point is the original image of pieces in jpg format. An example is shown in the Fig.5.

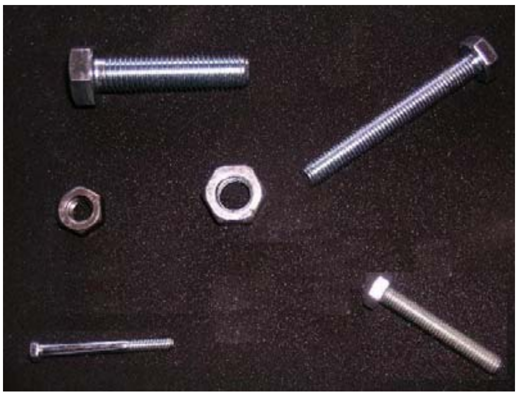

Fig.5 Original RBG image

First of all the image was converted from RBG to gray levels and resized from 640 x 480 to 320 x 240 pixels. The last passage in order to save time computing. The image, added with random noise, constitutes the input matrix for the neural network (Fig.6)

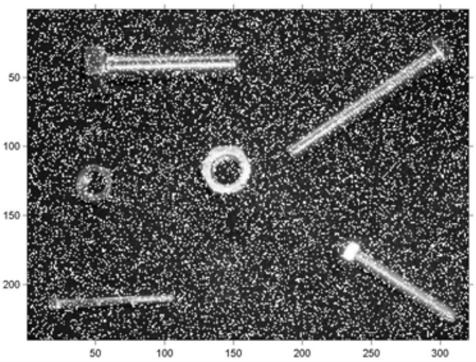

Fig. 6 Noised input image

The results of neural computing is shown in the Figures reported below (Fig.7-12).

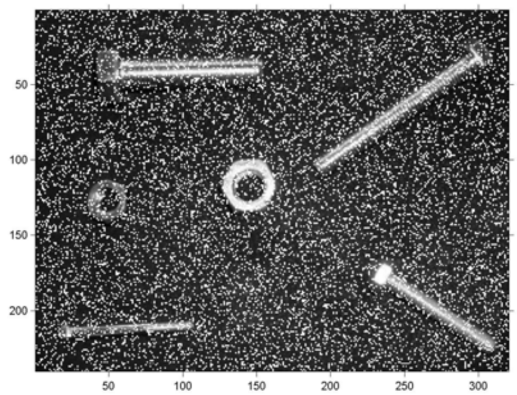

Fig.7 Input 1st layer

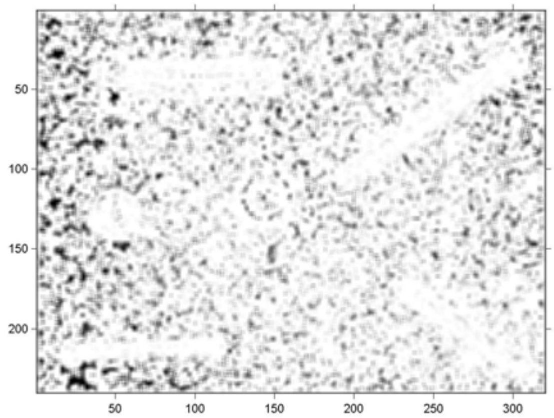

Fig. 8 Output 1st layer

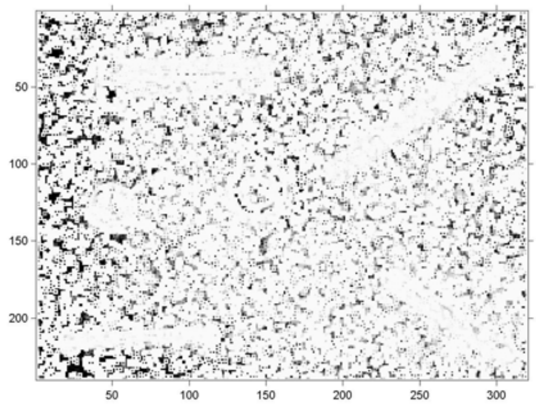

Fig. 9 Input 2nd layer

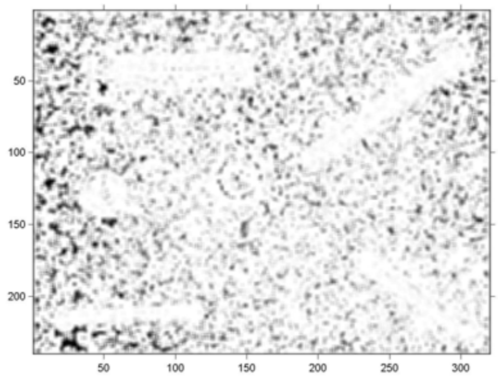

Fig. 10 output 2nd layer

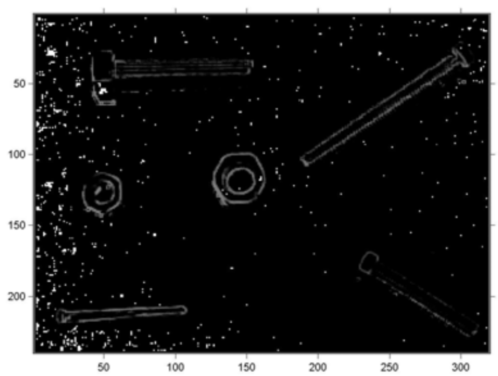

Fig. 11 Input 3rd layer

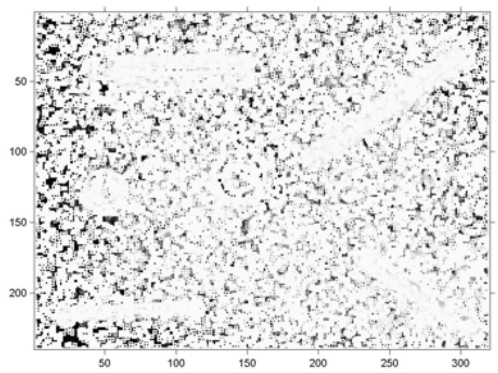

Fig. 12 Output 3rd layer

Note that a significative result is already evident at the end of the first iteration: the net has filtered good part of the noise providing an acceptable image. The procedure was iterated 3 times and produced the image as depicted in the Fig.13. In the 90% of the cases the net converges successfully at the end of the 3rd iteration with a minimal error.

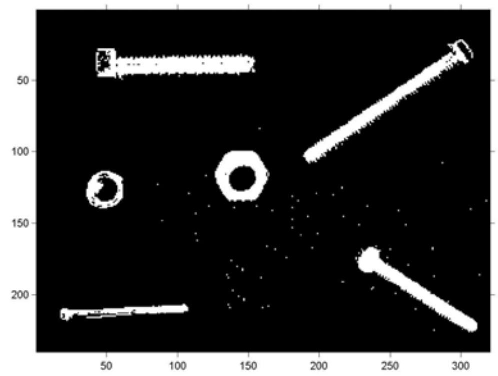

Fig. 13 Output 3rd layer after 3rd iteration

Therefore the first result was: the neural network provides denoised images, with the extraction of content of interest (i.e., denoised objects).

The second goal of the work was represented by the fuzzy cluster analysis computed on the images denoised by the neural network. A built-in function of Matlab was employed. It receives as input the matrix of coordinates of each pixel constituting the binarized image. The number of clusters is equal to the number of objects to be individuated. Each of them is strictly connected. Moreover, for each cluster is individuated the *centroid* indicating, approximately, the barycentre. Finally, the program depict each of cluster with a color in order to individuate the pixels which participate to form the cluster. The centroids are localized with a cross (+) or a circle (O) see Fig. 14

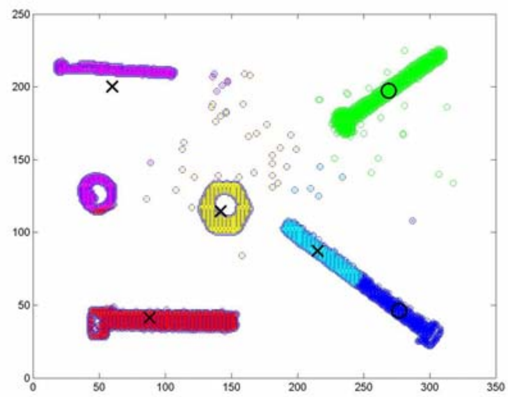

Fig. 14 Result of cluster analysis

Note that the disposition of objects was turned (due to different position of origin of axis) and that not all the objects have been localized correctly. In fact the fuzzy algorithm confused the points belonging to two different objects as the same cluster. However the result is quite good: on six objects four are perfectly indicated in terms of centroids coordinates. The image (Fig.15) is processed by subtracting the objects correctly individuated and so on.

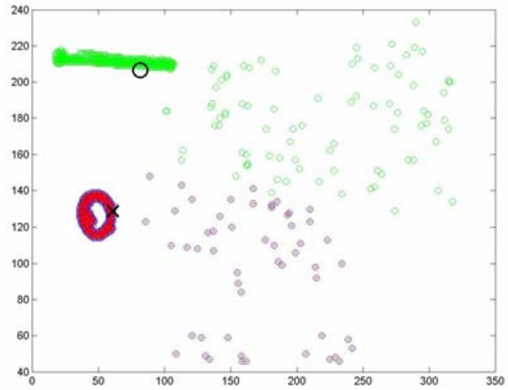

Fig.15 Positional identification of last two objects

7 Conclusions

The paper represents a first study for a new approach to the location of objects randomly placed in a scene with the presence of significative noise. The future development of the work here proposed is multiple. First of all it could be analyzed the learning rate compared with computing time and robustness of results. An other important problem is the selection of new error measures and activation function in order to improve the response. The continuation should be the identification of each object by applying the wavelet transform.

References

[1] R.P. Lippmann, *An Introduction to computing with neural nets*. IEEE ASSP Magazine, pp. 3-22, 1987.

[2] L.A. Zadeh. Fuzzy Sets. *Inform. And Control, Vol. 8, pp. 338-353, 1965.*

[3] L.O. Chua, L. Yang. Cellular neural networks: Applications. *IEEE Trans. Circuits Syst.*, Vo. 35, pp. 1273-1290, 1988.

[4] T. Kohonen. *Self-organized formation of topologically feature maps.* Biol. Cybern., Vol. 43, pp. 59-69, 1982.

[5] S. Amari, Mathematical theory of self-organization in neural nets. In Organization of Neural Networks: Structures and Models. W.V. Seelen, G. Shaw and U.M. Leinhos. Eds.\quad New York: Academic Press, 1988.

[6] A.Ghosh, N.R. Pal, S. Pal. Self-Organization for Object Ectraction Using a Multylayer Neural Network and Fuzziness Meausres. *IEEE Trans. On Fuzzy Systems*, Vol. I, No. 1, Feb. 1993., pp.54-68.

[7] N. Babaguchi, K. Yamada, K. Kise and Y. Tezuku. Connectionist model binarization. In Proc. 10th *Int. Conf. Pattern Recon., 1990*, pp. 51-56.

[8] S.K. Pal, D.D.Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New York: John Wiley (Halsted Press), 1986.

[9] J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57

[10] J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", *Plenum Press*, New York

[11] J.C. Bezdek and S. Pal (eds.) Fuzzy Models for Pattern Recognition. *IEEE Press*, 1991.

[12] Image Processing Toolbox. User's guide. The Mathworks Inc., Version 2, 1998.