International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:8, No:10, 2014

Nature Inspired Metaheuristic Algorithms for Multilevel Thresholding Image Segmentation - A Survey

C. Deepika, J. Nithya

Abstract**—**Segmentation is one of the essential tasks in image processing. Thresholding is one of the simplest techniques for performing image segmentation. Multilevel thresholding is a simple and effective technique. The primary objective of bi-level or multilevel thresholding for image segmentation is to determine a best thresholding value. To achieve multilevel thresholding various techniques has been proposed. A study of some nature inspired metaheuristic algorithms for multilevel thresholding for image segmentation is conducted. Here, we study about Particle swarm optimization (PSO) algorithm, artificial bee colony optimization (ABC), Ant colony optimization (ACO) algorithm and Cuckoo search (CS) algorithm.

Keywords**—**Ant colony optimization, Artificial bee colony optimization, Cuckoo search algorithm, Image segmentation, Multilevel thresholding, Particle swarm optimization.

I. Introduction

SEGMENTATION is generally the first stage in any attempt to analyze or interpret an image automatically. The role of segmentation is crucial in most tasks requiring image analysis. A reliable and accurate segmentation is very difficult to achieve by purely automatic means. Image segmentation algorithms generally are based on two basic properties of intensity values: discontinuity and similarity. The discontinuity based approach is partitioning an image based on abrupt changes in intensity. The similarity based approach is partitioning an image based on regions that are similar according to a set of predefined criteria. Thresholding method is a similarity based approach. The simplest method of image segmentation is thresholding method, which has many applications in image processing, including segmentation, clustering, classification, and so on [1], [2]. The segmented image obtained from thresholding has the benefit of smaller storage space, fast processing speed. The advantage of thresholding technique is that, it is simple to implement and fast. Thresholding techniques can be divided into bi-level and multilevel category, depending on the number of image segments. Separating the objects in an image from the background by determining a single threshold value is termed as bi-level thresholding [3]. Consider an image, represented by L gray levels, bi-level thresholding can be described as follows:

$$M_0 = \{\, g(x,y) \in I \mid 0 \le g(x,y) \le t-1\}$$
$$M_1 = \{g(x,y) \in I \mid t \le g(x,y) \le L-1\}$$

Thresholding generates from gray-level to binary images by changing all pixels below some threshold to zero and all pixels about that threshold to one. Dividing an image into several regions is called as multilevel thresholding. Multilevel thresholding is a process that segments a gray level image into numerous distinct regions. Multilevel Thresholding utilize more than one threshold value and creates an output image with multiple groups as:

$$M_0 = \{g(x,y) \in I \mid 0 \le g(x,y) \le t_1 - 1\}$$
$$M_1 = \{g(x,y) \in I \mid t_1 \le g(x,y) \le t_2 - 1\}$$
$$M_i = \{g(x,y) \in I \mid t_i \le g(x,y) \le t_{i+1} - 1\}$$
$$M_m = \{g(x,y) \in I \mid t_m \le g(x,y) \le L - 1\}$$

where, t_i (i = 1,…,m) is the ith threshold value, and m is the number of thresholds.

The multilevel thresholding uses a new class of algorithm called metaheuristics. Metaheuristics are mainly a higher level procedure, which produces a simpler way to solve an optimization problem [4], [5]. Many metaheuristic algorithms have inspiration coming from nature. There are many such examples where the organisms (a population or a swarm) have optimized and adapted themselves to survive in this world. Some nature inspired metaheuristic algorithms are: Genetic Algorithm [6]-[8], Particle Swarm Optimization [9], [10], Cuckoo Algorithms, Bacterial Foraging [10], [11] and many more. Heuristic methods for finding optimal thresholds attained the attention of researchers due to the computational inefficiency of the traditional exhaustive method [12]. The use of these algorithms was widely spread because of its high-quality solutions for difficult problems [13]. A survey is conducted on the following nature inspired metaheuristic algorithms: Particle swarm optimization (PSO), Artificial bee colony optimization (ABC), Ant colony optimization (ACO) and Cuckoo search algorithms.

II. Particle Swarm Optimization

Particle swarm optimization is motivated by the swarming behavior of animals like bird flocking. This optimization [14] was initiated by Kennedy and Eberhart. In PSO, a swarm is a

Ms.C.Deepika, Post Graduate Scholar, Department of Information Technology, K. S. Rangasamy College of Technology, Namakkal District, Tamil Nadu India (Mobile: 9487862755 ; e-mail: deepikac1611@gmail.com)
Mrs. J.Nithya, Associate Professor, Department of Information Technology, K. S. Rangasamy College of Technology, Namakkal District, Tamil Nadu India (Mobile: 9443846125;e-mail:nithyajk@yahoo.co.in)

collection of particles which is moving in a search space. Depending on earlier occurrence and the best occurrence of the swarm, the particles vary their positions to find the global optimum.

Algorithm 1 (Main steps of the PSO algorithm)

1. Initialize the population
2. **Repeat**
3. Estimate the fitness values of the particles
4. Update the best experience of each particle
5. Pick the best particle
6. Estimate the velocities of the particles
7. Update the positions of the particles
8. **Until** requirements are met

Each and every particle consists of a position vector T, which corresponds to the candidate solution to the optimization problem F(T) of solution T, a velocity vector E and a memory vector Tbest of the best vector solution met by the particle with its recorded fitness.

The particle's position is given by,

$$T(t+1) = T(t) + E(t) \tag{1}$$

and its velocity in accordance with

$$E(t+1) = E(t) + \varphi_1\left(Tbest - T(t)\right) + \varphi_2\left(T^* - T(t)\right) \tag{2}$$

where φ_1, φ_2 are uniformly distributed random numbers within the range of [φ_{min}=0,φ_{max}=2]

The problem of multilevel thresholding deals with finding optimal thresholds within the range [0, L−1] that maximize a fitness condition. The dimension of the optimization problem is the number of thresholds (m), and the search space is [0, L−1] m. In the initialization phase of the algorithm, a population of the solutions is generated at random within the range [0, L − 1] on each dimension.

In the PSO algorithm, the position of the particle represents the thresholds, and the aim is to discover the optimal thresholds by altering the velocities and the positions of particles in the search space. The PSO algorithm iterates the steps until a termination condition is satisfied.

III. Artificial Bee Colony Algorithm

Karaboga [15] introduced an algorithm based on the behavior of honey bees called an artificial bee colony algorithm. In the natural world, there exists a division of labor in the hive and the forager bees work as a group without a central control system to maximize the amount of nectar loaded into the hive.

In the artificial bee colony algorithm, the position of each food source refers to the solution. A source was selected by each bee and the neighborhood of the solution is examined. The foraging process involves three types of bees. They are employed bees, onlooker bees and scout bees. The categorization of bees is based on the fact that how they decide on the food source to utilize. An employed bee hunts the neighborhood of the source in her memory. If the employed bee discovered an improved solution, then it will update her memory; if not the employed bee calculates the number of hunts around the source in her memory. An onlooker bee chooses a possibly beneficial food source and she does not contain any source in her memory. An onlooker bee hunts the neighborhood of the source while they pick a source. When she discovers a better solution, updates the position of the food source same as an employed bee does. Information regarding the profitability of the source is collected from the experiences of the employed bees.

If the number of hunts associated with the source surpasses the limit, then it implies that the solution has been exploited suitably and it could be believed to be exhausted. The bee, which belongs to the exhausted source abandon her source and turn out to be a scout. A scout bee chooses a random source to exploit [16]-[18]. The main phases of ABC algorithm are:

Algorithm 2 (Main steps of the ABC algorithm)

1. Initialization phase
2. Evaluation phase
3. **Repeat**
4. Employed Bee Phase
5. Onlooker Bee Phase
6. Scout Bee Phase
7. Remember the most excellent solution attained so far
8. **Until** A termination condition is satisfied

In the first phase of initialization, the population of food source is produced randomly by,

$$Y_{ij} = y_j^{min} + rand(0,1)(y_j^{max} - y_j^{min}) \tag{3}$$

where $\vec{y}_I$ is said to be the position of ith particle, i=1,...,ZZ, ZZ is the swarm size, j=1,...,H and H is said to be the dimension of the problem.

In the phase of employed bee, a search is carried out by (4) around the source in each bee's memory. If the solution obtained from (4) is better than the solution in the bee's memory, then the memory is updated by a greedy selection approach:

$$Y'_{ij} = y_{ij} + \Phi_{ij}(y_{ij} - y_{kj}) \tag{4}$$

where k $\in$ [1,CS] is the uniform random index, CS is the number of food sources, j $\in$ [1,H] is the uniform random index, k is the dimension of the problem.

The employed bees share their experience through the onlookers, which is fulfilled by passing on each solution a normalized fitness based probability:

$$P_i = \frac{fitness}{\sum_{i=1}^{cs} fitness} \tag{5}$$

Each onlooker bee picks a source by a probabilistic, roulette wheel like the selection. After picking up a source, a search is carried out locally by (4) and a greedy selection is applied to maintain the solution that is better for the population. In employed bee and onlooker bee phases, if a solution cannot be advanced by local searches during a number of cycles, that source is said to weaken and its bee turns out to be a scout in the phase of scout bees'. A new random solution generated by (4) is discovered by scout bee and that solution will be replaced with exhausted source.

In the ABC algorithm, the position of food source corresponds to the thresholds, and the algorithm iterates the employed bee, the onlooker bee and the scout bee phases to select some sources and to execute the search operators by (4).

IV. Ant Colony Optimization

Ant Colony Optimization (ACO) is a meta-heuristic and a population based approach, which was proposed by Dorigo in order to solve a number of discrete optimization problems [19], [20]. The ACO mimics the way of the real ants in order to find the shortest route between a food source and their nest. The communication between ants takes place using the pheromone trails and they exchange information regarding which path should be followed. When a more number of ants traces a given path, then that path becomes more attractive and that path will be followed by all other ants by depositing their own pheromone. This cooperative behavior results in the launching of the shortest route. This real-life behavior leads to the formulation of artificial ant algorithms. The first algorithm proposed was called Ant System (AS). It is not competitive compared to other established approaches, but its computational results were promising. Ant Colony System (ACS) and Max – Min Ant System is the enhanced versions that have been proposed.

In ACO, at every generation a colony of N simple ants or artificial ants search for better and better solutions. The ants develop set of solutions and form a population of threshold values. The threshold values generated by the ant j is noted as pop(j)_t_k (k = 1,2,.....,K). Every artificial ant j of a generation builds up a solution T= ($t_1,t_2,...,t_k,...,t_k$) by using provided information given by the pheromone matrix denoted as τ, where each and every element τ_{ik} is the pheromone trail which lies in the gray level i to the kth threshold, i=0,2;..., L-1 and K=1,2,...,k. The trail pheromones τ_{ik} at first are generated randomly in the range of [0,1]

In order to generate solution T, the agent selects the threshold value for each and every element of string T according the following rule: using the probability q_0, the gray value having the highest pheromone concentration is selected as threshold value, i.e. Pop(j)_t_k = argmaxτ_{ik}. Otherwise the kth threshold value is i ∈ [g_{min} g_{max}] and this is determined using a stochastic distribution with a probability of (1-q_0), such that: pop(j)_t_k = rand[g_{min} g_{max}]. Q_0 is the priori defined number, 0<q_0<1. This rule is called pseudo-random-proportional rule in ACS.

Pseudo-Code for Ant Colony Optimization

1- Create the pheromone matrix τ
2- Initialize the population *pop*
3- Store the best solution *T** of the population *pop* with its fitness in a separate location.
4- For a fixed number of iterations

"From the pheromone matrix τ, determine the population *pop*"
For all persons j in population *pop*
If rand > q_0 then
Pop(j)_t_k = arg max τ_{ik} i∈[g_{min} g_{max}]
Else *pop(j)_t_k = $rand_{int}$*[g_{min} g_{max}]
Evaluate population *pop* "evaluate all candidate solutions"
"Update the best solution T*"
Compare the best individual T of the *pop* with T*. If *T* has a fitness better than T*, then replace T* with T.F_{max}=F(T*)
"Update the pheromone trails"
$\tau_{ik} = \rho\tau_{ik} + (1 \quad \rho)F_{max}$
Endfor // iteration

As soon as all ants have built up their solutions, then these solutions were evaluated according to the objective functions. If it has a better objective function value than that of the best solution in memory, the value of best solution in memory (T*) is updated through the value of the solution obtained as "current iteration best solution". Update the pheromone trails which is an important process. The updating process of trail of the algorithm is carried out as follows:

$$T_{ik} = \rho\tau_{ik} + (1 - \rho)\Delta\tau \qquad (6)$$

where ρ is the persistence of trail which lies within the range of [0,1] and (1-ρ) is the evaporation rate. $\Delta\tau$ indicates the amount of pheromone trail added to τ_{ik} by the best ant corresponding to the best solution found until now: $\Delta\tau$=F_{max}, this is the fitness of best solution T*.

Such a pheromone updating process mirrors the usefulness of dynamic information provided by the artificial ants. Thus, the pheromone matrix is a sort of adaptive memory that holds information provided by the previously found superior solution, and that will be updated at the end of the iterations. At any iteration level, two steps essentially executes, (1) generation of new N solutions by artificial ants using the altered pheromone trail information available from previous iterations and (2) updating pheromone trail matrix. These steps were carried out repeatedly by the algorithm for a maximum number of given iterations, and a solution containing the best function value denotes the optimal threshold values.

V. Cuckoo Search Algorithm

Cuckoo search algorithm was proposed by Yang and deb [21], [22]. This algorithm is based on the brood parasitism of some cuckoo species. The CS algorithm is improved by means of levy flights, rather than using simple isotropic random walks. This algorithm was motivated by the aggressive reproduction strategy of certain cuckoo species, for instance, Ani and Guira cuckoos. These kinds of cuckoos lay their eggs in communal nests, although they might remove others' eggs to increase the probability of hatching of their own eggs. Only a small number of species involve brood parasitism by the way of laying their eggs in the nests of other host birds.

A. Cuckoo Search Algorithm Concept

The three idealized rules of a standard cuckoo search algorithm are:

- Every single cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nest with high quality of eggs will run over to the succeeding generation.
- The number of availability of host nest is fixed, and the egg laid by a cuckoo is discovered by the host bird with probability p$\propto \in[0,1]$.

For the problem of maximization or minimization, the fitness function is taken as the objective function itself. For generating new solution, $x^{(t+1)}$ for cuckoo I, a levy flight is performed as (7):

$$X_i^{(t+1)} = x_i^t + \propto \oplus \, levy(\lambda) \tag{7}$$

where, α is the step size ($\alpha > 1$) and is associated with the size of the problem. In most cases, α = 1 is used. The product $\oplus$ deals with entry wise multiplication processes. However, levy flights provide a random walk, however, their random step lengths are drawn from a levy distribution for large steps well-defined by (8):

$$Levy \sim u = t^{-\lambda}, Where\ 1 < \lambda \leq 3 \tag{8}$$

According to the application, the levy function can be changed. One of the levy functions is Mantegna's algorithm, which has an infinite variance and infinite mean.

B. Levy Flight Distribution

In this levy flight distribution, the animals and birds search for food in a random manner and mainly follow a random walk because of the next step which is based on the current place and the transition probability of next states. This kind of behavior can be modeled mathematically. The Levy flight distribution can be expressed using its Fourier transform as follows

$$I(s) = \exp[-\lambda|s|^{\mu}], 0 < \mu \leq 2 \tag{9}$$

where, λ is a scaling parameter. Except for the parameter of special case, the inverse transform does not have explicit analytical formulae. In case of μ = 2, the equations can be changed as

$$I(s) = \exp[-\lambda s^2] \tag{10}$$

The inverse transform of the above equation provides Gaussian distribution. The inverse integral becomes:

$$M(s) = \frac{1}{\pi}\int_0^{\infty} \mathrm{Cos}(ns)\ \exp[-\lambda|s|^{\mu}]\ Dn \tag{11}$$

This can be calculated for the greater value of s as:

$$M(s) = \frac{\lambda\mu\Gamma(\mu)\,\mathrm{Sin}\left(\frac{\pi\mu}{2}\right)}{\pi|s|^{1+\mu}}\ When\ s \to \infty \tag{12}$$

$$\Gamma(y) = \int_0^{\infty} f^{y-1}e^{-f}df \tag{13}$$

where, Γ(y) represents the gamma function.

C. Explanation of Efficiency Levy Flight Distribution

As compared to the Brownian random walks, the levy flights are recognized to be more efficient in exploring large scale search space. The motive behind stating the above phenomenon is owing to the fact that the variance of levy flights is

$$\tau^2 \sim f^{3-\mu}, 1 \leq \mu \leq 2 \tag{14}$$

which enlarges at a greater rate than the linear relationship of Brownian random walks followed by $\tau^2(f) \to f$.

D. Implementation of Levy Flight Distribution

The generation of levy flight distribution numbers can be attained by following two steps:

1. The choice of the random direction.
2. The generation of steps should obey the selected levy distribution.

The implementation can be done in many ways, but one of the easiest ways is the Mantegna algorithm for a symmetric levy stable distribution. For Mantegna algorithm, step length p is calculated by:

$$P = \frac{w}{|\delta|^{1/\mu}} \tag{15}$$

Here, y and z are derived from normal distributions.

$$W \sim Q(0, \tau_w^2)$$
$$S \sim Q(0, \tau_s^2) \tag{16}$$

$$\sigma_u = \left\{\frac{\Gamma(1+\mu)\sin(\pi\mu/2)}{\Gamma\left[\frac{1+\mu}{2}\right]\mu 2^{\frac{\mu-1}{2}}}\right\}^{1/\mu} \ \&\ \tau_s = 1 \tag{17}$$

These settings follow the levy distribution for $|p| \geq |p_0|$, where, p_0 is the smallest step and it is chosen to be anywhere between the value 0.1 – 1.

The CS algorithm controls the boundary conditions in each and every computation step. Accordingly, when the value of an attribute overflows the allowable search space limits, then the value of the associated attribute is updated with the value of a nearer limit value for the associated attribute. The Cuckoo search algorithm appears to be very effective over other optimization techniques as it includes only few parameters for tuning.

TABLE I
SURVEY OF NATURE INSPIRED ALGORITHM IN IMAGE PROCESSING DOMAIN

Algorithm	Nature inspiration	Image	References	Image processing domain	Performance
Particle swarm optimization	Swarming behavior	Natural image	[23]	Image compression	Image block, image quality is preserved.
Particle swarm optimization	Social behavior	Natural image	[24]	Image matching	Provide a more effective way for image matching in applications.
Particle swarm optimization	Social behavior	Human face image	[25]	Face recognition	Pso can efficiently find the optimal solution in large search space.
Particle swarm optimization	Social behavior	Random images	[26]	Feature selection	Pso has the ability to quickly converge; it has a strong search capability in the problem space and can efficiently find minimal reducts.
Particle swarm optimization	Social behavior	Natural image	[27]	Image interpolation	The overall visual quality of the interpolated image is improved.
Artificial bee colony optimization	Artificial forager bees (agents) search for rich artificial food sources	Gray scale images	[28]	Image deblurring	Reduced the nn model by improving its training via a new powerful optimization algorithm based on artificial bee colony.
Artificial bee colony optimization	Self-organizing Behavior	Natural image	[29]	Image fusion	Abc algorithm is exploited to carry out the structural and parametric optimization of the model.
Artificial bee colony optimization	The behavior of the real bees in finding the food source	Random images	[30]	Face recognition	Used for solving face recognition problems.
Ant colony optimization	Behavior of ant in finding food source	Random images	[31]	Blind noisy image quality evaluation	It is able to achieve a consistent image quality evaluation performance.
Ant colony optimization	Social behavior of ants	Random image	[32]	Image feature selection	Can obtain high classification accuracy.
Cuckoo search algorithm	The brood parasitic behavior of Some cuckoo species	Satellite images	[33]	Image contrast And brightness enhancement	Can obtain better enhancement results.

VI. CONCLUSION

In conclusion, four nature-inspired metaheuristic algorithms were reviewed: particle swarm optimization (PSO), Ant colony optimization (ACO), artificial bee colony optimization (ABC) and Cuckoo search algorithm (CS). The ACO and ABC have been inspired by the social behavior within ants' and bees' food foraging process respectively.

REFERENCES

[1] H.F. Ng, "Automatic thresholding for defect detection", pattern Recognition Letters, Volume 27, Issue 14, pp.1644-1649,2006.

[2] Chen Y.L, "Night time Vehicle Light Detection on a Moving Vehicle using Image segmentation and Analysis Techniques", WSEAS transactions on computers, Volume 8, Issue 3, pp. 506-515,2009.

[3] C.C.Chang, L.L.Wang, "A fast multilevel thresholding method based on lowpass and highpass filtering", Pattern Recogn. Lett. 18 , 1469-1478,1997.

[4] X.S. Yang, "Nature-inspired metaheuristic algorithms", Luniver press, 2008.

[5] Kanika Malik, Akash Tayal, "Comparision of Nature Inspired Metaheuristic Algorithms", International Journal of Electronic and Electrical Engineering, Volume 7, Number 8, pp. 799-802, 2014.

[6] D.Oliva, E. Cuevas, et al., "Multilevel Thresholding segmentation based on harmony search optimization", J.Appl.Math, 1-24,2013.

[7] L. Cao, P.Bao,Z.Shi, "The strongest schema learning GA and its application to multilevel Thresholding", Image Vision Comput. 26 , 716-724, 2008.

[8] W.B. Tao, J.W.Tian, J.Liu, "Image segmentation by three-level Thresholding based on maximum fuzzy entropy and genetic algorithm", Pattern Recogn. Lett. 24,3069-3078, 2003.

[9] B.Akay," A study on particle swarm optimization and artificial bee colony algorithms for multilevel Thresholding", Appl. Soft Comput. 13 ,3066-3091, 2013.

[10] M.Maitra, A.Chatterjee, "A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel Thresholding", Expert Syst. Appl. 34, 1341-1350, 2008.

[11] P. D. Sathya, R.Kayalvizhi, "Optimal multilevel Thresholding using bacterial foraging algorithm", Expert Syst. Appl. 38, 15549-15564, 2011.

[12] M. H. Horng, "Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization", Expert Systems with Applications 37, 4580-4592, 2010.

[13] K.Hammouche, M.Diaf,P.Siarry, "A comparative study of various metaheuristic techniques applied to the multilevel Thresholding problem", Eng. Appl. Artif. Intell. 23, 676-688, 2010.

[14] Kennedy, J.,rhart, R., "Particle swarm optimization", In: Proceedings of the IEEE International Conference on Neural Networks (ICNN'95), vol. IV, Perth, Australia, pp. 1942–1948, 1985.

[15] D.Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department , 2005.

[16] D.Karaboa, B.Batrk, "A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony(ABC) algorithm", J.Global Optimiz. 39, 459-471, 2007.

[17] D.Karaboa, B.Batrk, "On the performance of Artificial Bee Colony(ABC) algorithm", Appl. Soft Comput. 8, 687-697, 2008.

[18] P.Civicioglu, E.Besdok, "A conceptual comparision of the cuckoo search, particle swarm optimization, differential evolution and artificial bee colony algorithms", Artif. Intell. Rev. 39, 315-346, 2013.

[19] Dorigo, M., Gambardella, L.M., "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE transactions on Evolutionary Computation 1 (1), 53-66, 1997.

[20] Dorigo, M., Stutzle, T., "The ant colony optimization metaheuristic: algorithms", applications and advances. Technical Report IRIDIA-2000-32, 2000.

[21] Xin-She Yang, Suash Deb, "Engineering Optimisation by Cuckoo Search",arxiv:1005.2908v3 [math.OC]; 2010.

[22] Yang, X. S., Deb. S., "Cuckoo search via levy flights", In: Proc. Of World Congress on Nature & Biologically Inspired Computing, pp. 210-214, 2009.

[23] Chun-Chieh Tseng, Jer-Guang Hsieh, Jyh-Horng Jeng, "Fractal image compression using visual based particle swarm optimization", Image and Vision Computing 26, 1154–1162, 2008.

[24] Fang Liu, Haibin Duan, Yimin Deng, "A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching", Optik 123,1955– 1960, 2012.

[25] Jin wei, Zhang jian-qi, Zhang Xiang, "Face recognition method based on support vector machine and particle swarm optimization", Expert Systems with Applications 38, 4390–4393, 2012.

[26] Xiangyang Wang, Jie Yang, Xialong Teng, Weiijun Xia, Richard Jensen, "Feature selection based on rough sets and particle swarm optimization", Pattern Recognition Letters 28, 459–471, 2007.

[27] suan – Ying Chen, Jin – Jang Leou, "Saliency-directed image interpolation using particle swarm optimization", Signal Processing 90, 1676–1692, 2010.

[28] Slami saadi, Abderrezak Guessoum, Maamar Bettayeb, "ABC optimized neural network model for image deblurring with its FPGA implementation", Microprocessors and Microsystems 37, 52–64, 2013.

[29] Jiaqian Yu, Haibin Duan, "Artificial Bee Colony approach to information granulation-based fuzzy radial basis function neural networks for image fusion", Optik 124, 3103– 3111, 2013.

[30] Ankush Chakrabarty, Harsh Jain, Amitava Chaterjee, "Volterra kernel based face recognition using artificial bee colony optimization", Engineering Applications of Artificial Intelligence 26, 1107–1114, 2013.

[31] Li Chen, Xiaotong Huang, Jing Tian, Xiaowei Fu, "Blind noisy image quality evaluation using a deformable ant colony algorithm", Optics & Laser Technology 57, 265–270, 2014.

[32] Bolun Chen, Ling Chen, Yixin Chen, "Efficient ant colony optimization for image feature selection", Signal Processing 93, 1566–1576, 2013.

[33] A.K.Bhandari, V.Soni, A.Kumar, G.K.Singh, "Cuckoo search algorithm based satellite image contrast and brightness enhancement using DWT–SVD", ISA Transactions 53 , 1286–1296, 2014.