International Journal of Hybrid Information Technology
Vol.6, No.5 (2013), pp.131-140
http://dx.doi.org/10.14257/ijhit.2013.6.5.12

Hybrid PSO and Genetic Algorithm for Multilevel Maximum Entropy Criterion Threshold Selection

Elham Akbari Baniani and Abdolah Chalechale

Department of Computer Engineering, Razi University, Kermanshah, Iran

Baniani87@gmail.com, Chalechale@razi.ac.ir

Abstract

Multilevel thresholding is one of the most important techniques for image processing and pattern recognition. The maximum entropy thresholding (MET) has been widely applied in multilevel thresholding. In this paper, a novel multilevel MET algorithm based on the hybrid of particle swarm optimization (PSO) and Genetic algorithm is presented. In standard PSO the non-oscillatory route can quickly cause a particle to stagnate and also it may prematurely converge on suboptimal solutions that are not even guaranteed local optimal solution. To overcome this problem, we used Genetic algorithm. To obtain an optimal solution in Genetic algorithm, operation such as selection, reproduction, and mutation procedures are used to generate next generations. The capability of this hybrid PSO that called HPGT is enhanced by cloning of fitter particles instead of worst particles that is determined based on their fitness values. The performance of HPGT algorithm and PSO algorithm compared. The results show the convergence of the HPGT is very good.

Keywords: *Image Segmentation; Thresholding; Genetic Algorithm; particle swarm optimization Algorithm*

1. Introduction

Image segmentation is the most widely studied problem in image processing and a considerable research effort has been carried out on this problem. Despite this, segmentation is one of the unsolved grand challenges in computer vision. Image segmentation is applicable in an endless list of areas and applications, for example: medical, outdoor object recognition, robot vision, content-based image retrieval [1-3]. The main goal of segmentation is to subdivide an image into its constituent regions or objects [4].

One of the most popular techniques for image segmentation is thresholding that is based on the histogram. Histogram-based thresholding is commonly known as a very popular tool for image segmentation. Here, the objective is to determine an accurate threshold (for bi-level thresholding) or multiple thresholds (for multilevel thresholding), so that the image can be subdivided into several levels, for easier analysis and interpretation. Bi-level thresholding is the simplest problem, where in the histogram of the image grabbed, only one single valley is found and accordingly the pixels are grouped into two classes: one group of pixels with image intensity above the threshold and another below the threshold. Multilevel thresholding problems are more complicated and the corresponding image segmentation problem can be configured as a multiclass classification problem where, based on the determined thresholds, pixels having a particular characteristic, within a specified range, are grouped into one class. Usually it is not simple to determine exact location of distinct valleys in a multimodal histogram of an image, that can segment the image efficiently and hence the problem of

ISSN: 1738-9968 IJHIT

multilevel thresholding is regarded as a more challenging task. Excellent reviews on early thresholding methods can be found in [5, 6], whereas the latest developed in this topic was in [7].

So far, several algorithms have been proposed for bi-level and multilevel thresholding of image histograms. Many of these methods attempt to achieve optimization of an objective function by, *e.g.*, maximizing posterior entropy that indicates homogeneity of segmented classes [8], maximizing some measure of separability [9], minimizing Bayesian error [10] etc. To solve the multilevel thresholding problem have been proposed several techniques using genetic (GAs) and PSO algorithms [11-16]. The Genetic Algorithm (GA), proposed by Holland [11], is a probabilistic optimal algorithm that is based on the evolutionary theories. GAs are optimization algorithms based on the mechanics of natural selection and natural genetics.

Beside GAs, particle swarm optimization (PSO) is another latest evolutionary optimization technique which was used for the multilevel thresholding. The PSO, first introduced by Kennedy and Eberhart [15] is a flexible, robust, population based on stochastic search/optimization algorithm with inherent parallelism. The method presented in [16] uses improved PSO to optimize the Maximum Entropy Criterion. Both of PSO and GA have problem for convergence to local optimal point [17-18]. In this paper, we propose a novel optimal multilevel thresholding algorithm for histogram-based image segmentation. In this method, we used the feature of genetic algorithm for precocious convergence of PSO algorithm. We use Maximum Entropy Criterion fitness function for evaluating the algorithm performance.

The rest of the paper is organized as follows: Section 2 is definition of multilevel thresholding formula. In the Section 3 the proposed technique is presented that is hybrid of PSO and Genetic algorithms. The experimental results are studied in Section 4 and Section 5 is conclusion.

2. Multilevel Thresholding Problem Formulation

2.1 Entropy criterion based on measure

The entropy criterion, proposed by Kapur in 1985, was widely used in determining the optimal thresholding in image segmentation. The original algorithm had been developed for bi-level thresholding. The method can also extend to solve multilevel thresholding problems and can be described as follows: Let there be L gray levels in a given image I and these gray levels are in the range {0,1,…,L-1}. Then one can define P_i=h(i)/N, $(0 \le i \le L-1)$ where h(i) means the number of pixels with gray-level i and N means total number of pixels in the image.

To select D threshold, $[t_1,t_2,\ldots,t_D]$ for a given image I, the objective function *f* should be maximize:

$$f([t_1,t_2,\ldots,t_D])=H_0+H_1+\ldots+H_D \tag{1}$$

$$w_0=\sum_{i=0}^{t_1-1} P_i \quad H_0=-\sum_{i=0}^{t_1-1}\frac{P_i}{w_0}\ln\frac{P_i}{w_0}$$

$$w_1 = \sum_{i=t_1}^{t_2-1} P_i \quad H_1 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1} \ln \frac{P_i}{w_1} \cdots$$

$$w_D = \sum_{i=t}^{t_2-1} P_i \;\; H_D = -\sum_{i=t_c}^{L-1} \frac{P_i}{w_D} \ln \frac{P_i}{w_D}$$

The maximum entropy thresholding method has been proven as an efficient method for bi-level thresholding in image segmentation. However, when these methods are extended to multilevel thresholding, the computation time grows exponentially with the number of thresholds. It would limit the multilevel thresholding applications. To overcome the above problem, this paper proposes the maximum entropy thresholding based on HPGT algorithm for solving multilevel thresholding problem. The aim of this proposed method is to maximize the entropy criterion objective function using Equation (1).

3. Image thresholding based on PSO and Genetic algorithm

In this section, firstly basic principles of the PSO algorithm and Genetic algorithm will be described, and then the proposed method that is a hybrid of PSO and GA to be explained.

3.1. PSO Algorithm

Particle swarm optimization (PSO) is an evolutionary computation technique motivated by simulation of social behavior. PSO is similar to a Genetic algorithm (GA) that both algorithms are initialized with a population of random solutions.

The basic idea of the PSO is the mathematical modeling and simulation of the food searching activities of a swarm of birds that called particles. To moving the particles in the space of problem, they need velocity. The velocity of a particle is influenced by three components including internal momentum, cognitive, and social. The internal component simulates the inertial behavior of the bird to fly in the previous direction. The cognitive component models the memory of the bird about its previous best position, and the social component models the memory of the bird about the best position among the particles.

PSO procedure based on the above concept can be described as follows. Bird flocking optimizes a certain objective function. Each particle knows its best value so far (P_{id}) and its position (X_{id}). Moreover, each particle knows the best value in the group (P_{gd}) among P_{id}s. Each particle tries to modify its position using the current velocity and the distance from the P_{id} and P_{gd}. Based on the above discussion, the mathematical model for PSO is as follows:

Velocity update equation is given by:

$$V_{id}^{new} = w \times V_{id}^{old} + c_1 \times rand_1 \times (P_{id} - X_{id}^{old}) + \\ c_2 \times rand_2 \times (P_{gd} - X_{id}^{old}) \tag{2}$$

Using Equation (2), a certain velocity that gradually gets close to P_{id} and P_{gd} can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$X_{id}^{new} = X_{id}^{old} + V_{id}^{new} \tag{3}$$

$$w = w_{max} - (w_{max} - w_{min}) \times Iter / Iter_{max} \tag{4}$$

Where W_{max} and W_{min} are initial and final weight, respectively and *Iter* is current iteration number and $Iter_{max}$ is maximum iteration number.

Where

V_{id} : velocity of particle i in d dimension

X_{id} : position of the particle

w : inertia weight

c_1 and c_2: cognition and social acceleration coefficient

P_{id}: own best position of particle i

P_{gd}: global best position among the group of particles

$rand_1$ and $rand_2$: uniformly distributed random numbers in the range [0 to 1].

3.2. Genetic Algorithm

GAs is efficient and robust search and optimization techniques guided by the principles of evolution, and have implicit parallelism. The essential components of GAs are the following:

1- A representation strategy called chromosomes.

2- A population of chromosomes.

3- Mechanism for evaluating each string or fitness function.

4- Selection/reproduction procedure.

5- Genetic operators such as crossover and mutation.

A population is created with a group of individuals generated randomly. The individuals in the population are evaluated. The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. In order to GA starts with the current population. Selection is applied to the current population to create an intermediate population. Then crossover and mutation are applied to the intermediate population to create the next population. Between the new population and their parents are selected individuals to execute again genetic algorithm.

3.3. Proposed method

PSO shares many similarities with evolutionary computing techniques such as GAs. PSO and GA techniques begin with a group of a randomly generated population; both utilize a fitness value to evaluate the population. They update the population and search for the optimum with random techniques. The difference between them is that PSO lacks crossover and mutation and more easily to falls into local optimal solutions.

As mentioned, Although PSO is faster in finding quality solutions; the drawback of PSO is that the swarm may prematurely converge. The underlying principle behind this problem is that, for the global best PSO, particles converge to a single point, which is on the line between the global best and the personal best positions. Another reason for this problem is the fast rate of information flow between particles, resulting in the creation of similar particles with a loss in diversity that increases the possibility of being trapped in local optima. For solving the drawback of PSO, we can used the GA. GA can be utilized to change chromosomes and escape the local optimal solution and quickly attain the global optimal solution.

In this paper, we propose a hybrid of the PSO and GA methods can produce a very effective search strategy. According to the idea, the global optimal solution cannot be found when the optimal particle is a local optimal solution. Thus, this study added GA to overcome this drawback. So, according to our idea a particle is generated through PSO, and after a few iterations, if that particle is appropriate then is selected for crossover and mutation operators (by the best particle in the swarm) in GA to generate next generation. This will obtain better results than using only PSO or GA. The evolutionary steps for the hybrid algorithm are as follows.

Step1: Set up the values of parameters, including the population size (number of particles is equal to N), maximum and minimum inertia weight (W_{max}, W_{min}), learning factors (C_1, C_2), the number of thresholds (Nt). Also, set up mutation rate for Genetic algorithm section.

Step2: Generate randomly the initial position (X_i) and velocity (V_i) of each particle where $X_i = (x_1, x_2,\ldots,x_{Nt})$ and $V_i = (v_1, v_2,\ldots,v_{Nt})$. In which, $x_i \in \{1,\ldots,255\}$ and indicates characteristic of the i^{th} particle.

Step3: Calculate the fitness values of all particles to measure their performance. In this paper, we use Kapur approach for calculating fitness.

Step 4: Select P_{id} and P_{gd}. As mentioned, P_{id} is best position of particle i and P_{gd} is global best position among the group of particles.

Step5: Update the velocity (V_i) and position (X_i) of each particle according to Equation (2) and Equation (3).

Step 6: Return to Step 3 until the pre-specified number of iterations is satisfied.

Step 7: Select N/2 of the best particles according to the fitness for crossover and mutation operation to generate next population. In this paper, we do crossover between P_{gd} and N/2 particles, and do mutation in the P_{gd} to generate the population 2.

Step 8: Perform elitist selection for population 1 and 2 to generate the next iterative population. Population 1 is best position of N particles in PSO algorithm and population 2 is generated by crossover between P_{gd} and N/2 particles in step 7.

Step 9: This algorithm will not stop returning to Step 3 until a pre-specified number of iterations is satisfied.

4. Experimental Results

In this section, we compared proposed approach with the PSO thresholding algorithm. We tested some standard gray-level images using Matlab 2009 software running at XP windows, 2.5GHz CPU, 4G RAM. Table 1 and 2 represents the various parameters chosen for the implementation of HPGT and PSO algorithms, respectively. Since, the PSO algorithm is better than the Genetic algorithm, we compared HPGT algorithm only with the PSO algorithm. In Figure 1, four well-known images namely Lena, Pepper, Cameraman and Boats are taken as the test images and are gathered with their histograms.

The quality of threshold images for maximum entropy thresholding method has been evaluated in Tables 3 and 4. The Table 3 shows the optimal thresholds values and also shows objective values for HPGT and PSO algorithms. It is observed from objective values tables that in each case, the HPGT algorithm could perform well as compared with the PSO algorithm. In Table 3, the higher value of the objective function results in better segmentation.

In order to consistent comparisons, a performance indicator that is peak signal to noise ratio (PSNR) is used. For the sake of completeness we define PSNR, measured in decibel as:

$$PSNR = 20\log\left(\frac{255}{RMSE}\right) \tag{5}$$

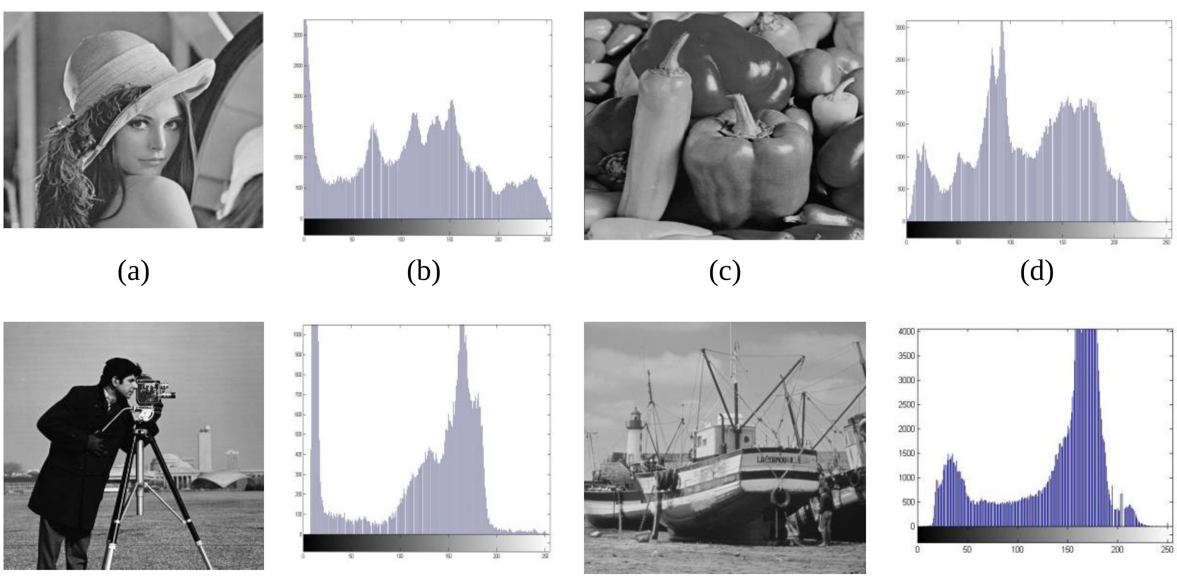
Where RMSE is the root mean-squared error, defined as

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - \hat{I}(i,j))^2} \tag{6}$$

Here I and $\hat{I}$ are original and thresholded images of size $M \times N$, respectively. The higher value of PSNR shows that the results of experiment are better. The PSNR values from the maximum entropy thresholding function are given in Table 4.

Another parameter for comparison of HPGT and PSO algorithms is standard deviation. The standard deviation values from maximum entropy thresholding are given in Table 4. The higher value of standard deviation shows that the results of experiment are not stable. From the table, it is seen that the HPGT algorithm is more stable than the PSO algorithm.

Figure 2 shows the segmentation results, the segmented Lena, Pepper, Cameraman and Boats images for maximum entropy thresholding function with 3 and 5 level of thresholding. According to these figures, the quality of the segmented images is better by selecting the higher level thresholds.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure1. Test Images and their histograms (a) Lena, (c) Pepper, (e) Cameraman, (g) Boats

Table 1. Parameters chosen for PSO implementation

number	Parameters of PSO	
	Parameters	*value*
1	Population size	30
2	No. of iteration	100
3	W_{max},W_{min}	5,-5
4	C_1,C_2	2

Table 2. Parameters chosen for HPGT implementation

number	Parameters of HPGT	
	Parameters	*Value*
1	Population size	30
2	No. of iteration	10
3	No. of iteration of PSO	8
4	W_{max},W_{min}	5,-5
5	C_1,C_2	2
6	Crossover probability	1
7	Mutation probability	0.1

Table 3. Comparison of optimal threshold values and objective values

Test images	Optimal threshold values		Objective values	
	HPGT	*PSO*	*HPGT*	*PSO*
Lena	98,173	98,173	12.9513	12.9513
	75,131,188	76,133,189	16.1177	16.1165
	66,113,163,208	66,113,161,206	19.0709	19.0686
	48,89,129,169,211	52,95,134,171,211	21.8315	21.8247
Pepper	70,142	70,142	12.5567	12.5567
	64,113,163	62,111,160	15.5837	15.5824
	55,99,143,188	50,98,144,188	18.4383	18.4362
	40,73,109,147,188	40,71,104,142,186	21.1440	21.1266
Cameraman	127,192	128,195	12.1688	12.1677
	43,103,192	43,103,193	15.2274	15.2241
	43,96,145,196	41 ,91,142,196	18.3955	18.3705
	24,61,99,146,194	25,63,97,143,193	21.1446	21.1273
Boats	72,133	72,133	12.1772	12.1772
	72,133,189	73,133,190	15.3231	15.3178
	53,98,141,189	56,101,141,190	18.2497	18.2436
	49,85,120,153,189	55,94,126,155,189	20.9417	20.9060

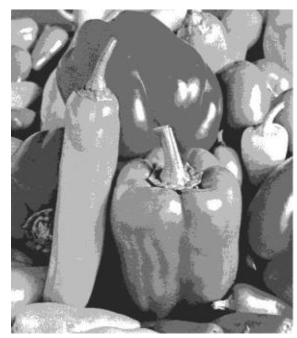
Table 4. Comparison of PSNR values and SD values

Test images	PSNR values		SD values	
	HPGT	*PSO*	*HPGT*	*PSO*
Lena	13.2429	13.2429	5.3e-15	2.5e-4
	15.7485	15.7206	3.2e-5	0.0034
	17.1694	17.2101	2.6e-4	0.0116
	19.4042	19.0872	0.0032	0.0147
Pepper	12.1587	12.1587	7.1e-15	2.7e-4
	15.3839	15.1256	6.6e-6	0.0014
	19.2523	19.2466	0.0187	0.0320
	20.1162	19.8618	0.0223	0.0390
Cameraman	12.3841	12.2285	1.9e-14	8.4e-4
	16.0991	15.9656	1.3e-4	0.0681
	18.8492	18.7995	5.2e-4	0.1021
	20.2514	20.2094	0.0414	0.1321
Boats	10.5027	10.5027	5.3e-15	0.0180
	13.2429	13.2429	5.3e-15	2.5e-4
	15.7485	15.7206	3.2e-5	0.0034
	17.1694	17.2101	2.6e-4	0.0116


(a)


(b)


(c)


(d)


(e)


(f)


(g)


(h)

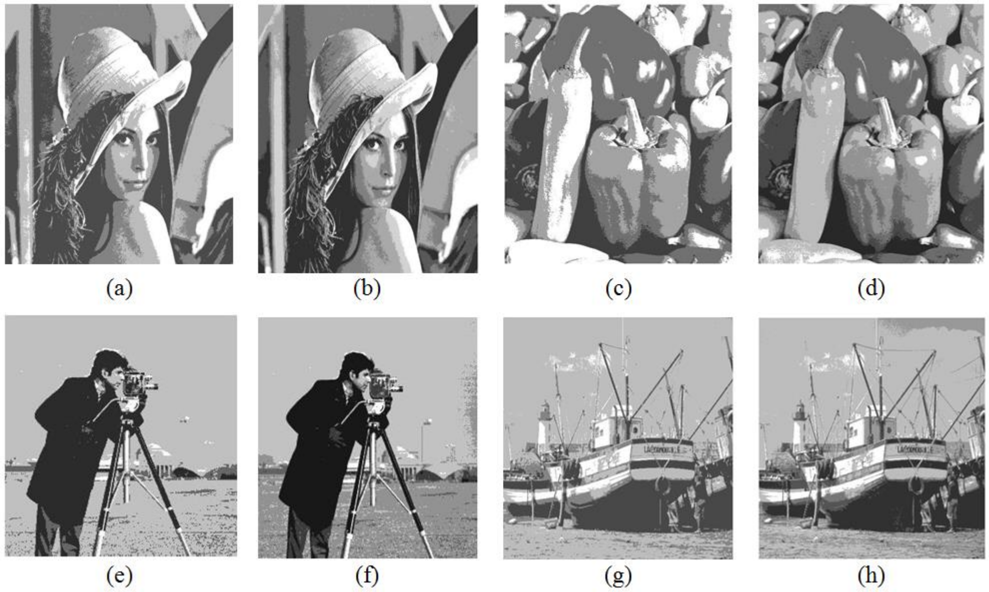


Figure 2. Threshold images obtained by maximum entropy thresholding method by 3-level thresholds (a), (c), (e) and (g), and by 5-level thresholds (b), (d), (f) and (h)

5. Conclusion

In this paper, we have presented a new algorithm that is based on HPGT algorithm. In this algorithm, we combine the ability of fast convergence of PSO algorithm with the ability of GA for escape from the local optima. For this purpose, the PSO algorithm runs for number of iteration, then GA algorithm generate new offspring by mutation and crossover operations. This new offspring replacement by worst individuals in the generation and the PSO algorithm run by new individuals. After some iteration, the algorithm is converged to a best result.

In this paper, we use Kapur to evaluate fitness function. The PSNR and standard deviation are computed for this function. The performance of HPGT algorithm has been compared with PSO algorithm, and we found that HPGT algorithm is better than PSO algorithm in terms of solution quality, convergence and robustness.

References

[1] Z. Maa, J. Tavares, R. Jorgea and T. Mascarenhas, “A Review of Algorithms for Medical Image Segmentation and their Applications to the Female Pelvic Cavity”, Computer Methods Biomechanics and Biomedical Engineering, vol. 13, **(2010)**, pp. 235-246.

[2] J. Chamorro-martinez, D. Sanchez and B. Prados-Suarez, “A Fuzzy Colour Image Segmentation Applied to Robot Vision”, Advances in Soft Computing-Engineering, Design and Manufacturing, Springer, **(2003)**, pp. 129-138.

[3] N. Singhai and S. K. Shandilya, "A Survey On: Content Based Image Retrieval Systems", International Journal of Computer Applications, vol. 4, no. 2, **(2010)** July, pp. 485-490.

[4] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Addison-Wesley, Reading, MA, **(2002)**.

[5] P. K. Sahoo, S. Soltani and A. K. C. Wong, "A survey of thresholding techniques", computer vision, Graphics and image processing, vol. 41, no. 2, **(1988)** February, pp.233-260.

[6] N. P. Pal and S. K. Pal, "A review on image segmentation techniques", Pattern Recognition, vol. 26, no. 9, **(1993)** September, pp. 1277-1294.

[7] M. Sezgin and B. Snakar, "Survey over Image Thresholding Techniques and Quantitative performance evaluation", Journal of Electronic Imaging, vol. 13, no. 1, **(2004)** January, pp. 146-165.

[8] J. N. Kapur, P. K. Sahoo and A. K. C Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," Computer Vision Graphics Image Processing, vol. 29, **(1985)** March, pp. 273–285.
[9] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Transactions on Systems, Man, Cybernetics, vol. 9, **(1979)** January, pp. 62–66.
[10] Q. Ye and P. Danielsson, "On Minimum Error Thresholding and Its Implementations", Pattern Recognition Letters, vol. 7, no. 4, **(1988)**, pp. 201–206.
[11] J. Holland, "Adaption in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, MI, (**1975)**.
[12] P. Y. Yin, "A Fast Scheme for Optimal Thresholding Using Genetic Algorithms", Signal Processing, vol. 72, **(1999)**, pp. 85−95.
[13] N. Razmjooy, B. S. Mousavi, P. Sargolzaei and F. Soleymani, "Image Thresholding based on Evolutionary Algorithms", International Journal of the Physical Sciences, vol. 6, **(2011)** November, pp. 7203-7211.
[14] O. Banimelhem and Y. A. Yahya, "Multi-Thresholding Image Segmentation Using Genetic Algorithm", World Congress in Computer Science, Computer Engineering, and Applied Computing, (**2011)**.
[15] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE Conference on Neural Networks – ICNN"95, Perth, Australia, **(1995)**, pp. 1942-1948.
[16] M. Maitra and A. Chatterjee, "A Hybrid Cooperative–Comprehensive Learning Based PSO Algorithm for Image Segmentation Using Multilevel Thresholding", Expert Systems with Applications, vol. 34, **(2008)**, pp. 1341–1350.
[17] M. Settles and T. Soule, "Breeding Swarms: A GA/PSO Hybrid", Genetic and evolutionary computation, **(2005)**, pp. 161-168.
[18] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", In e. a V. William Porto, editor, Evolutionary Programming, Vol. 1447 of Lecture Notes in Computer Science, **(1998)**, pp. 611–616, Springer.

