

Automatic Tuning of the Pulse-Coupled Neural Network Using Differential Evolution for Image Segmentation

Juanita Hernández and Wilfrido Gómez^(✉)

Information Technology Laboratory,
Center for Research and Advanced Studies of the National Polytechnic Institute,
Ciudad Victoria, Tamaulipas, Mexico
{jhernandez,wgomez}@tamps.cinvestav.mx

Abstract. The pulse-coupled neural network (PCNN) is based on the cortical model proposed by Eckhorn and is widely used in tasks such as image segmentation. The PCNN performance is particularly limited by adjusting its input parameters, where computational intelligence techniques have been used to solve the problem of PCNN tuning. However, most of these techniques use the entropy measure as a cost function, regardless of the relationship of inter-/intra-group dispersion of the pixels related to the objects of interest and their background. Therefore, in this paper, we propose using the differential evolution algorithm along with a cluster validity index as a cost function to quantify the segmentation quality in order to guide the search to the best PCNN parameters to get a proper segmentation of the input image.

Keywords: Automatic image segmentation · Pulse-coupled neural network · Differential evolution · Cluster validity index

1 Introduction

The pulse-coupled neural network (PCNN) is a bio-inspired model based on the cortical model proposed by Eckhorn in 1989. It is used in different applications of image segmentation, although its performance strongly depends on the adequate tuning of its input parameters like decay constants, radio link, number of iterations, etc. [1].

The problem of tuning the PCNN parameters for image segmentation has been addressed by computational intelligence (CI) techniques because they are able to solve optimization problems in complex and changing environments in reasonable computation time [2]. Some relevant works that have optimized the segmentation performance of the PCNN include algorithms based on particle swarm optimization (PSO) [3], genetic algorithm (GA) [4], and differential evolution (DE) [5,6]. These CI-based methods require a proper choice of the cost function to quantify the quality of a potential solution to determine its survival

in the population for ensuring an adequate and consistent segmentation of the input image.

The maximum entropy criterion has been widely used by CI-based techniques for tuning the PCNN parameters to quantify the segmentation quality generated by a potential solution. The maximum entropy criterion only provides a measure of overlap between the intensity probability distributions related to the objects and their back ground, but it does not consider the inter-/intra-group dispersion, that is, how similar are the intensity levels of the objects and how dissimilar are relative to the background intensities. This inter-/intra-group dispersion could be measured by a cluster validity index (CVI).

A CVI is an internal validation index usually used by clustering algorithms to evaluate the quality of a candidate grouping. Currently, using a CVI as a cost function by segmentation algorithms based on CI techniques has not been explored, although it is feasible to be applied because the process of image segmentation is basically a grouping process of pixels with similar intensities. In this context, a CVI quantifies the segmentation quality considering the inter-/intra-group ratio, that is, minimizing the intra-group dispersion while maximizing the inter-group dispersion simultaneously.

Hence, the purpose of this study is to demonstrate that the automatic tuning of the PCNN parameters by means the DE algorithm guided by a CVI improves the performance of image segmentation than solely maximizing the entropy criterion.

The organization of this paper is divided into five sections. Section 2 describes the problem of image segmentation as an optimization problem. Section 3 presents the materials and methods used in this study as well as the description of the proposed approach. Section 4 summarizes the experimental results. Finally, Sect. 5 gives the conclusion and future work.

2 Problem Statement

The problem of image segmentation can be considered as a clustering problem, where an input image \mathbf{R} is partitioned into two groups, c_1 and c_2 , containing the pixels that belong to the objects of interest and their background, respectively, to form a grouping denoted by $\mathbf{C} = \{c_1, c_2\}$ that should satisfy the following three conditions:

1. $c_i \neq \emptyset$ for $i = 1, 2$;
2. $c_1 \cup c_2 = \mathbf{R}$;
3. $c_1 \cap c_2 = \emptyset$.

On the other hand, let $\mathbf{x} = [x_1, \dots, x_d]$ be the vector containing the d input parameters of a PCNN that generates a grouping (or segmentation) \mathbf{C} given \mathbf{R} . Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be the set of vectors of PCNN parameters that generates N feasible groupings of \mathbf{R} . Then, the problem of finding the best clustering

can be formulated as an optimization problem, where $\Omega = \{\mathbf{C}^{\mathbf{x}_1}, \dots, \mathbf{C}^{\mathbf{x}_N}\}$ is the set of candidate groupings of the pixels in \mathbf{R} given the set \mathbf{X} , so the optimal grouping $\mathbf{C}^* \in \Omega$ should satisfy

$$\forall \mathbf{C} \in \Omega : f(\mathbf{C}^*) < f(\mathbf{C}) \quad (1)$$

where $f(\cdot)$ is a cost function given in terms of a CVI, which measures the dispersion intra-/inter-group between c_1 and c_2 . Note that $f(\cdot)$ is minimized without loss of generality.

3 Materials and Methods

3.1 PCNN Model

The PCNN is a bidimensional single layer, laterally connected network of integrate-and-fire neurons, with a 1-to-1 correspondence between the image pixels and network neurons as illustrated in Fig. 1.

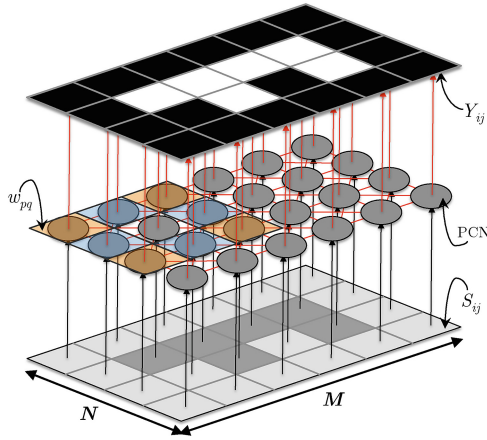


Fig. 1. PCNN configuration, where M and N are the width and height of the input image, S_{ij} is the intensity level, w_{pq} is a synaptic value, and Y_{ij} is the output of a single neuron.

A single pulse-coupled neuron (PCN) has two input channels named feeding and linking, whose responses are combined to regulate the internal neuron activity, which is further compared with a trigger threshold to generate a pulse. Hence, a PCN consists of three main parts: input field, modulation field, and pulse generator, as shown in Fig. 2 [7].

The input field can be seen as an integrator of leaks simulating the dendritic part of the biological neuron, in which each neuron ($N_{i,j}$) receives signals from external sources, in the form of stimuli ($S_{i,j}$) that represents the pixel intensity

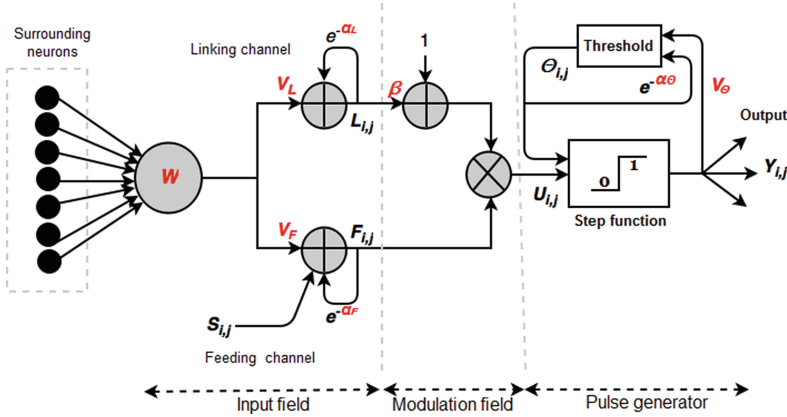


Fig. 2. Typical model of a single PCN. In red, the input parameters.

in the input image, and internal sources, which are the responses of neighboring neurons within a specified radius linked by synaptic weights ($W_{k,l}$). At iteration t , these input signals reach the neuron via the feeding ($F_{i,j}$) and linking ($L_{i,j}$) channels expressed by

$$F_{i,j}[t] = e^{-\alpha_F} F_{i,j}[t - 1] + V_F \sum_{(k,l) \in N(i,j)} W_{i,j,k,l} Y_{i,j}[t - 1] + S_{i,j} \quad (2)$$

$$L_{i,j}[t] = e^{-\alpha_L} L_{i,j}[t - 1] + V_L \sum_{(k,l) \in N(i,j)} W_{i,j,k,l} Y_{i,j}[t - 1] \quad (3)$$

In the modulation field, the signals from $F_{i,j}$ and $L_{i,j}$ channels are combined in a nonlinear way to generate the internal neuron activity expressed by

$$U_{i,j}[t] = F_{i,j}[t](1 + \beta L_{i,j}[t]) \quad (4)$$

where β is a connection factor that regulates the internal activity, which simulates the electrical potential generated in the biological neuron.

An adaptative threshold $\theta_{i,j}$ (Eq. 5) is used for the pulse generator, that operates as a step function, which controls the trigger event $Y_{i,j}$ (Eq. 6). This process simulates the action of polarization and repolarization generated in biological neurons, obviously considering a refractory period dependent on a time interval.

$$\theta_{i,j}[t] = e^{-\alpha_\theta} \theta_{i,j}[t - 1] + V_\theta Y_{i,j}[t] \quad (5)$$

$$Y_{i,j}[t] = \begin{cases} 1 & \text{if } U_{i,j}[t] > \theta_{i,j}[t] \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

3.2 Differential Evolution Algorithm

The DE algorithm is inspired by the natural evolution of individuals within a population, that is, the survival of the fittest. DE maintains a population of potential solutions that mutate and recombine to produce new individuals, which are further evaluated and selected based on their fitness measured by a cost function. The DE process involves the following basic steps:

1. **Initialization:** the population with N individuals is denoted by the set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. For the i th individual, a d -dimensional vector is defined by $\mathbf{x}_i = [x_{i,1}, \dots, x_{i,d}]$, where each variable is randomly initialized in the range $[\text{LL}, \text{UL}]$ representing the lower and upper limits, respectively, of the search space.
2. **Mutation:** for the i th target vector in generation g , $\mathbf{x}_{i,g}$, a mutant vector, $\mathbf{v}_{i,g}$, is created, which combines three members of the population, the current best individual, $\mathbf{x}_{\text{best},g}$, and two individual randomly chosen from the current population, $\mathbf{x}_{r1,g}$, and $\mathbf{x}_{r2,g}$, such that $r1 \neq r2 \neq i$. The mutant vector is generated by using the current-to-best strategy as [8]

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F \cdot (\mathbf{x}_{\text{best},g} - \mathbf{x}_{i,g}) + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \quad (7)$$

where $F = 0.8$ is the scaling factor that controls the amplification of the vector differences.

3. **Crossover:** a test vector, $\mathbf{u}_{i,g}$, is created by exchanging the elements of the target vector $\mathbf{x}_{i,g}$ and the mutant vector $\mathbf{v}_{i,g}$, which is performed by the binomial crossover as

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } \text{rand}(0, 1) < CR \\ x_{i,j} & \text{otherwise} \end{cases} \quad (8)$$

where $j = 1, \dots, d$ and $CR = 0.9$ is the crossover factor that controls the amount of information that is copied from the mutant to the test vector.

4. **Penalty:** in order to prevent the solution falling outside the search space limits $[\text{LL}, \text{UL}]$, the bounce-back strategy [9] is used to reset out-of-bound test variables by selecting a new value that lies between the target variable value and the bound being violated.
5. **Selection:** if the fitness of the test vector $f(\mathbf{u}_{i,g})$ is better than the fitness of the target vector $f(\mathbf{x}_{i,g})$, then $\mathbf{u}_{i,g}$ replaces $\mathbf{x}_{i,g}$ in the next generation, which is expressed by

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases} \quad (9)$$

where $f(\cdot)$ is a cost function, which is minimized without loss of generality.

3.3 Proposed Segmentation Approach

The pseudo-code of the proposed segmentation method based on DE and PCNN is shown in Algorithm 1. Note that each individual in the population codifies the nine PCNN parameters summarized in Table 1, which are randomly initialized using their respective lower and upper limit values. Also, $N = 20$ individuals are considered in the population, which evolves during $G_{\max} = 50$ generations. As mentioned previously, a CVI represents the cost function used by the DE algorithm to evaluate the segmentation quality produced by the PCNN given a potential solution. Here, four CVIs are considered: Calinski-Harabasz (CH) [10], PBM (PBM) [11], Davies-Bouldin (DB) [12] and Xie-Beni (XB) [13]. Depending on the CVI type used by the proposed approach, four algorithm variants are defined: DE-PCNN-CH, DE-PCNN-DB, DE-PCNN-PBM, and DE-PCNN-XB. Besides, for performance comparison purposes, the CVI in the proposed algorithm is replaced by the entropy criterion (ENT) [14] to define the DE-PCNN-ENT algorithm.

Table 1. PCNN parameters and their limit values.

Parameter	Description	Lower limit	Upper limit
α_L	Link attenuation factor	0.01	2.50
α_F	Feed attenuation factor	0.01	2.50
α_θ	Impulse attenuation factor	0.01	2.50
V_L	Link potential	0.01	2.50
V_F	Feed potential	0.01	2.50
V_θ	Impulse potential	1	25
β	Connection factor	0.01	2.50
W	Radius of synaptic weights	1	5
n	Number of iterations	2	10

3.4 Performance Evaluation

For evaluating the proposed approach, an image data set containing 30 natural gray scale images is considered. Every image includes three reference segmentations defined manually by three different persons [15]. The image data set is public and it can be downloaded from http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/2obj/index.html.

The output of a segmentation algorithm (S_A) is compared with a reference segmentation (S_R) by using the Jaccard index defined by

$$J(S_A, S_R) = \frac{|S_A \cap S_R|}{|S_A \cup S_R|} \quad (10)$$

Algorithm 1. Proposed segmentation algorithm based on PCNN tuned by DE.

Require: grayscale image (\mathbf{R}), population size (N), number of generations (G_{\max}), CVI type (f)

Ensure: segmented image (\mathbf{C}^*), best PCNN parameters (\mathbf{x}^*)

Initialize population randomly: $\mathbf{X}_0 = \{\mathbf{x}_{1,0}, \dots, \mathbf{x}_{N,0}\}$

Segment \mathbf{R} with \mathbf{X}_0 using PCNN: $\Omega_0 = \{\mathbf{C}^{\mathbf{x}_{1,0}}, \dots, \mathbf{C}^{\mathbf{x}_{N,0}}\}$

Evaluate initial population: $f(\Omega_0)$

for $g = 1$ **to** G_{\max} **do**

for $i = 1$ **to** N **do**

 Apply mutation strategy (current-to-best): $\mathbf{v}_{i,g}$

 Apply binomial crossover: $\mathbf{u}_{i,g}$

 Apply bounce-back strategy to $\mathbf{u}_{i,g}$

 Segment \mathbf{R} with $\mathbf{u}_{i,g}$ using PCNN: $\mathbf{C}^{\mathbf{u}_{i,g}}$

if $f(\mathbf{C}^{\mathbf{u}_{i,g}}) < f(\mathbf{C}^{\mathbf{x}_{i,g}})$ **then**

 Replace target vector with test vector: $\mathbf{x}_{i,g} \leftarrow \mathbf{u}_{i,g}$

else

 Keep the target vector in the population

end if

end for

end for

Get best individual \mathbf{x}^* and its associated segmentation \mathbf{C}^* that satisfies Eq. 1.

This index returns a value in the range $[0,1]$, where ‘1’ indicates perfect similarity between both segmentations and ‘0’ indicates total disagreement.

In order to statistically determine the segmentation performance of the proposed approach, 31 runs are considered for each algorithm variant. From the Jaccard index results, the median (MED) and the median absolute deviation (MAD) are calculated to determine the central tendency and the dispersion, respectively. These estimators are chosen because they are capable of coping with outliers and non-normal distributions. Additionally, statistical significance analysis is conducted by the Kruskal-Wallis test ($\alpha = 0.05$) to evaluate whether the median values between groups are different under the assumption that the shapes of the underlying distributions are the same. Finally, the wall clock time in seconds is also measured.

The testing platform employed a Linux-based computer with 16 cores at 2.67 GHz (Intel Xeon) and 32 GB of RAM. All the algorithms were developed in MATLAB 2015a (The MathWorks, Natick, MA, USA).

4 Results

The experimental results in terms of the Jaccard index are shown in Table 2. The Kruskal-Wallis test indicated that all groups are statistically significant different ($p < 0.001$), where the DE-PCNN-XB variant attained the best segmentation performance compared to its counterparts, whereas the DE-PCNN-ENT obtained the worst performance.

Table 2. Performance segmentation in terms of the Jaccard index. The results are sorted from the best to the worst performance.

Algorithm	MED	MAD
DE-PCNN-XB	0.738	0.154
DE-PCNN-PBM	0.694	0.201
DE-PCNN-CH	0.660	0.246
DE-PCNN-DB	0.368	0.336
DE-PCNN-ENT	0.251	0.140

Figure 3 shows the computation time of the five algorithm variants, whose median values are in the range 5–7 s. Besides, the DE-PCNN-XB variant obtained the lowest MAD value with 0.42 s, whereas the DE-PCNN-ENT variant reached the largest MAD value with 0.69 s.

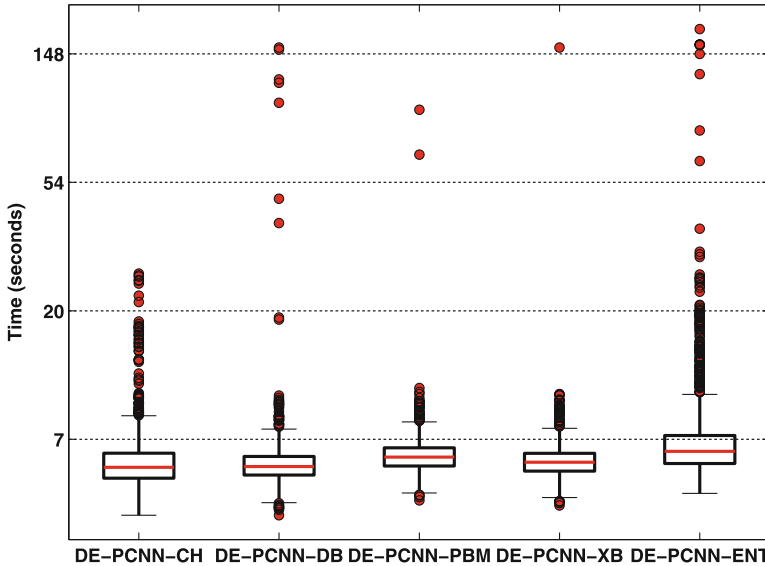


Fig. 3. Boxplots of the computation time regarding the five algorithm variants. For illustrative purposes, the vertical axis is log-compressed.

Figure 4 illustrates a subjective comparison among the outputs of the five segmentation algorithm variants considering four different images (flowers, moth, helicopters, and iceland) from the data set. Notice that all segmentations obtained with the DE-PCNN-XB variant are quite close to their respective reference images. Also, the DE-PCNN-PBM variant is capable to adequately segment three images (flowers, moth, and iceland), although for images where the objects

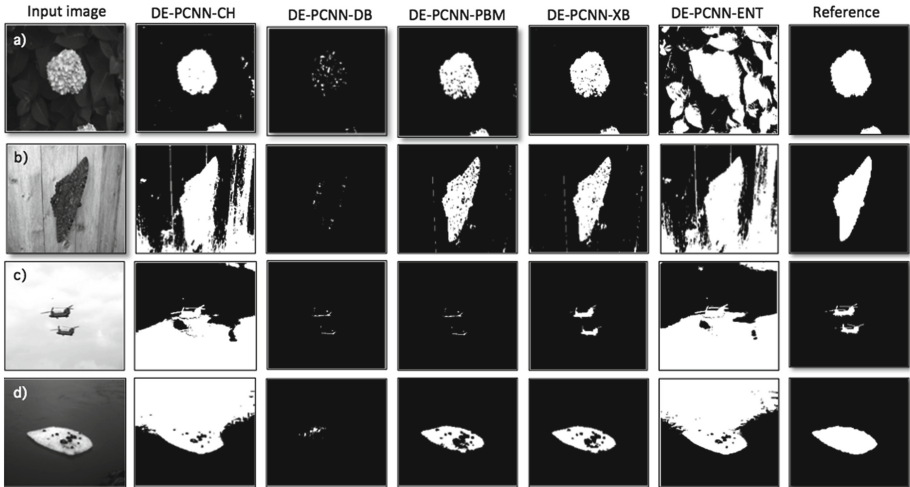


Fig. 4. Segmentation comparison using the five algorithm variants including the reference segmentation. Four images from the data set are considered: (a) flowers, (b) moth, (c) helicopters, and (d) iceland.

are small (such as helicopters) the segmentation is not satisfactory. Finally, DE-PCNN-CH, DE-PCNN-DB and DE-PCNN-ENT failed to properly segment all the images: DE-PCNN-DB tends to under-segment the input image, whereas DE-PCNN-CH and DE-PCNN-ENT tend to over-segment the objects.

5 Conclusion and Future Work

In this paper, a segmentation method based on PCNN tuned by DE algorithm was presented. Five cost functions were evaluated: CH index, PBM index, DB index, XB index, and the maximum entropy criterion. These cost functions quantified the segmentation quality to guide the DE algorithm to find a set of PCNN parameters.

The experimental results pointed out that the DE-PCNN-XB variant obtained the best segmentation performance with low dispersion for distinct algorithm runs. In terms of the Jaccard index, the MED/MAD values were 0.738/0.154. These findings indicated that using a CVI as a cost function is suitable to obtain adequate and consistent segmentations.

Therefore, it was demonstrated that using the XB index instead of the entropy criterion is appropriate to find adequate PCNN parameters to obtain satisfactory segmentation results. This is because a CVI quantifies the relationship between the intensities of the objects and their background, that is, how similar are the intensity levels of the objects and how dissimilar are relative to the background intensities.

Future work involves evaluating other variants of PCNN (e.g., simplified models) as well as other CI-based techniques such as PSO and GA.

Acknowledgments. The authors would like to thanks to CONACyT Mexico for the financial support received through a scholarship to pursue Masters studies at Center for Research and Advanced Studies of the National Polytechnic Institute, Information Technology Laboratory.

References

1. Lindblad, T., Kinser, J.: Image processing using pulse-coupled neural networks. Springer Verlag, Heidelberg (2005)
2. Engelbrecht, A.P.: Computational Intelligence: An Introduction, 2nd edn. Wiley Publishing, Hoboken (2007)
3. Xu, X., Ding, S., Shi, Z., Zhu, H., Zhao, Z.: Particle swarm optimization for automatic parameters determination of pulse coupled neural network. *J. Comput.* **6**, 1546–1553 (2011)
4. Using a genetic algorithm to find an optimized pulse coupled neural network solution. vol. 6979 (2008)
5. A Self-Adapting Pulse-Coupled Neural Network Based on Modified Differential Evolution Algorithm and Its Application on Image Segmentation. vol. 6 (2012)
6. Gómez, W., Pereira, W., Infantosi, A.: Evolutionary pulse-coupled neural network for segmenting breast lesions on ultrasonography. *Neurocomputing* **129**, 216–224 (2015)
7. Wang, Z., Ma, Y., Cheng, F., Yang, L.: Review of pulse-coupled neural networks. *Image Vis. Comput.* **28**, 5–13 (2010)
8. Zhang, J., Sanderson, A.: Jade: Adaptive differential evolution with optional external archive. *Evol. Comput. IEEE Trans.* **13**, 945–958 (2009)
9. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Natural Computing Series. Springer, Heidelberg (2005)
10. Caliski, T., Harabasz, J.: A dendrite method for cluster analysis. *Commun. Stat.* **3**, 1–27 (1974)
11. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. *Pattern Anal. Mach. Intell. IEEE Trans.* **24**, 1650–1654 (2002)
12. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *Pattern Anal. Mach. Intell. IEEE Trans.* **PAMI-1**, 224–227 (1979)
13. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. *Pattern Anal. Mach. Intell. IEEE Trans.* **13**, 841–847 (1991)
14. Ma, Y., Qi, C.: Study of automated pcnn system based on genetic algorithm. *J. Syst. Simul.* **18**, 722–725 (2006)
15. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. *Pattern Anal. Mach. Intell. IEEE Trans.* **29**, 929–944 (2007)