CrossMark

# Image segmentation by minimum cross entropy using evolutionary methods

**Diego Oliva[1]** · **Salvador Hinojosa[2]** · **Valentín Osuna-Enciso[3]** · **Erik Cuevas[1]** ·
**Marco Pérez-Cisneros[1]** · **Gildardo Sanchez-Ante[4]**

**Abstract** The segmentation of digital images is one of the most important steps in an image processing or computer vision system. It helps to classify the pixels in different regions according to their intensity level. Several segmentation techniques have been proposed, and some of them use complex operators. The techniques based on thresholding are the easiest to implement; the problem is to select correctly the best threshold that divides the pixels. An interesting method to choose the best thresholds is the minimum cross entropy (MCET), which provides excellent results for bi-level thresholding. Nevertheless, the extension of the segmentation problem into multiple thresholds increases significantly the computational effort required to find optimal threshold values. Each new threshold adds complexity to the formulation of the problem. Classic methods for image thresholding perform extensive searches, while new approaches take advantage of heuristics to reduce the search. Evolutionary algorithms use heuristics to optimize criteria over a finite number of iterations. The correct selection of an evolutionary algorithm to minimize the MCET directly impacts the performance of the method. Current approaches take a large number of iterations to converge and a high rate of MCET function evaluations. The electromagnetism-like optimization (EMO) algorithm is an evolutionary technique which emulates the attraction–repulsion mechanism among charges for evolving the individuals of a population. Such technique requires only a small number of evaluations to find the optimum. This paper proposes the use of EMO to search for optimal threshold values by minimizing the cross entropy function while reducing the amount of iterations and function evaluations. The approach is tested on a set of benchmark images to demonstrate that is able to improve the convergence and velocity; additionally, it is compared with similar state-of-the-art optimization approaches.

**Keywords** Image processing · Segmentation · Evolutionary algorithms · Cross entropy · Electromagnetism optimization

✉ Diego Oliva
diego.oliva@cucei.udg.mx

Salvador Hinojosa
salvahin@ucm.es

Valentín Osuna-Enciso
valentin.osuna@cutonala.udg.mx

Erik Cuevas
erik.cuevas@cucei.udg.mx

Marco Pérez-Cisneros
marco.perez@cucei.udg.mx

Gildardo Sanchez-Ante
gildardo.sanchez@upy.edu.mx

[1] División de Electrónica y Computación, CUCEI, Universidad de Guadalajara, Av. Revolución 1500, Guadalajara, Jalisco, Mexico

[2] Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense de Madrid, Av. Complutense S/N, 28040 Madrid, Spain

[3] Departamento de Ciencias de la Información y Desarrollos Tecnológicos, CUTONALA, Universidad de Guadalajara, Av. Nuevo Periférico No. 555, Tonalá, Jalisco, Mexico

[4] Universidad Politécnica de Yucatán, Km. 4.5. Carretera Mérida - Tetiz, Ucú, Yucatán, Mexico

## 1 Introduction

Many computer vision systems highly rely on the quality of the input image to provide useful outputs. As a result, many researches have been made on enhancing the input images to avoid undesired features such as noise, visual artifacts

and reducing redundancy on the information. Segmentation is one of the most commonly used techniques to address such issues; it separates the pixels of the image into different classes based on the intensity level of each pixel. Segmentation has been used to extract features of digital images (Kong et al. 2015), for object identification (Cao et al. 2014) and surveillance (Bhandari et al. 2014). Thresholding is the easiest and simplest method for image segmentation; it takes the information of the histogram from the analyzed image and determinates a threshold (*th*) value to separate the pixels in different regions. Two approaches are used to segment an image using thresholding: bi-level thresholding (BT) and multilevel thresholding (MT). BT requires a single value *th* to generate two classes (e.g., foreground and background). Likewise, MT uses a finite number of threshold values to divide the image into more than two homogeneous classes (Gonzalez and Woods 1992; Akay 2013).

Thresholding-based techniques are divided in parametric and nonparametric (Otsu 1979; Kapur et al. 1985; Horng 2010; Olugbara et al. 2015). A parametric approach considers that each class, or group of pixels in an image, can be modeled using probability density functions and that the mixture of those classes will represent all the pixels in the image. In other words, the parametric methods approximate the histogram of an image using different mechanisms and assuming that it has a Gaussian distribution (Horng 2010). On the other hand, nonparametric techniques use discriminative criteria to separate the pixels into homogenous regions (Horng 2010). Such criteria are metrics that verify the quality of a *th* value and are also used as objective function since they result as an attractive option due to their robustness and accuracy. Thresholding segmentation has also become popular due to its simple concept. Contrary to clustering approaches, thresholding only requires the number of thresholds to be set a priori, while most classical cluster-based techniques for MT also require the centroid of each class and extra information to be correctly initialized (Olugbara et al. 2015).

On the literature, there are two classical nonparametric methods for bi-level thresholding: the first one was proposed by Otsu (1979) and maximizes the variance between classes. The second one was proposed by Kapur et al. (1985), and it proposes the maximization of entropy as a measure of the homogeneity among classes. The two methods are extensively used in image processing, and both of them have proved to be efficient and accurate alternatives to segment pixels into two classes (Sathya and Kayalvizhi 2011). Both methods can be extended for multilevel thresholding; however, their computational complexity increases, while their accuracy decreases with each new threshold added into the searching process (Sathya and Kayalvizhi 2011; Agrawal et al. 2013). The importance of thresholding techniques is evidenced by the large amount of research made in recent years (Bhandari et al. 2014; Liu et al. 2015; Khairuzzaman and Chaudhury 2017).

The minimum cross entropy (MCET) was originally proposed by Kullback (1968) and is also known as direct divergence. The MCET is a distance metric between two probabilistic distributions, proposed as an extension of the maximum entropy principle. It has been used to select correctly the best threshold, minimizing the cross entropy between the original image and the segmented results (Li and Lee 1993). An improved version of the MCET was presented by Yin (2007) which uses a recursive programming to reduce the computational effort of computing the MCET for multilevel thresholding. Different to the Otsu and Kapur methods, the MCET produces a functional formulation whose accuracy does not depend on the number of threshold points (Horng and Liou 2011). Although the MCET is an efficient method and it gives excellent results for bi-level thresholding, its performance is affected for multilevel thresholding. The complexity increases with each threshold that is added, generating both restrictions as well as multimodality in the solutions, even for the recursive programming approach. The problem in MCET is to find the appropriate threshold values without affecting the computational cost. In order to overcome this problem, different evolutionary techniques are applied as a search mechanism using as objective function the MCET, generating interesting segmentation approaches that have been reported in the related literature. For example, the use of the popular particle swarm optimization (Kennedy and Eberhart 1995) to minimize the MCET is proposed by Yin (2007). Sarkar presented an algorithm based on differential evolution (Sarkar et al. 2011). Horng presented two interesting segmentation mechanisms, one of them based on the Honey Bee Mating algorithm (Horng 2010) and the second one uses the Firefly algorithm (FF) (Horng and Liou 2011). All these approaches optimize the MCET objective function despite of its high multimodality characteristics. However, to obtain the optimal values, they usually require a large number of iterations and evaluations of the objective function, even if they use the recursive programming techniques. Moreover, their results are suboptimal because they have a lack of accuracy in the optimization process; for instance, the accuracy of the FF depends directly on the control parameters (Yang 2014). Recently, MCET has proven to be an interesting tool in important fields such as medical image processing by performing brain image and tumor segmentation (Kaur et al. 2016; Oliva et al. 2017). However, the use of a high number of iterations and evaluations to produce satisfying results is still present.

The aim of this paper is to present the use of the electromagnetism-like algorithm (EMO) for multilevel thresholding to reduce the amount of iterations and function evaluations. EMO is an evolutionary computation technique that uses a population of charged particles to find the optimal

solution. It was introduced by Birbil and Fang (2003) to solve unconstrained optimization problems. The analogy of EMO is the electromagnetism, in specific the attraction-repulsion of charged particles within an electromagnetic field. Under the EMO context, the particles are candidate solutions, and they contain an amount of charge that depends on the objective function. Such particles have a position in the search space and in each iteration they are moved to new position considering the force exerted among them. Different to other evolutionary methods, EMO exhibits interesting search capabilities such as fast convergence still keeping its ability to avoid local minima in high modality environments (De Jong 1988; Dorigo et al. 1996; De Castro and Von Zuben 2002). Recent works in this field (Birbil et al. 2004; Wu et al. 2004; Rocha and Fernandes 2009a, b) demonstrate that the EMO algorithm presents the best balance between optimization results and demand of function evaluations. Considering such features, EMO has been effectively employed to solve problems from different fields of engineering. Some examples of the areas where EMO has been implemented are: flow-shop scheduling (Naderi et al. 2010), communications (Hung and Huang 2011), vehicle routing (Yurtkuran and Emel 2010), array pattern optimization in circuits (Jhang and Lee 2009), neural network training (Lee and Chang 2010), image processing and control systems (Ghamisi et al. 2012). For that reason, this technique is used to find the best threshold values, by minimizing the cross entropy. Such approach only takes as input the histogram and the number of thresholds to be found. In this sense, the size of the image does not affect the quality and accuracy of the results and doesn't require any additional information or initialization about the problem. As a result, the proposed algorithm substantially reduces the number of function evaluations preserving the excellent search capabilities of an evolutionary method. The segmentation algorithm encodes as a particle the set of candidate threshold points. The MCET objective function evaluates the quality of the candidate particle. Guided by the values of this objective function, the set of encoded candidate solutions are modified using the operators present on EMO so that they can improve their segmentation quality as the optimization process evolves. In comparison to similar approaches, the proposed method deploys better segmentation results yet consuming less MCET function evaluations, and it is reflected on the lesser computational effort.

The remainder sections of the paper is organized as follows. In Sect. 2, the standard EMO algorithm is presented. Section 3 gives a simple description of the minimum cross entropy method. Section 4 explains the implementation of the proposed algorithm. Section 5 discusses experimental results and comparisons after testing the proposal over a set of benchmark images. Finally, in Sect. 6 the conclusions are discussed.

## 2 Electromagnetism-Like Optimization Algorithm (EMO)

Birbil and Fang proposed the EMO algorithm (Birbil and Fang 2003), a population-based evolutionary method created to solve unconstrained optimization problems. EMO is able to converge fast to the optimal values avoiding the local values that are commonly found in multimodal search spaces (De Jong 1988; Dorigo et al. 1996; De Castro and Von Zuben 2002). Another important feature of EMO is a good balance between optimization results and function evaluations (Birbil et al. 2004; Wu et al. 2004; Rocha and Fernandes 2009a, b). EMO has a population $\mathbf{Sp}_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ of $N$ particles $(x_{i,t})$ with an $n-$dimensional size $(i = 1, 2, \ldots, n)$. The population is used to search for a feasible set $\mathbf{X} = \{x \in \Re^n \, | l_i \leq x \leq u_i\}$ where $t$ represents the iteration (or generation) of the algorithm. Prior to the optimization process, the population $\mathbf{Sp}_t$ is initialized being $t = 1$ taking uniformly distributed samples from the search region $\mathbf{X}$. After the initialization of $\mathbf{Sp}_t$, the EMO iterative process continues until a stopping condition (e.g., the maximum number of iterations) is met. An iteration of EMO consists of two main steps: in the first step, each particle in $\mathbf{Sp}_t$ moves to a different location by using the attraction–repulsion mechanism of the electromagnetism theory (Birbil et al. 2004). In the second step, the particles moved by the electromagnetism principle are further perturbed locally by a local search and then become members of $\mathbf{Sp}_{t+1}$ in the next iteration. Both, the attraction–repulsion mechanism and the local search in EMO are responsible for driving the particles of $\mathbf{Sp}_t$ near to the global optimum.

Similar to the electromagnetism theory for charged particles, every particle $x_{i,t} \in \mathbf{Sp}_t$ in the search space $\mathbf{X}$ is considered as a charged particle where the charge is computed based on its objective function value. Points with better objective function value have higher charges than remaining particles. The attraction–repulsion mechanism is a process in EMO by which particles with more charge attract other particles from $\mathbf{Sp}_t$, and particles with less charge repel other elements. Finally, a total force vector $F_i^t$, exerted on a point (e.g., the $i$-th point $x_{i,t}$) is calculated by the sum of these attraction–repulsion forces, and each $x_{i,t} \in \mathbf{Sp}_t$ is moved in the direction of its total force to the location $y_{i,t}$. A local search is used to explore the vicinity of the each particle according to its fitness. The members of the new population, $x_{i,t+1} \in \mathbf{Sp}_{t+1}$ are determined by using the following update equation:

$$x_{i,t+1} = \begin{cases} y_{i,t} & \text{if} \quad f(y_{i,t}) \leq f(z_{i,t}) \\ z_{i,t} & \text{otherwise} \end{cases} \tag{1}$$

In Eq. (1), $z_{i,t}$ is the particle perturbed in the local search procedure. Both $y_{i,t}$ and $z_{i,t}$ are evaluated using the objective

function $f(\cdot)$. This equation updates of an element of the population if the value of the objective function is lower. Notice that the sign $\leq$ is used for minimization and it could be changed for maximization problems.

Algorithm 1 shows the general scheme of EMO, where each step is accordingly described.

The force that exists between two points $x_{i,t}$ and $x_{j,t}$ of the population is obtained using Eq. (5).

$$F_{i,j}^t = \begin{cases} (x_{j,t} - x_{i,t}) \frac{q_{i,t} \cdot q_{j,t}}{\|x_{j,t} - x_{i,t}\|^2} & \text{if } f(x_{i,t}) > f(x_{j,t}) \\ (x_{i,t} - x_{j,t}) \frac{q_{i,t} \cdot q_{j,t}}{\|x_{j,t} - x_{i,t}\|^2} & \text{if } f(x_{i,t}) \leq f(x_{j,t}) \end{cases}$$

(5)

---

**Algorithm 1** [EMO ( $N, Iter_{max}, Iter_{local}, \delta$ )]

1.      Input parameters: Maximum number of iteration $Iter_{max}$, values for the local search parameter such $Iter_{local}$ and, and the size $N$ of the population.
2.      Initialize: set the iteration counter $t=1$, initialize the number of $\mathbf{Sp}_t$ uniformly in $\mathbf{X}$ and identify the best point in $\mathbf{Sp}_t$.
3.      while $t < Iter_{max}$ do
4.          $F_i^t \leftarrow \text{CalcF}(\mathbf{Sp}_t)$
5.          $y_{i,t} \leftarrow \text{Move}(x_{i,t}, F_i^t)$
6.          $z_{i,t} \leftarrow \text{Local}(Iter_{local}, \delta, y_{i,t})$
7.          $x_{i,t+1} \leftarrow \text{Select}(\mathbf{Sp}_{t+1}, y_{i,t}, z_{i,t})$
8.      end while

---

*Input parameters (Line 1)* EMO algorithm runs for $Iter_{max}$ iterations. In the local search phase, $n \times Iter_{local}$ is the maximum number of locations $z_{i,t}$, within a $\delta$ distance of $y_{i,t}$, for each $i$ dimension.

*Initialize (Line 2)* The points $x_{i,t}$, $t = 1$, are selected uniformly in $\mathbf{X}$, i.e., $x_{i,1} \approx \text{Unif}(\mathbf{X})$, $i = 1, 2, \ldots, N$, where Unif represents the uniform distribution in $\mathbf{X}$. The objective function values $f(x_{i,t})$ are computed, and the best point is identified for minimization in Eq. (2) and for maximization in Eq. (3).

$$x_t^B = \arg \min_{x_{i,t} \in \mathbf{Sp}_t} \{f(x_{i,t})\}$$

(2)

$$x_t^B = \arg \max_{x_{i,t} \in \mathbf{Sp}_t} \{f(x_{i,t})\}$$

(3)

From Eqs. (2) and (3) the best value $x_t^B$ is selected from the current population in the $t$ iteration.

*Calculate force (Line 4)* In this step, a charged-like value ($q_{i,t}$) is assigned to each point ($x_{i,t}$). The charge $q_{i,t}$ of $x_{i,t}$ depends on $f(x_{i,t})$ and points with better objective function have more charge than others. The charges are computed according to Eq. (4).

$$q_{i,t} = \exp\left(-n \frac{f(x_{i,t}) - f(x_t^B)}{\sum_{j=1}^N f(x_{j,t}) - f(x_t^B)}\right)$$

(4)

From Eq. (4), $n$ is the number of dimensions of the search space, exp is the exponential function, $N$ is the number of candidate solutions and $x_t^B$ is the best point of the population at the iteration $t$.

The total force, $F_i^t$, corresponding to a specific particle $x_{i,t}$ is now calculated as follow:

$$F_i^t = \sum_{j=1, j\neq i}^N F_{i,j}^t$$

(6)

Once the forces are computed, the next step (Line 5) is to move the selected point $x_{i,t}$ along the direction of the vector $F_i^t$ using the next equation:

$$y_{i,t} = x_{i,t} + \lambda \frac{F_i^t}{\|F_i^t\|}(\text{RNG}), \quad i = 1, 2, \ldots, N; i \neq B$$

(7)

where $\lambda \approx \text{Unif}(0, 1)$ is a uniformly distributed random number in the interval [0,1] for each coordinate of $x_{i,t}$, and RNG denotes the allowed range of movement toward the lower or upper bound for the corresponding dimension. $B$, refers to the index of the best element of the population.

*Local search (Line 6)* For each $y_{i,t}$ a maximum of $iter_{local}$ points are generated in each coordinate direction in the $\delta$ neighborhood of $y_{i,t}$. This means that the process of generating local point is continued for each $y_{i,t}$ until either a better $z_{i,t}$ is found or the $n \times Iter_{local}$ trial is reached.

*Selection for the next iteration (Line 7)* In this step, $x_{i,t+1} \in \mathbf{Sp}_{t+1}$ are selected between $y_{i,t}$ and $z_{i,t}$ using Eq. (1), and the best point is identified using Eq. (2) for minimization or Eq. (3) for maximization.

The optimization process performed by EMO involves different operations that are able to manage local and global information of the search space. This process is more complex than other evolutionary approaches which only employ

one (or two) mathematical operations to modify the population.

## 3 Minimum cross entropy (MCET)

The cross entropy also known as divergence is an information theoretic metric used to verify the amount of information in a random process (Tang et al. 2011). In other words, it helps to measure the distance between two probabilistic distributions (Pal 1996). The cross entropy introduced by Kullback (1968) is in general defined as:

$$D\left(\mathbf{B}, \mathbf{C}\right) = \sum_{k=1}^{N} b_k \log\left(\frac{b_k}{c_k}\right) \tag{8}$$

where $\mathbf{B} = \{b_1, b_2, \ldots, b_N\}$ and $\mathbf{C} = \{c_1, c_2, \ldots, c_N\}$ are two probabilistic distributions of the same set. $D$ is the minimum cross entropy that is an extension of the maximum cross entropy. The MCET could be interpreted as the expectation of change in the information content in when is used $\mathbf{B}$ instead of $\mathbf{C}$ (Li and Lee 1993). It is important to mention that a higher value of MCET represents more uncertainty in the random process.

In image processing, and especially for image segmentation, a set of different thresholds ($\mathbf{th} = [th_1, th_2, \ldots th_{nt}]$) is selected from the image histogram ($h$). The values contained in $\mathbf{th}$ must minimize the MCET that exists between the original image $\mathbf{I}_{or}$ and the thresholded image $\mathbf{I}_{th}$. The simplest example of image segmentation is using one threshold ($\mathbf{th} = [th_1]$), once it was selected the $\mathbf{I}_{th}$ is generated according to the following rule:

$$\mathbf{I}_{th}\left(i, j\right) = \begin{cases} \mu\left(1, th_1\right) & \text{if } \mathbf{I}_{or}\left(i, j\right) < th_1, \\ \mu\left(th_1, L+1\right) & \text{if } \mathbf{I}_{or}\left(i, j\right) \geq th_1 \end{cases} \tag{9}$$

where $L$ is the higher intensity value of the histogram ($L = 255$ for an 8 bits gray scale image). The rule presented in Eq. (9) could be easily expanded for more than one $th$. Considering the histogram ($h$) of the original image, the normalized value $\mu$ for a specific range restricted by $a$ and $b$ is then computed as:

$$\mu\left(a, b\right) = \frac{\sum_{i=a}^{b-1} ih\left(i\right)}{\sum_{i=a}^{b-1} h\left(i\right)}, \quad i = 1, 2, \ldots, L \tag{10}$$

In order to verify if the original image was correctly segmented is necessary to apply the minimum cross entropy for digital images. The MCET is then computed using the method proposed by Li and Lee (1993) that is defined for one threshold ($\mathbf{th} = [th_1]$) as follows:

$$D\left(\mathbf{th}\right) = \sum_{i=1}^{th_1-1} ih\left(i\right) \log\left(\frac{i}{\mu\left(1, th_1\right)}\right)$$
$$+ \sum_{i=th_1}^{L} ih\left(i\right) \log\left(\frac{i}{\mu\left(th_1, L+1\right)}\right) \tag{11}$$

The aim is to find the best set of thresholds that minimizes the cross entropy. To achieve this, Eq. (12) is defined as the objective function.

$$\mathbf{th}_{opt} = \arg\min_{\mathbf{th}}\left(D\left(\mathbf{th}\right)\right) \tag{12}$$

The computational complexity to obtain an optimal threshold ($\mathbf{th} = [th_1]$) is $O\left(nt \cdot L^2\right)$ but it becomes computationally expensive for multilevel thresholding approaches, for $nt$ thresholds values the complexity is $O\left(nt \cdot L^{nt+1}\right)$ (Tang et al. 2011).

### 3.1 Recursive MCET

As is mentioned above, the MCET is computationally expensive for multilevel thresholding. In order to reduce the computational effort, there is a faster recursive programming technique proposed to obtain the optimal thresholds for a digital image (Yin 2007).

In the recursive programming MCET Eq. (11) could be rewritten as (Yin 2007; Hammouche et al. 2008):

$$D\left(\mathbf{th}\right) = \underbrace{\sum_{i=1}^{L} ih\left(i\right) \log\left(i\right)}_{\text{Image Entropy}} - \underbrace{\sum_{i=1}^{th_1-1} ih\left(i\right) \log\left(\mu\left(1, th_1\right)\right)}_{\text{Entropy from 1 to } th_1-1}$$
$$- \underbrace{\sum_{i=th_1}^{L} ih\left(i\right) \log\left(\mu\left(th_1, L+1\right)\right)}_{\text{Entropy from } th_1 \text{ to } L+1} \tag{13}$$

Eq. (13) is defined for only one $th$ and two classes. The first term is constant for a given digital image, and the remainder elements depend directly on the selected thresholds. Considering such facts, the objective function could be rewritten as:

$$\varphi\left(\mathbf{th}\right) = -\sum_{i=1}^{th_1-1} ih\left(i\right) \log\left(\mu\left(1, th_1\right)\right)$$
$$- \sum_{i=th_1}^{L} ih\left(i\right) \log\left(\mu\left(th_1, L+1\right)\right)$$
$$= -\left(\sum_{i=1}^{th_1-1} ih\left(i\right)\right) \log\left(\frac{\sum_{i=1}^{th_1-1} ih\left(i\right)}{\sum_{i=1}^{th_1-1} h\left(i\right)}\right)$$

$$-\left(\sum_{i=th_1}^{L} ih(i)\right)\log\left(\frac{\sum_{i=th_1}^{L} ih(i)}{\sum_{i=th_1}^{L} h(i)}\right)$$

$$= -m^1(1, th_1)\log\left(\frac{m^1(1, th_1)}{m^0(1, th_1)}\right)$$

$$-m^1(th_1, L+1)\log\left(\frac{m^1(th_1, L+1)}{m^0(th_1, L+1)}\right) \quad (14)$$

here for a partial range of the image histogram, the values of the zero-moment point $m^0(a, b) = \sum_{i=a}^{b-1} h(i)$ and the first-moment point $m^1(a, b) = \sum_{i=a}^{b-1} ih(i)$ are computed. Eq. (14) can be extended for multilevel thresholding, considering a set of thresholds denoted by $\mathbf{th} = [th_1, th_2, \ldots th_{nt}]$, where $nt$ is the number of thresholds to be found. For convenience two additional dummy thresholds $th_0 \equiv 0$ and $th_{nt+1} \equiv L + 1$ are added, it is important the order of such values: $th_0 < th_1 < \cdots th_{nt} < th_{nt+1}$. The recursive programming is applied to formulate the objective function for multilevel thresholding using the minimum cross entropy that is defined as:

$$\varphi(\mathbf{th}) = m^1(th_{i-1}, th_i)\log\left(\frac{m^1(th_{i-1}, th_i)}{m^0(th_{i-1}, th_i)}\right),$$
$$\mathbf{th} = [th_1, th_2, \ldots, th_{nt}], \quad i = 1, 2, \ldots, nt \quad (15)$$

With the use of Eq. (15) the complexity is reduced from $O\left(nt \cdot L^{nt+1}\right)$ to $O\left(nt \cdot L^{nt}\right)$ (Tang et al. 2011). However, it is still computationally expensive (Sarkar et al. 2015).

## 4 Multilevel thresholding using EMO and minimum cross entropy (MCET)

The computation of MCET for image thresholding is a complex task even if it is redefined using recursive programming. The main problem is to obtain the best thresholds that reduce the MCET; an exhaustive search requires a high computational effort. This paper proposes the use of the electromagnetism-like optimization (EMO) for multilevel segmentation based on the MCET. The approach uses the EMO algorithm to find the optimal $\mathbf{th}$ vectors by minimizing the complex objective function defined by the MCET. Compared with other evolutionary methods, EMO exhibits interesting search capabilities such as fast convergence while still keeping its ability to avoid local minima in high modality environments (De Jong 1988; Dorigo et al. 1996; De Castro and Von Zuben 2002). In the related literature, there are studies (Birbil et al. 2004; Wu et al. 2004; Rocha and Fernandes 2009a, b) that demonstrate that EMO provides one of the best trade-offs between the optimization performance and the demand for function of evaluations. The results showed that this algorithm substantially reduce the number of func-

tion evaluations while preserving its good search capabilities. Although EMO is a good alternative for global optimization, it includes a process to compute the elements of the new population that involve several operations described in Eqs. (1–7). This section describes the implementation of the proposed approach based on EMO and MCET.

### 4.1 Particle representation

Each particle uses $nt$ decision variables in the optimization algorithm. Such elements represent a different threshold value used for the segmentation. Therefore, the entire population is represented as:

$$\mathbf{Sp}_t = [\mathbf{th}_1, \mathbf{th}_2, \ldots, \mathbf{th}_N], \quad \mathbf{th}_i = [th_1, th_2, \ldots, th_{nt}]^T \quad (16)$$

where $t$ represents the iteration number, $T$ refers to the transpose operator, $N$ is the size of the population and $nt$ is the number of thresholds for each member of the population ($i = 1, 2, \ldots, nt$).

### 4.2 EMO implementation

The proposed segmentation algorithm has been implemented considered the minimum cross entropy as objective function (Eq. 15). The implementation of the EMO algorithm can be summarized into the following steps:

*Step 1* Read the image $I$ and store it as the gray scale image $I_{Gr}$.

*Step 2* Obtain histogram $h^{Gr}$ of $I_{Gr}$.

*Step 3* Initialize the EMO parameters: $Iter_{max}$, $Iter_{local}$, $\delta$, $k$ and $N$.

*Step 4* Initialize a population $\mathbf{Sp}_t$ of $N$ random particles with $nt$ dimensions.

*Step 5* Evaluate $\mathbf{Sp}_t$ in the recursive programming objective function $\varphi(\mathbf{Sp}_t)$, Eq. (15).

*Step 6* Compute the charge of each particle using Eq. (4), and with Eqs. (5) and (6) compute the total force vector.

*Step 7* Move the entire population $\mathbf{Sp}_t$ along the total force vector using Eq. (7).

*Step 8* Apply the local search to the moved population and select the best elements of this search based on their objective function values.

*Step 9* The $t$ index is increased in 1, If $t \geq Iter_{max}$ or if the stop criteria is satisfied the algorithm finishes the iteration process and jump to step 10. Otherwise, jump to step 5.

*Step 10* Select the particle that has the best $x_t^{B^c}$ objective function value (Eqs. (2) and (15)).

*Step 11* Apply the thresholds values contained in $x_t^{B^c}$ to the image $I_{Gr}$.

### 4.3 Multilevel thresholding

Once the EMO algorithm finds the best threshold values which maximize the objective function, the pixels of the image are segmented using such values. In this paper, we use the following rule for two levels and three classes:

$$I_s(r,c) = \begin{cases} I_{Gr}(r,c) & \text{if } I_{Gr}(r,c) \leq th_1 \\ th_1 & \text{if } th_1 < I_{Gr}(r,c) \leq th_2 \\ I_{Gr}(r,c) & \text{if } I_{Gr}(r,c) > th_2 \end{cases} \quad (17)$$

where $I_s(r,c)$ is the gray value of the segmented image, $I_{Gr}(r,c)$ is the gray value of the original image both in the pixel position $r, c$. $th_1$ and $th_2$ are the threshold values obtained by the EMO approach. Equation (17) can be easily extended to more than two levels using Eq. (18).

$$I_s(r,c) = \begin{cases} I_{Gr}(r,c) & \text{if } I_{Gr}(r,c) \leq th_1 \\ th_{i-1} & \text{if } th_{i-1} < I_{Gr}(r,c) \leq th_i \ , \ i = 2,3,\ldots nt-1 \\ I_{Gr}(r,c) & \text{if } I_{Gr}(r,c) > th_{nt} \end{cases}$$
$$(18)$$

## 5 Experimental results

In this paper, a set of benchmark images is used to test the proposed approach. This set contains 11 images with different complexity level; all the images have the same size ($512 \times 512$ pixels), and they are in JPGE format. Some of these images are widely used in the image processing literature to test different methods (Lena, Cameraman, Hunter, Baboon, etc.) (Agrawal et al. 2013).

The proposed multilevel thresholding algorithm based on EMO is compared with standard versions of the differential evolution (DE) (Storn and Price 1997), the particle swarm optimization (PSO) (Kennedy and Eberhart 1995), the harmony search (HS) [HS], social spider algorithm (SSA) [SSA] and the artificial bee colony (ABC) [ABC]. All those methods were programmed and tested on MATLAB 8.3 using a Xeon E5-2620 CPU @ 2.4Ghz with 16GB of RAM. Since the six algorithms are stochastic, statistical metrics are employed to verify the efficiency of the algorithms. For experimental purposes, all the algorithms are executed 35 times for each image. In accordance with the related literature, the number of thresholds is set to $nt = 2, 3, 4, 5$ (Horng 2010; Horng and Liou 2011). The stop criteria for each experiment are set to 50 iterations, once each test is performed the standard deviation (STD) is computed using Eq. (19). The STD represents the stability of the tested method, and if the STD increases, the algorithm becomes more instable (Ghamisi et al. 2012).

$$STD = \sqrt{\sum_{i=1}^{Iter_{max}} \frac{(\sigma_i - \mu)}{Ru}} \quad (19)$$

Once the best thresholds are obtained, the segmented images are generated using Eq. (18). However, it is necessary to verify the quality of the results. The peak-to-signal ratio (PSNR) is used to compare the similarity of an image (image segmented) against a reference (original image) based on the mean square error (MSE) of each pixel (Il-Seok Oh et al. 2004; Horng 2011; Akay 2013; Agrawal et al. 2013). Both PSNR and MSE are defined as:

$$PSNR = 20 \log_{10}\left(\frac{255}{RMSE}\right), \ (dB)$$
$$RMSE = \sqrt{\frac{\sum_{i=1}^{ro}\sum_{j=1}^{co}(I_{Gr}(i,j)-I_{th}(i,j))}{ro \times co}} \quad (20)$$

where $I_{Gr}$ is the original image, $I_{th}$ is the segmented image obtained after applying the best **th** and $ro, co$ are the total number of rows and columns of the image, respectively. Another interesting metric is the structure similarity index (SSIM) that is used to compare the structures of the original against the thresholded image (Wang et al. 2004). The SSIM method is defined in Eq. (21), and a higher value indicates a better segmentation performance.

$$SSIM(I_{Gr}, I_{th}) = \frac{(2\mu_{I_{Gr}}\mu_{I_{th}} + C1)(2\sigma_{I_{Gr}I_{th}} + C2)}{(\mu_{I_{Gr}}^2 + \mu_{I_{th}}^2 + C1)(\sigma_{I_{Gr}}^2 + \sigma_{I_{th}}^2 + C2)}$$

$$\sigma_{I_o I_{Gr}} = \frac{1}{N-1}\sum_{i=1}^{N}(I_{Gr_i} + \mu_{I_{Gr}})(I_{th_i} + \mu_{I_{th}})$$
$$(21)$$

From Eq. (21), $\mu_{I_{Gr}}$ and $\mu_{I_{th}}$ are the mean value of the original and the segmented image, respectively, and for each image the values of $\sigma_{I_{Gr}}$ and $\sigma_{I_{th}}$ correspond to the standard deviation. $C1$ and $C2$ are constants used to avoid the instability when $\mu_{I_{Gr}}^2 + \mu_{I_{th}}^2 \approx 0$, experimentally in (Agrawal et al. 2013) both values are $C1 = C2 = 0.065$. Another method used to measure the quality of the segmented image is the feature similarity index (FSIM) (Zhang et al. 2011). FSIM calculates the similarity between two images: in this case, the original gray scale image and the segmented image. As PSNR and SSIM, the higher value is interpreted as better performance of the thresholding method. The FSIM is then defined as:

$$FSIM = \frac{\sum_{w \in \Omega} S_L(w) PC_m(w)}{\sum_{w \in \Omega} PC_m(w)} \quad (22)$$

where $\Omega$ represents the entire domain of the image, the values of $S_L$ are then computed as:

$$S_L(w) = S_{PC}(w) S_G(w)$$

**Table 1** EMO parameters for the MCET

| $\text{Iter}_{max}$ | $\text{Iter}_{local}$ | $\delta$ | $N$ |
|---|---|---|---|
| 200 | 5 | 0.25 | 25 |

$$S_{PC}(w) = \frac{2PC_1(w) PC_2(w) + T_1}{PC_1^2(w) + PC_2^2(w) + T_1}$$

$$S_G(w) = \frac{2G_1(w) G_2(w) + T_2}{G_1^2(w) + G_2^2(w) + T_2} \tag{23}$$

From Eq. (23), $G$ is the gradient magnitude (GM) of a digital image and is defined as:

$$G = \sqrt{G_x^2 + G_y^2} \tag{24}$$

On Eq. (22), the value of PC is the phase congruence defined as follows:

$$PC(w) = \frac{E(w)}{\left(\varepsilon + \sum_n A_n(w)\right)} \tag{25}$$

here $A_n(w)$ is the local amplitude on scale $n$ and $E(w)$ is the magnitude of the response vector in $w$ on $n$. $\varepsilon$ is a small positive number and $PC_m(w) = \max(PC_1(w), PC_2(w))$.

The parameters of EMO algorithm are set according to the criteria presented in Birbil and Fang (2003) and reported in Table 1; also, they are used for all the tests, and only the $\text{Iter}_{max}$ is modified for the statistical analysis. The values in Table 1 are specially selected for the optimization problem considering the MCET; they could be tuned depending on the problem to solve.

### 5.1 Minimum cross entropy results

This section analyzes the thresholding results obtained by the EMO algorithm using as objective function the MCET (Eq. 15) (Li and Lee 1993). The proposed approach is applied to the complete set of benchmark images, and the segmentation results are presented in Table 2. Such values correspond to the best thresholds found by the EMO algorithm considering four different segmentation levels $nt = 2, 3, 4, 5$. Table 2 also shows the values of the statistical metrics (PSNR, STD, SSIM, and FSIM) obtained for each image in each level. From Table 2, it is possible to analyze that the values obtained by EMO are stable (STD column) even if the number of thresholds is increased. On the other hand, the quality of the output segmented image obtained by EMO is measured with the PSNR, SSIM, and FSIM. The corresponding columns in Table 2 provide evidence of the segmentation capacities of the proposed approach according to the definition of each metric.

From the set of eleven benchmark images, five of them are selected due to the complexity of their histograms. The pictures have been chosen to show the segmentation results graphically. Figure 1 presents the images selected from the benchmark set and their respective histograms which possess irregular distributions (particularly Fig. 1j). Under such circumstances, classical methods face great difficulties to find the best threshold values.

The five selected images are processed using the proposed algorithm based on EMO and MCET. The results presented in Fig. 2 consider four different threshold levels $th = 2, 3, 4, 5$. Figure 2 also shows the evolution of the objective function during one execution. From the results, it is possible to appreciate that the EMO–MCET converges (stabilizes) around the first 100 iterations. The segmented images provide evidence that the outcome is better with $th = 4$ and $th = 5$; however, if the segmentation task does not require to be extremely accurate, then it is possible to select $th = 3$.

### 5.2 Comparisons

In order to demonstrate that the use of EMO and MCET is an interesting alternative for MT, the proposed algorithm is compared with two state-of-the-art implementations. The methods employed for comparison are: the differential evolution (DE) (Storn and Price 1997; Sarkar et al. 2015), the particle swarm optimization (PSO) (Kennedy and Eberhart 1995; Yin 2007), the harmony search (HS) (Loganathan et al. 2001), the social spider algorithm (SSA) (Yu and Li 2015), and the artificial bee colony (ABC) (Karaboga and Basturk 2007) all those methods use the minimum cross entropy as objective function. The six algorithms were run 35 times over each selected image. The images used for this test are the same of the selected in Sect. 5.1 (Cameraman, Lena, Baboon, Hunter and Butterfly). For each image (I) the PSNR, STD, SSIM, FSIM values and the mean of the objective function are computed.

The comparison results between the six methods are divided into two groups: in the first group, Table 3 shows the STD and mean values of the MCET as the fitness function. On the other hand, in the second group Tables 4 and 5 present the values of the quality metrics obtained after applying the thresholds over the test images. Such values provide evidence that the segmented images obtained using the thresholds computed by the MCET have better quality, in specific the computed by EMO.

The fitness (also called objective function) values of five methods are statistically compared with EMO using a nonparametric significance proof known as the Wilcoxon's rank test (García et al. 2009) that is conducted with 35 independent samples. Such proof allows assessing result differences among two related methods. The analysis is performed considering a 5% significance level over the best fitness (MCET)

**Table 2** Results after applying the EMO with MCET to the set of benchmark images

| Image | k | Thresholds $x_t^B$ | PSNR | STD | SSIM | FSIM |
|---|---|---|---|---|---|---|
| Camera man | 2 | 51, 137 | 18.3128 | 0.0000 | 0.8564 | 0.8100 |
| | 3 | 31, 84, 144 | 21.6180 | 3.5466 E−05 | 0.9003 | 0.8536 |
| | 4 | 30, 77, 126, 158 | 23.2970 | 1.7459 E−04 | 0.9103 | 0.8848 |
| | 5 | 28, 70, 114, 145, 171 | 23.9009 | 4.1780 E−03 | 0.9237 | 0.8968 |
| Lena | 2 | 81, 140 | 20.9383 | 3.2483 E−05 | 0.8612 | 0.8462 |
| | 3 | 73, 119, 165 | 21.5202 | 0.0000 | 0.8794 | 0.8426 |
| | 4 | 70, 108, 140, 176 | 23.4480 | 7.2898 E−05 | 0.9006 | 0.8702 |
| | 5 | 59, 86, 115, 143, 177 | 24.9227 | 2.0784 E−04 | 0.9206 | 0.8930 |
| Baboon | 2 | 90, 145 | 21.4273 | 0.0000 | 0.9130 | 0.9155 |
| | 3 | 71, 110, 152 | 23.1514 | 1.5435 E−05 | 0.9431 | 0.9443 |
| | 4 | 64, 99, 131, 163 | 23.9436 | 8.9906 E−05 | 0.9591 | 0.9601 |
| | 5 | 56, 84, 111, 138, 168 | 25.0505 | 4.4126 E−04 | 0.9679 | 0.9676 |
| Hunter | 2 | 26, 90 | 22.8504 | 0.0000 | 0.9008 | 0.8920 |
| | 3 | 18, 60, 116 | 22.1675 | 5.2461 E−04 | 0.8942 | 0.8961 |
| | 4 | 14, 43, 84, 131 | 23.1216 | 7.4499 E−05 | 0.9189 | 0.9216 |
| | 5 | 11, 31, 60, 96, 137 | 24.3002 | 1.2439 E−03 | 0.9392 | 0.9425 |
| Airplane | 2 | 97, 161 | 25.4876 | 2.3546 E−05 | 0.9624 | 0.9142 |
| | 3 | 76, 126, 178 | 24.2310 | 1.7566 E−04 | 0.9543 | 0.9058 |
| | 4 | 63, 105, 147, 190 | 24.0633 | 1.5277 E−04 | 0.9407 | 0.9029 |
| | 5 | 61, 100, 135, 174, 203 | 23.6548 | 1.0091 E−04 | 0.9395 | 0.9072 |
| Peppers | 2 | 57, 130 | 19.4660 | 0.0000 | 0.8795 | 0.8642 |
| | 3 | 45, 88, 138 | 23.2943 | 5.5620 E−03 | 0.9164 | 0.8858 |
| | 4 | 43, 83, 124, 168 | 21.9553 | 0.0000 | 0.9015 | 0.8674 |
| | 5 | 42, 80, 111, 142, 175 | 23.4937 | 1.4000 E−02 | 0.9238 | 0.8929 |
| Living room | 2 | 73, 134 | 19.4210 | 0.0000 | 0.8490 | 0.8427 |
| | 3 | 44, 95, 145 | 20.2041 | 1.0570 E−05 | 0.8780 | 0.8760 |
| | 4 | 38, 82, 123, 160 | 21.4640 | 1.5015 E−05 | 0.9101 | 0.9052 |
| | 5 | 33, 68, 102, 133, 167 | 23.1660 | 2.5034 E−03 | 0.9370 | 0.9303 |
| Blonde | 2 | 38, 122 | 18.4613 | 0.0000 | 0.8557 | 0.8598 |
| | 3 | 33, 107, 155 | 18.0949 | 0.0000 | 0.8336 | 0.8422 |
| | 4 | 31, 96, 132, 164 | 19.5506 | 2.5181 E−06 | 0.8801 | 0.8881 |
| | 5 | 19, 63, 104, 139, 168 | 22.9896 | 9.1403 E−03 | 0.9339 | 0.9150 |
| Bridge | 2 | 76, 142 | 20.4912 | 2.0776 E−03 | 0.8705 | 0.8768 |
| | 3 | 57, 104, 161 | 20.9807 | 1.7097 E−03 | 0.9014 | 0.9114 |
| | 4 | 49, 88, 129, 181 | 21.5630 | 1.1741 E−03 | 0.9208 | 0.9229 |
| | 5 | 43, 76, 108, 146, 192 | 22.5800 | 3.7022 E−03 | 0.9414 | 0.9398 |
| Butterfly | 2 | 88, 138 | 22.9957 | 3.5474E−04 | 0.9064 | 0.8580 |
| | 3 | 77, 113, 156 | 23.3959 | 8.0843E−05 | 0.9290 | 0.8802 |
| | 4 | 70, 97, 125, 161 | 25.3474 | 9.2064E−05 | 0.9548 | 0.9230 |
| | 5 | 66, 88, 111, 135, 165 | 26.6674 | 6.6659E−03 | 0.9681 | 0.9489 |
| Lake | 2 | 73, 141 | 24.6248 | 0.0000 | 0.9467 | 0.9305 |
| | 3 | 62, 104, 162 | 24.2344 | 1.3607 E−04 | 0.9523 | 0.9398 |
| | 4 | 57, 91, 144, 195 | 20.9612 | 2.5874 E−04 | 0.9370 | 0.9170 |
| | 5 | 50, 75, 113, 158, 198 | 22.9608 | 3.1096 E−04 | 0.9540 | 0.9391 |

**(a)**
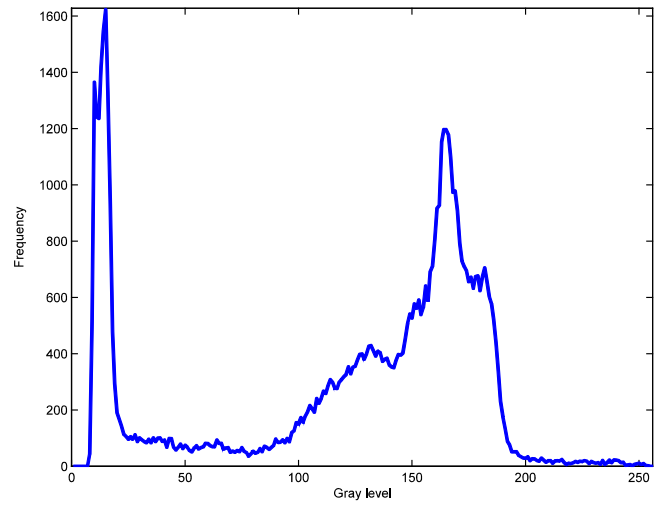


**(b)**



**(c)**
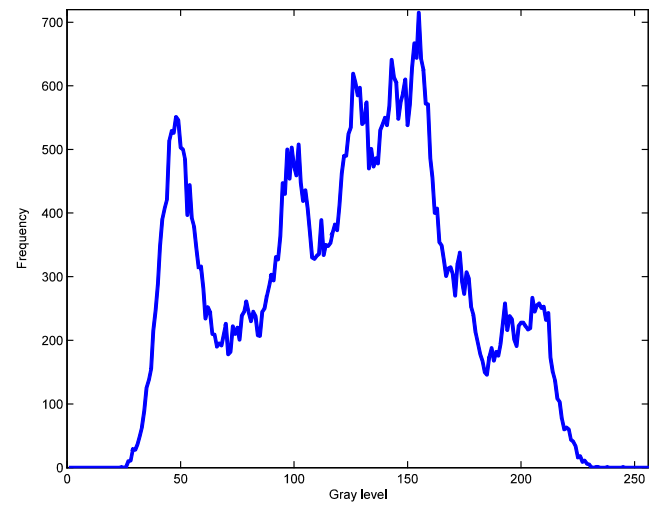


**(d)**
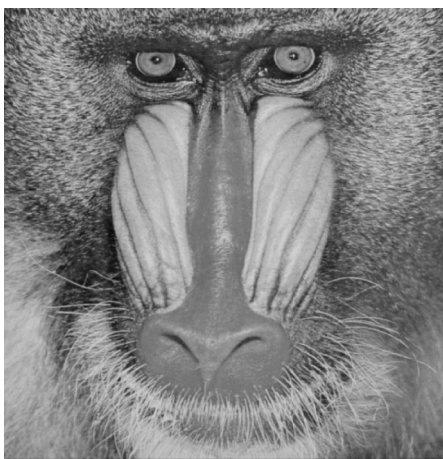


**(e)**
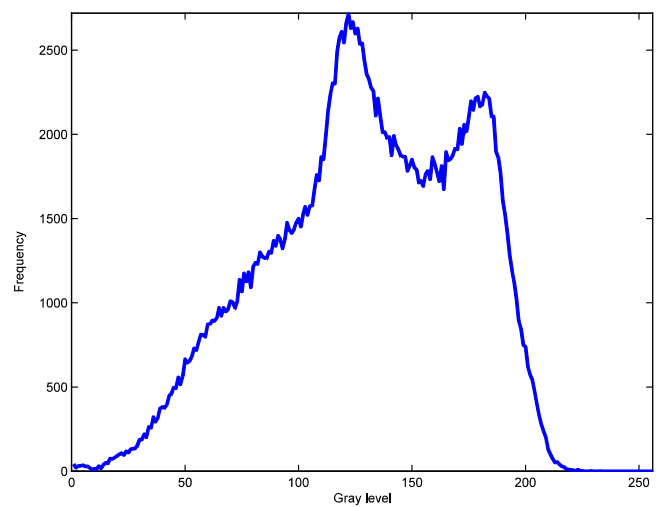


**(f)**

**Fig. 1** **a** Camera man, **c** Lena, **e** Baboon, **g** Hunter and **i** Butterfly, the selected benchmark images. **b**, **d**, **f**, **h**, **j** histograms of the images
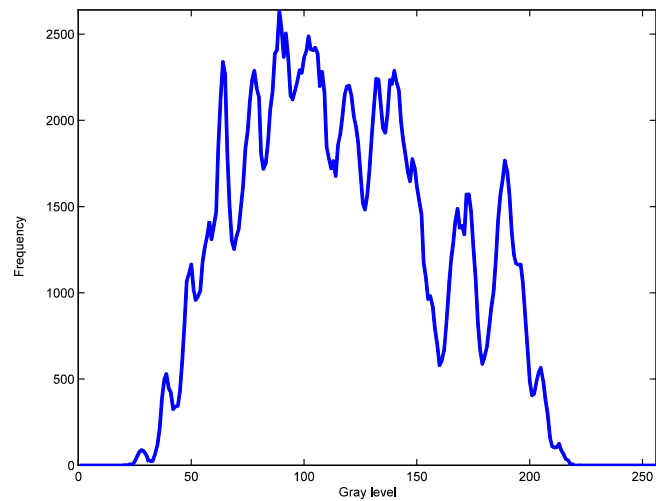
**(g)**
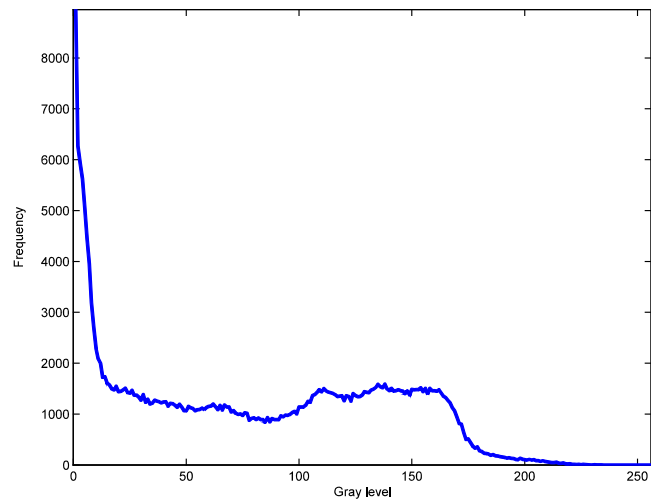


**(h)**
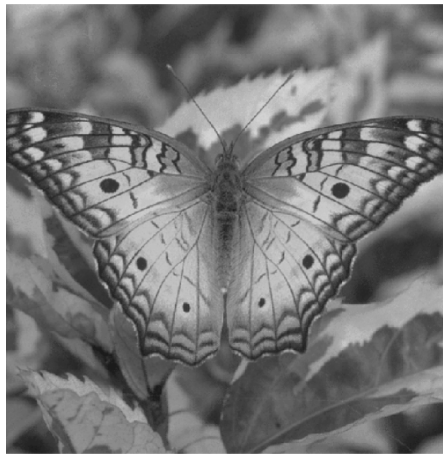


**(i)**



**(j)**

**Fig. 1** continued

value data corresponding to the five threshold points. Table 6 reports the $p$-values produced by Wilcoxon's test for a pairwise comparison of the fitness function between two groups formed as EMO versus DE, EMO versus PSO, EMO versus HS, EMO versus SSA and EMO versus ABC. As a null hypothesis, it is assumed that there is no difference between the values of the two algorithms tested. The alternative hypothesis considers an existent difference between the values of both approaches. All $p$ values reported in Table 6 are less than 0.05 (5% significance level) which is strong evidence against the null hypothesis, indicating that the EMO fitness values for the performance are statistically better, and it has not occurred by chance.

On the other hand, Table 7 presents a comparison between the EMO and the five selected algorithms regarding the computational time. This measurement is used to evaluate the computational effort required for each algorithm. For this comparison, 35 independent experiments are performed using each algorithm. For this test, the number of iterations is set to 1000, and each algorithm runs over all the images. The time is stored after each experiment for a single image. At the end, the mean is computed. This test is performed to provide evidence of the speed of EMO. Since the six algorithms are complex stochastic processes, it is difficult to analyze their complexity and the computational time is the best tool to support the results.

From Table 7, it is possible to see that the EMO-based approach in most of the cases needs less time to find the best optimal solution. When the number of thresholds increases (for example with $k = 5$), EMO requires more time than DE, PSO or SSA. However, the time is consistent for each experiment. On the other hand, the HS approach requires more time
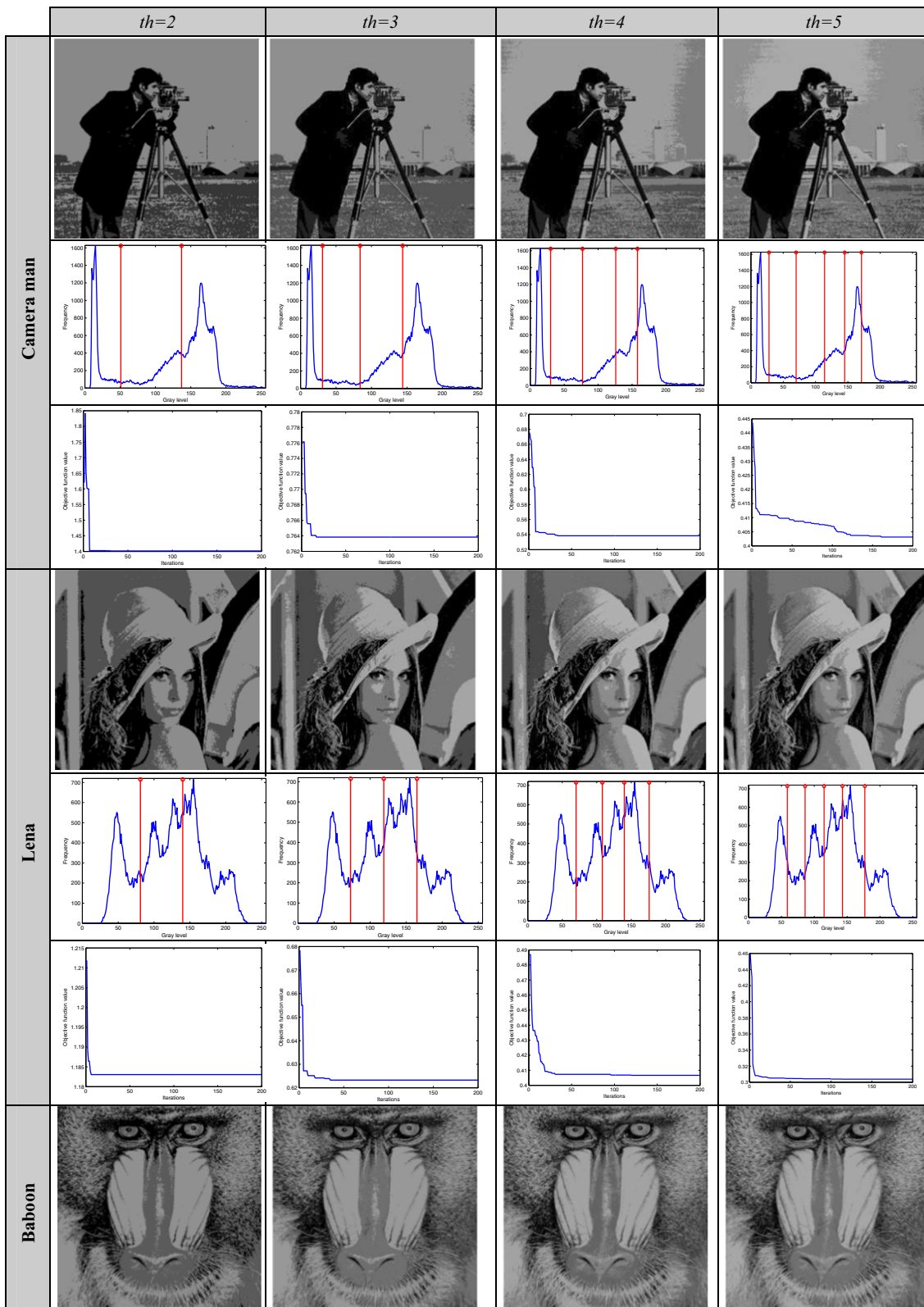
**Fig. 2** Results after applying the EMO using minimum cross entropy over the selected benchmark images
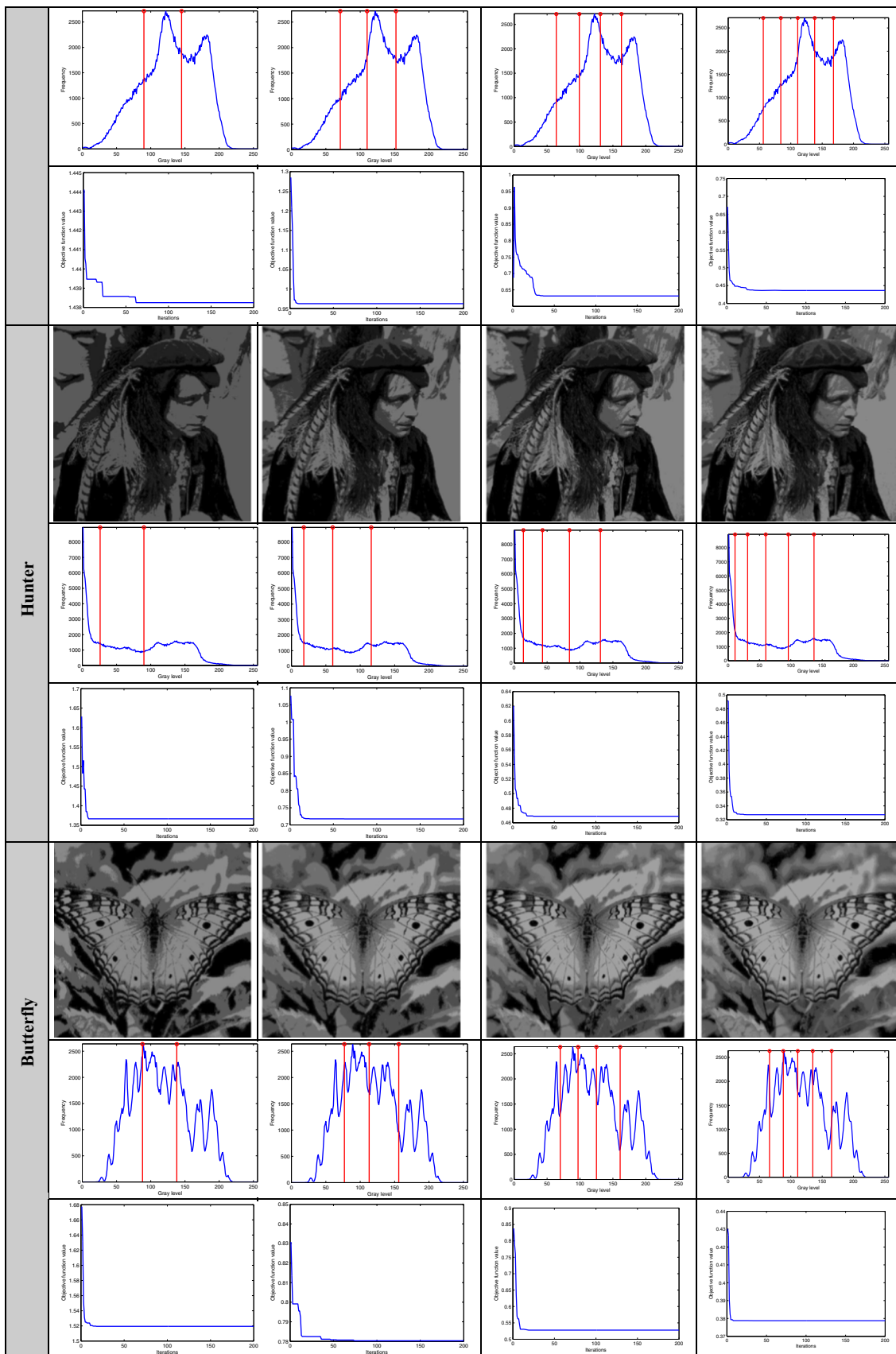
**Fig. 2** continued

**Table 3** Comparison of the STD and mean values of the EMO, DE, PSO, HS, SSA and ABC applied over the selected test images using minimum cross entropy

| I | k | EMO | | DE | | PSO | | HS | | SSA | | ABC | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | STD | Mean | STD | Mean | STD | Mean | STD | Mean | STD | Mean | STD | Mean |
| Camera man | 2 | 0.0000 E−00 | 1.4016 | 1.9944 E−02 | 1.4168 | 3.7218 E−02 | 1.4280 | 1.8416 E−02 | 1.4171 | 2.5182 E−02 | 1.4303 | 2.0502 E−02 | 1.4244 |
| | 3 | 3.5500 E−05 | 0.7638 | 4.4425 E−02 | 0.8208 | 5.8427 E−02 | 0.8352 | 5.4954 E−02 | 0.8256 | 6.3950 E−02 | 0.8450 | 6.9090 E−02 | 0.8499 |
| | 4 | 1.7458 E−04 | 0.5385 | 4.2329 E.02 | 0.6162 | 4.8551 E−02 | 0.6350 | 4.1648 E−02 | 0.6122 | 4.9743 E−02 | 0.6317 | 6.1882 E−02 | 0.6337 |
| | 5 | 4.1780 E−03 | 0.4061 | 2.7961 E−02 | 0.4792 | 4.7013 E−02 | 0.4982 | 3.0436 E−02 | 0.4808 | 3.5352 E−02 | 0.5009 | 5.0324 E−02 | 0.5011 |
| Lena | 2 | 3.2500 E−05 | 1.3663 | 2.1788 E−02 | 1.3829 | 2.3231 E−02 | 1.3915 | 1.6316 E−02 | 1.3834 | 3.5854 E−02 | 1.4001 | 2.4503 E−02 | 1.3897 |
| | 3 | 0.0000 E 00 | 0.7174 | 4.8968 E−02 | 0.7853 | 4.5776 E−02 | 0.7930 | 4.9205 E−02 | 0.7884 | 5.2360 E−02 | 0.7992 | 6.7885 E−02 | 0.7927 |
| | 4 | 7.2900 E−05 | 0.4687 | 4.2863 E−02 | 0.5511 | 5.1150 E−02 | 0.5751 | 4.0849 E−02 | 0.5645 | 5.7182 E−02 | 0.5847 | 6.7782 E−02 | 0.5731 |
| | 5 | 2.0784 E−04 | 0.3273 | 3.7746 E−02 | 0.4278 | 5.7899 E−02 | 0.4486 | 2.8957 E−02 | 0.4391 | 4.9478 E−02 | 0.4703 | 5.5795 E−02 | 0.4695 |
| Baboon | 2 | 0.0000 E 00 | 1.1905 | 1.6596 E−02 | 1.2096 | 3.2448 E−02 | 1.2188 | 1.5767 E−02 | 1.2070 | 2.2011 E−02 | 1.2129 | 2.1466 E−02 | 1.2157 |
| | 3 | 1.5400 E−05 | 0.7319 | 3.5551 E−02 | 0.7902 | 4.3116 E−02 | 0.8014 | 3.0580 E−02 | 0.7775 | 4.7458 E−02 | 0.8043 | 4.9929 E−02 | 0.7973 |
| | 4 | 8.9900 E−05 | 0.5001 | 4.6194 E−02 | 0.5783 | 4.0912 E−02 | 0.5961 | 3.6306 E−02 | 0.5811 | 4.6260 E−02 | 0.6045 | 7.3100 E−02 | 0.6113 |
| | 5 | 4.4126 E−04 | 0.3641 | 3.7222 E−02 | 0.4577 | 4.6472 E−02 | 0.4772 | 4.2383 E−02 | 0.4646 | 3.1903 E−02 | 0.4746 | 5.2018 E−02 | 0.4808 |
| Hunter | 2 | 0.0000 E 00 | 2.7409 | 2.5944 E−02 | 2.7653 | 6.5864 E−02 | 2.8044 | 2.7857 E−02 | 2.7723 | 4.1153E−02 | 2.7946 | 5.9354 E−02 | 2.7886 |
| | 3 | 5.2461 E−04 | 1.6281 | 5.8087 E−02 | 1.7308 | 8.3182 E−02 | 1.7503 | 5.9673 E−02 | 1.7144 | 9.8277E−02 | 1.7862 | 1.0149 E−01 | 1.7979 |
| | 4 | 7.4500 E−05 | 1.0308 | 9.5446 E−02 | 1.2322 | 1.1723 E−01 | 1.2585 | 1.0727E−02 | 1.2189 | 1.1770E−01 | 1.2866 | 1.4014 E−01 | 1.2744 |
| | 5 | 1.2439 E−03 | 0.7557 | 8.1586 E−02 | 0.9689 | 8.9279 E−02 | 1.0207 | 8.2197 E−02 | 0.9635 | 1.0031E−01 | 1.0154 | 1.4574 E−01 | 1.0311 |
| Butterfly | 2 | 3.5474 E−04 | 1.1832 | 6.8641 E−03 | 1.1896 | 2.1270 E−02 | 1.2046 | 1.3592 E−02 | 1.1975 | 1.8247E−02 | 1.2054 | 1.7517 E−02 | 1.1970 |
| | 3 | 8.0800 E−05 | 0.6233 | 4.8867 E−02 | 0.6973 | 6.5857 E−02 | 0.7193 | 5.7131 E−02 | 0.6941 | 5.9435E−02 | 0.7196 | 4.6726 E−02 | 0.6912 |
| | 4 | 9.2100 E−05 | 0.4065 | 3.9764 E−02 | 0.5133 | 6.3008 E−02 | 0.5353 | 4.8171 E−02 | 0.5128 | 6.0795E−02 | 0.5351 | 6.4948 E−02 | 0.5357 |
| | 5 | 6.6659 E−03 | 0.3076 | 3.4683 E−02 | 0.4024 | 4.9155 E−02 | 0.4195 | 4.5875 E−02 | 0.4016 | 5.4862E−02 | 0.4243 | 6.1034 E−02 | 0.4281 |

**Table 4** Comparison of the PSNR, SSIM and FSIM values of the EMO, DE and PSO applied over the selected test images using the MCET method

| I | k | EMO | | | DE | | | PSO | | |
|---|---|------|------|------|------|------|------|------|------|------|
| | | PSNR | SSIM | FSIM | PSNR | SSIM | FSIM | PSNR | SSIM | FSIM |
| Camera man | 2 | 18.4825 | 0.8564 | 0.8139 | 18.2128 | 0.8411 | 0.7986 | 18.1977 | 0.8450 | 0.8002 |
| | 3 | 21.7818 | 0.8994 | 0.8525 | 21.4468 | 0.8640 | 0.8113 | 21.0884 | 0.8089 | 0.8106 |
| | 4 | 23.2188 | 0.9091 | 0.8833 | 22.1301 | 0.8617 | 0.8277 | 22.0455 | 0.8290 | 0.8288 |
| | 5 | 23.6259 | 0.9204 | 0.8932 | 23.4241 | 0.8782 | 0.8383 | 22.5410 | 0.8141 | 0.8291 |
| Lena | 2 | 21.2340 | 0.8625 | 0.8619 | 20.8154 | 0.8313 | 0.8157 | 20.7911 | 0.8268 | 0.8063 |
| | 3 | 21.5202 | 0.8794 | 0.8712 | 21.4249 | 0.8395 | 0.7940 | 21.0090 | 0.8467 | 0.7804 |
| | 4 | 23.5538 | 0.9001 | 0.8847 | 22.4228 | 0.8628 | 0.8287 | 22.9989 | 0.8589 | 0.8165 |
| | 5 | 24.8234 | 0.9203 | 0.8995 | 23.5816 | 0.8820 | 0.8426 | 23.3937 | 0.8925 | 0.8228 |
| Baboon | 2 | 21.9085 | 0.9149 | 0.9183 | 21.4273 | 0.8879 | 0.8919 | 21.3669 | 0.8724 | 0.8843 |
| | 3 | 23.0794 | 0.9425 | 0.9438 | 22.6824 | 0.9130 | 0.9145 | 22.7909 | 0.8810 | 0.9043 |
| | 4 | 23.8567 | 0.9583 | 0.9590 | 23.4748 | 0.9201 | 0.9175 | 22.9763 | 0.9160 | 0.9117 |
| | 5 | 24.8360 | 0.9669 | 0.9669 | 24.0320 | 0.9211 | 0.9151 | 23.7103 | 0.9131 | 0.9145 |
| Hunter | 2 | 22.8504 | 0.9008 | 0.8976 | 22.8220 | 0.8757 | 0.8722 | 22.7911 | 0.8521 | 0.8680 |
| | 3 | 22.7167 | 0.9007 | 0.8958 | 22.1533 | 0.8615 | 0.8586 | 21.0875 | 0.8649 | 0.8681 |
| | 4 | 23.0168 | 0.9172 | 0.9205 | 22.3770 | 0.8840 | 0.8796 | 22.6372 | 0.8591 | 0.8576 |
| | 5 | 24.1474 | 0.9376 | 0.9408 | 23.3941 | 0.9070 | 0.9027 | 22.5532 | 0.9047 | 0.8990 |
| Butterfly | 2 | 24.0269 | 0.9055 | 0.9039 | 22.9003 | 0.8885 | 0.8330 | 22.8248 | 0.8726 | 0.8301 |
| | 3 | 23.3668 | 0.9281 | 0.9216 | 22.9889 | 0.8917 | 0.8259 | 23.2509 | 0.8852 | 0.8417 |
| | 4 | 25.2881 | 0.9543 | 0.9380 | 24.2203 | 0.9103 | 0.8490 | 24.1546 | 0.9005 | 0.8227 |
| | 5 | 26.2271 | 0.9655 | 0.9425 | 24.5201 | 0.9217 | 0.8740 | 24.1146 | 0.8405 | 0.7619 |

**Table 5** Comparison of the PSNR, SSIM and FSIM values of the HS, SSA and ABC applied over the selected test images using the MCET method

| I | k | HS | | | SSA | | | ABC | | |
|---|---|------|------|------|------|------|------|------|------|------|
| | | PSNR | SSIM | FSIM | PSNR | SSIM | FSIM | PSNR | SSIM | FSIM |
| Camera man | 2 | 18.2443 | 0.8445 | 0.8085 | 18.2825 | 0.8459 | 0.7999 | 18.2044 | 0.8542 | 0.8102 |
| | 3 | 21.5535 | 0.8870 | 0.8550 | 20.7268 | 0.8725 | 0.8378 | 21.5797 | 0.8848 | 0.8422 |
| | 4 | 21.9599 | 0.8721 | 0.8578 | 22.7573 | 0.8959 | 0.8627 | 22.2802 | 0.8913 | 0.8675 |
| | 5 | 22.9764 | 0.9020 | 0.8798 | 23.0061 | 0.9001 | 0.8701 | 22.1411 | 0.9106 | 0.8814 |
| Lena | 2 | 20.9209 | 0.8450 | 0.8475 | 20.9073 | 0.8506 | 0.8254 | 20.8491 | 0.8511 | 0.8458 |
| | 3 | 21.0785 | 0.8533 | 0.8371 | 21.2115 | 0.8617 | 0.8257 | 21.2352 | 0.8532 | 0.8368 |
| | 4 | 21.3584 | 0.8572 | 0.8315 | 22.5265 | 0.8892 | 0.8469 | 22.3370 | 0.8627 | 0.8524 |
| | 5 | 23.1562 | 0.8925 | 0.8622 | 23.1375 | 0.8986 | 0.8641 | 23.2626 | 0.8795 | 0.8671 |
| Baboon | 2 | 21.3029 | 0.8754 | 0.8940 | 21.0843 | 0.8983 | 0.9201 | 21.7885 | 0.9067 | 0.9089 |
| | 3 | 22.6206 | 0.9245 | 0.9358 | 22.5477 | 0.9238 | 0.9350 | 22.1142 | 0.9289 | 0.9308 |
| | 4 | 23.0713 | 0.9319 | 0.9444 | 22.6867 | 0.9310 | 0.9327 | 23.0584 | 0.9435 | 0.9464 |
| | 5 | 22.8485 | 0.9301 | 0.9366 | 23.7274 | 0.9340 | 0.9446 | 23.2269 | 0.9516 | 0.9516 |
| Hunter | 2 | 22.4354 | 0.8829 | 0.8838 | 21.8970 | 0.8767 | 0.8754 | 21.8218 | 0.8867 | 0.8800 |
| | 3 | 22.3198 | 0.8801 | 0.9015 | 21.9116 | 0.8744 | 0.8860 | 21.3750 | 0.8830 | 0.8845 |
| | 4 | 22.5711 | 0.8912 | 0.9019 | 22.0705 | 0.8926 | 0.9040 | 22.7938 | 0.9090 | 0.9103 |
| | 5 | 22.5337 | 0.8906 | 0.9093 | 22.6585 | 0.9179 | 0.9046 | 22.7033 | 0.9165 | 0.9148 |
| Butterfly | 2 | 22.9358 | 0.8567 | 0.8722 | 22.8141 | 0.8845 | 0.8544 | 22.9281 | 0.9066 | 0.8567 |
| | 3 | 22.7386 | 0.8792 | 0.8595 | 23.1200 | 0.8916 | 0.8741 | 22.9124 | 0.9205 | 0.8709 |
| | 4 | 22.8761 | 0.9091 | 0.8722 | 23.2632 | 0.8990 | 0.9017 | 23.5058 | 0.9319 | 0.8904 |
| | 5 | 23.4250 | 0.9284 | 0.8875 | 24.2389 | 0.9012 | 0.9077 | 24.3144 | 0.9429 | 0.9095 |

**Table 6** Wilcoxon *p* values of the compared algorithms EMO versus DE, EMO versus PSO, EMO versus HS, EMO versus SSA and EMO versus ABC

| I | k | *p* values | | | | |
|---|---|---|---|---|---|---|
| | | EMO versus DE | EMO versus PSO | EMO versus HS | EMO versus SSA | EMO versus ABC |
| Camera man | 2 | 1.5291E−14 | 1.5337E−14 | 1.5337E−14 | 1.5328E−14 | 5.8049E−14 |
| | 3 | 2.3579E−13 | 2.3579E−13 | 2.3579E−13 | 2.3579E−13 | 2.3579E−13 |
| | 4 | 2.1460E−13 | 2.1460E−13 | 2.1460E−13 | 2.1460E−13 | 2.1460E−13 |
| | 5 | 5.6572E−13 | 5.6572E−13 | 6.1678E−13 | 5.6572E-13 | 5.6572E−13 |
| Lena | 2 | 1.3652E−11 | 2.5260E−13 | 2.5248E−13 | 2.3305E−11 | 2.5198E−13 |
| | 3 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 |
| | 4 | 2.3579E−13 | 2.3579E−13 | 2.3579E−13 | 2.3579E−13 | 2.3579E−13 |
| | 5 | 5.3642E−13 | 5.3642E−13 | 5.3642E−13 | 5.3642E−13 | 5.3642E−13 |
| Baboon | 2 | 1.5300E−14 | 1.5346E−14 | 1.5337E−14 | 1.5328E−14 | 5.8049E−14 |
| | 3 | 2.6616E−13 | 2.6616E−13 | 2.6616E−13 | 2.6616E−13 | 2.6616E−13 |
| | 4 | 2.2955E−13 | 2.2955E−13 | 2.2955E−13 | 2.2955E−13 | 2.2955E−13 |
| | 5 | 5.9456E−13 | 5.9456E−13 | 5.9456E−13 | 5.9456E−13 | 5.9456E−13 |
| Hunter | 2 | 5.7948E−14 | 5.8015E−14 | 1.5346E−14 | 1.5337E−14 | 5.7914E−14 |
| | 3 | 1.7183E−13 | 1.7E−183E−13 | 1.7183E−13 | 1.7183E−13 | 1.7183E−13 |
| | 4 | 4.6150E−13 | 4.6150E−13 | 4.6150E−13 | 4.6150E−13 | 4.6150E−13 |
| | 5 | 5.1937E−13 | 5.1937E−13 | 5.1937E−13 | 5.1937E−13 | 5.1937E−13 |
| Airplane | 2 | 2.5210E−14 | 2.1814E−14 | 2.6427E−14 | 2.1840E−14 | 8.7304E−14 |
| | 3 | 2.4220E−13 | 2.4220E−13 | 2.4220E−13 | 2.4220E−13 | 2.4220E−13 |
| | 4 | 5.0448E−13 | 5.0448E−13 | 5.0448E−13 | 5.0448E−13 | 5.0448E−13 |
| | 5 | 3.6185E−13 | 3.6185E−13 | 3.6185E−13 | 3.6185E−13 | 3.6185E−13 |
| Peppers | 2 | 1.5328E−14 | 1.5337E−14 | 5.7847E−14 | 1.5328E−14 | 1.5310E−14 |
| | 3 | 1.6535E−12 | 1.6535E−12 | 1.6535E−12 | 3.5421E−12 | 4.1516E−13 |
| | 4 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 |
| | 5 | 2.1478E−12 | 5.0024E−13 | 2.1478E−12 | 2.1478E−12 | 5.0024E−13 |
| Living room | 2 | 1.5346E−14 | 5.7981E−14 | 1.5310E−14 | 1.5328E−14 | 5.7948E−14 |
| | 3 | 2.1840E−14 | 2.1840E−14 | 2.1840E−14 | 2.1840E−14 | 2.1840E−14 |
| | 4 | 1.7580E−13 | 1.7580E−13 | 1.7580E−13 | 1.7580E−13 | 1.7580E−13 |
| | 5 | 3.0306E−13 | 3.0306E−13 | 3.0306E−13 | 3.0306E−13 | 3.0306E−13 |
| Blonde | 2 | 1.5291E−14 | 1.5310E−14 | 1.5310E−14 | 1.5319E−14 | 1.5328E−14 |
| | 3 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 | 1.5346E−14 |
| | 4 | 2.1840E−14 | 2.1840E−14 | 2.1840E−14 | 2.1840E−14 | 2.1840E−14 |
| | 5 | 1.1710E−12 | 6.4191E−13 | 7.6272E−13 | 7.6272E−13 | 1.2753E−12 |
| Bridge | 2 | 9.1839E−14 | 7.7840E−13 | 5.6912E−14 | 4.7117E−14 | 2.9739E−13 |
| | 3 | 3.6102E−13 | 2.5310E−13 | 3.6102E−13 | 3.0237E−13 | 3.6102E−13 |
| | 4 | 3.7405E−13 | 3.7405E−13 | 3.7405E−13 | 3.7405E−13 | 3.7405E−13 |
| | 5 | 5.6968E−13 | 5.6968E−13 | 5.6968E−13 | 5.6968E−13 | 5.6968E−13 |
| Butterfly | 2 | 2.1394E−12 | 4.4802E−13 | 1.2753E−12 | 3.1793E−12 | 7.5207E−13 |
| | 3 | 2.5273E−13 | 2.5273E−13 | 2.5273E−13 | 2.5273E−13 | 2.5273E−13 |
| | 4 | 3.0752E−13 | 3.0752E−13 | 3.0752E−13 | 3.0752E−13 | 3.0752E−13 |
| | 5 | 5.2969E−13 | 5.2969E−13 | 5.2969E−13 | 5.2969E−13 | 5.2969E−13 |
| Lake | 2 | 2.1198E−13 | 1.5337E−14 | 1.5337E−14 | 1.5337E−14 | 1.5319E−14 |
| | 3 | 4.7746E−13 | 4.7746E−13 | 4.7746E−13 | 4.7746E−13 | 4.7746E−13 |
| | 4 | 4.2073E−13 | 4.2073E−13 | 4.2073E−13 | 4.2073E−13 | 4.2073E−13 |
| | 5 | 5.9016E−13 | 5.9016E−13 | 5.9016E−13 | 5.9016E−13 | 5.9016E−13 |

**Table 7** Mean of the computational time of EMO, DE, PSO, HS, SSA and ABC for the problem of MTH using the minimum cross entropy

| I | k | Computational time in seconds | | | | | |
|---|---|---|---|---|---|---|---|
| | | EMO | DE | PSO | HS | SSA | ABC |
| Camera man | 2 | 0.9436 | 2.8702 | 2.9128 | 5.1731 | 6.6812 | 4.9803 |
| | 3 | 1.4140 | 3.3423 | 3.0573 | 5.6769 | 7.1879 | 4.9481 |
| | 4 | 2.5064 | 3.1482 | 3.0093 | 5.3187 | 7.1748 | 5.0035 |
| | 5 | 3.7611 | 3.1084 | 3.9338 | 5.8430 | 7.2849 | 5.0408 |
| Lena | 2 | 1.4837 | 2.9474 | 3.0571 | 4.1210 | 4.7815 | 5.4233 |
| | 3 | 2.3095 | 3.1527 | 3.1800 | 4.6949 | 4.8725 | 5.9854 |
| | 4 | 2.8467 | 3.0956 | 3.0698 | 5.1584 | 4.9822 | 5.6837 |
| | 5 | 3.6665 | 3.4252 | 3.9929 | 5.6943 | 5.0831 | 5.6679 |
| Baboon | 2 | 1.4262 | 3.1317 | 3.0253 | 4.4503 | 4.7491 | 5.5001 |
| | 3 | 2.1936 | 3.1358 | 3.0467 | 4.9133 | 3.8139 | 5.5502 |
| | 4 | 2.8608 | 3.1488 | 3.0035 | 5.3466 | 3.5713 | 5.5536 |
| | 5 | 3.5343 | 3.5108 | 3.1198 | 5.6283 | 3.6267 | 5.7071 |
| Hunter | 2 | 1.4975 | 2.8719 | 3.0142 | 4.5676 | 3.4160 | 5.5245 |
| | 3 | 2.2200 | 2.9633 | 2.9040 | 4.8718 | 3.4855 | 5.6224 |
| | 4 | 3.1627 | 3.0585 | 3.0083 | 5.2737 | 3.5588 | 5.8129 |
| | 5 | 3.7280 | 3.2096 | 3.0446 | 5.7714 | 3.9146 | 5.3861 |
| Airplane | 2 | 1.5541 | 3.0118 | 3.0582 | 4.5824 | 3.4150 | 5.0549 |
| | 3 | 2.2903 | 2.9289 | 2.9800 | 4.8948 | 3.4920 | 5.3252 |
| | 4 | 2.9445 | 3.0994 | 3.1955 | 5.1260 | 3.5504 | 5.1442 |
| | 5 | 3.7565 | 3.2444 | 3.9576 | 5.2884 | 3.9235 | 5.2147 |
| Peppers | 2 | 1.4794 | 2.8190 | 2.8964 | 4.1583 | 3.4155 | 5.0975 |
| | 3 | 2.2575 | 2.7814 | 3.0897 | 4.9220 | 3.4946 | 5.1322 |
| | 4 | 3.0144 | 2.8531 | 3.0865 | 5.3589 | 3.5559 | 5.2255 |
| | 5 | 3.7579 | 3.2648 | 3.0161 | 5.7062 | 3.6226 | 5.2086 |
| Living room | 2 | 1.6184 | 2.8374 | 2.9206 | 4.4757 | 3.4112 | 5.0795 |
| | 3 | 2.2417 | 2.9434 | 2.8998 | 4.9375 | 3.4766 | 5.0830 |
| | 4 | 3.0410 | 3.0548 | 3.0273 | 5.3077 | 3.5398 | 5.1982 |
| | 5 | 3.7145 | 3.1756 | 3.8089 | 5.3819 | 3.9101 | 5.3334 |
| Blonde | 2 | 1.5284 | 2.8266 | 2.7120 | 4.4955 | 3.4527 | 5.0428 |
| | 3 | 2.2144 | 2.9556 | 2.7727 | 4.9646 | 3.4924 | 5.1213 |
| | 4 | 2.9952 | 3.0334 | 2.6926 | 5.3587 | 3.5500 | 5.2055 |
| | 5 | 3.7327 | 3.3047 | 3.7712 | 5.6642 | 3.9076 | 5.2187 |
| Bridge | 2 | 1.4589 | 2.8325 | 2.7830 | 4.5453 | 3.4253 | 5.1206 |
| | 3 | 2.2875 | 2.8466 | 2.9345 | 4.9387 | 3.4903 | 5.2781 |
| | 4 | 2.9910 | 3.8041 | 3.0259 | 5.4024 | 3.5770 | 5.3119 |
| | 5 | 3.6740 | 3.8382 | 2.9650 | 5.6041 | 3.6278 | 5.4317 |
| Butterfly | 2 | 1.5039 | 3.0215 | 2.9142 | 4.7109 | 5.2115 | 5.0502 |
| | 3 | 2.2411 | 2.9894 | 2.9877 | 5.0698 | 5.3860 | 4.9276 |
| | 4 | 3.0091 | 3.2101 | 3.8212 | 5.3102 | 5.6150 | 5.0205 |
| | 5 | 3.7002 | 3.3017 | 3.3976 | 5.7413 | 5.8124 | 5.0661 |
| Lake | 2 | 1.4726 | 2.8825 | 3.4073 | 4.5335 | 5.1208 | 4.8969 |
| | 3 | 2.1217 | 2.9470 | 3.5072 | 4.8877 | 5.2051 | 4.9079 |
| | 4 | 2.9959 | 3.2373 | 3.4629 | 5.1438 | 5.5869 | 5.0462 |
| | 5 | 3.5776 | 3.2751 | 3.7326 | 5.4053 | 5.8710 | 5.5951 |
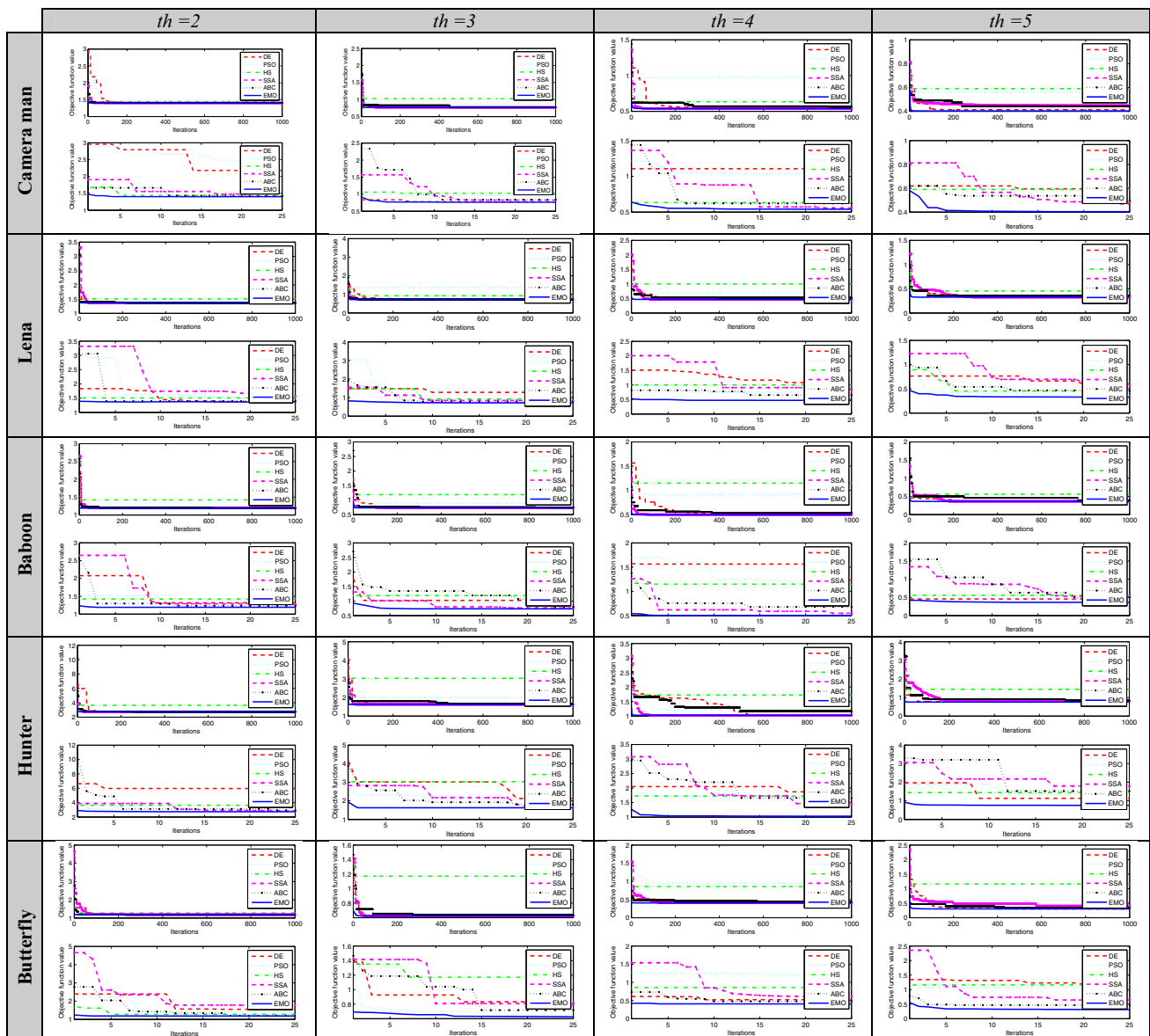
**Fig. 3** Fitness comparison of DE (*red line*), PSO (*cyan line*), HS (*green line*), SSA (*magenta line*), ABC (*black line*) and EMO (*blue line*) applied for multilevel thresholding using MCET (color figure online)

since it uses a single candidate solution along the iterative process. Meanwhile, the operators of ABC are completely iterative, and it also uses two populations to find the best solutions. The SSA algorithm has some initial configurations that affect their efficacy, the values of such configurations are difficult to set, and they also affect the computational effort. Based on these facts, the computational time of ABC is higher in comparison with EMO, DE or PSO, but is close to the values of HS and SSA.

In Fig. 3, the fitness values obtained for the five selected images are presented. For this experiment, each algorithm runs 1000 times, and the best values are stored at the end of each iteration. For a better understanding and appreci-

ation of the convergence, Fig. 3 includes the zoom of the graph of the objective function values. The zoom considers only 25 iterations for the six algorithms. This number is selected because experimentally we notice that the six algorithms decrease the MCET value in the first 25 iterations. From Fig. 3, it is possible to deduce that the proposed MTH algorithms based on EMO require fewer iterations than other similar approaches to obtain the best thresholds. In this context, the stop criteria could also be modified. Moreover, the results presented in Fig. 3 for HS support their low speed of convergence that occurs because it works only with one candidate solution at the time. Finally, it is possible to establish that the approach based on EMO and MCET reaches the

minimum cross entropy values in fewer iterations and obtain more accurately solutions than DE, PSO, HS, SSA and ABC.

## 6 Conclusions

This paper presents a new algorithm for multilevel segmentation based on the electromagnetism-like algorithm (EMO) to reduce iterations and function evaluations. The proposed approach considers the segmentation process as an optimization problem, where EMO is employed to find the optimal threshold points that minimize the cross entropy (MCET). MCET combined with EMO focuses directly on the search for the best set of threshold values. This method considers only the histogram and the number of thresholds as input. In contrast, similar approaches such as clustering techniques need to explore initialization conditions leading to more complex schemes.

The EMO-based algorithm can substantially reduce the number of function evaluations preserving the good search capabilities of an evolutionary method. The presented technique is able to find the best values even with large and complex images. In our approach, the algorithm uses particles to encode a set of candidate threshold points. The MCET is the objective function, where the quality of all the candidate solutions is evaluated. The particles are evolved using the force, charge, movement and local search operators of EMO. Once the optimal thresholds are obtained, they are used to segment the image. In order to evaluate the quality of the segmented images, the use of the PSNR, STD, SSIM and FSIM is proposed. Such metrics consider the coincidences between the original and the segmented image.

The experimental study compares the proposed approach with other five related approaches, differential evolution (DE), particle swarm optimization algorithm (PSO), harmony search (HS), social spider algorithm (SSA) and the widely used artificial bee colony (ABC). The efficiency of the algorithms is evaluated in terms of PSNR, STD, SSIM, FSIM and fitness values. Such comparisons provide evidence of the accuracy, convergence and robustness of the proposed approach. The high rate of convergence of EMO is evident on the comparisons reported. Likewise, EMO outperforms on most of the experiments providing high scores on the evaluated metrics. It is possible to establish that the approach based on EMO and MCET reaches the minimum cross entropy values in fewer iterations and obtains more accurately solutions than DE, PSO, HS, SSA and ABC. Although the results offer evidence to demonstrate that the EMO method can yield good results on complicated images, the aim of our paper is not to devise a multilevel thresholding algorithm that could beat all currently available methods, but to show that electromagnetism systems can be considered as an attractive alternative for this purpose.

In the spirit of contributing to future works, the development of thresholding techniques might take three equally important paths. The first is the current trend where thresholding approaches are beneficiated from new optimization strategies, especially if such methods improve the overall state of the art. A second path can be taken if the thresholding problem is solved with a multi-objective methodology. The incorporation of multi-objective optimizers can lead to simultaneous improvements on thresholding criteria. Such topic should be addressed to further improve the quality of segmented images. Third, the effectiveness of thresholding can be directly applied to critical image processing areas such as medical or satellite image processing. Any of the aforementioned paths will significantly contribute to the growth of the image processing community.

## References

Agrawal S, Panda R, Bhuyan S, Panigrahi BK (2013) Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm Evol Comput 11:16–30. doi:10.1016/j.swevo.2013.02.001

Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13:3066–3091. doi:10.1016/j.asoc.2012.03.072

Bhandari AK, Singh VK, Kumar A, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst Appl 41:3538–3560. doi:10.1016/j.eswa.2013.10.059

Birbil ŞI, Fang SC (2003) An electromagnetism-like mechanism for global optimization. J Glob Optim 25:263–282. doi:10.1023/A:1022452626305

Birbil ŞI, Fang SC, Sheu RL (2004) On the convergence of a population-based global optimization algorithm. J Glob Optim 30:301–318

Cao X, Li Q, Du X et al (2014) Exploring effect of segmentation scale on orient-based crop identification using HJ CCD data in Northeast China. IOP Conf Ser Earth Environ Sci 17:12047. doi:10.1088/1755-1315/17/1/012047

De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6:239–251. doi:10.1109/TEVC.2002.1011539

De Jong K (1988) Learning with genetic algorithms: an overview. Mach Learn 3:121–138. doi:10.1007/BF00113894

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B Cybern 26(1):29–41. doi:10.1109/3477.484436

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC 2005 special session on real parameter optimization. J Heuristics 15:617–644. doi:10.1007/s10732-008-9080-4

Ghamisi P, Couceiro MS, Benediktsson JA, Ferreira NMF (2012) An efficient method for segmentation of images based on fractional calculus and natural selection. Expert Syst Appl 39:12407–12417. doi:10.1016/j.eswa.2012.04.078

Gonzalez RC, Woods RE (1992) Digital image processing. Pearson, Prentice-Hall, New Jersey

Hammouche K, Diaf M, Siarry P (2008) A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Comput Vis Image Underst 109:163–175. doi:10.1016/j.cviu.2007.09.001

Horng M-H (2010) Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Syst Appl 37:4580–4592. doi:10.1016/j.eswa.2009.12.050

Horng M-H (2011) Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Syst Appl 38:13785–13791. doi:10.1016/j.eswa.2011.04.180

Horng M-H, Liou R-J (2011) Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Syst Appl 38:14805–14811. doi:10.1016/j.eswa.2011.05.069

Hung HL, Huang YF (2011) Peak to average power ratio reduction of multicarrier transmission systems using electromagnetism-like method. Int J Innov Comput Inf Control 7:2037–2050

Il-Seok O, Lee J-S, Moon B-R (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26:1424–1437. doi:10.1109/TPAMI.2004.105

Jhang JY, Lee KC (2009) Array pattern optimization using electromagnetism-like algorithm. AEU Int J Electron Commun 63:491–496. doi:10.1016/j.aeue.2008.04.001

Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29:273–285. doi:10.1016/0734-189X(85)90125-2

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471. doi:10.1007/s10898-007-9149-x

Kaur T, Saini BS, Gupta S (2016) Optimized multi threshold brain tumor image segmentation using two dimensional minimum cross entropy based on co-occurrence matrix. Springer, Berlin, pp 461–486

Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Neural Networks, 1995 proceedings, IEEE international conference, vol. 4, pp 1942–1948. doi:10.1109/ICNN.1995.488968

Khairuzzaman AKM, Chaudhury S (2017) Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Syst Appl. doi:10.1016/j.eswa.2017.04.029

Kong Y, Deng Y, Dai Q (2015) Discriminative clustering and feature selection for brain MRI segmentation. IEEE Signal Process Lett 22:573–577

Kullback S (1968) Information theory and statistics. Dover, New York

Lee CH, Chang FK (2010) Fractional-order PID controller optimization via improved electromagnetism-like algorithm. Expert Syst Appl 37:8871–8878. doi:10.1016/j.eswa.2010.06.009

Li CH, Lee CK (1993) Minimum cross entropy thresholding. Pattern Recognit 26:617–625. doi:10.1016/0031-3203(93)90115-D

Liu Y, Mu C, Kou W, Liu J (2015) Modified particle swarm optimization-based multilevel thresholding for image seg-mentation. Soft Comput 19:1311–1327. doi:10.1007/s00500-014-1345-2

Loganathan GVV, Geem ZW, Kim JH, Loganathan GVV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68. doi:10.1177/003754970107600201

Naderi B, Tavakkoli-Moghaddam R, Khalili M (2010) Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan. Knowl Based Syst 23:77–85. doi:10.1016/j.knosys.2009.06.002

Oliva D, Hinojosa S, Cuevas E et al (2017) Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm. Expert Syst Appl. doi:10.1016/j.eswa.2017.02.042

Olugbara OO, Adetiba E, Oyewole SA (2015) Pixel intensity clustering algorithm for multilevel image segmentation. Math Probl Eng. doi:10.1155/2015/649802

Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9:62–66. doi:10.1109/TSMC.1979.4310076

Pal N (1996) On minimum cross-entropy thresholding. Pattern Recognit 29:575–580. doi:10.1016/0031-3203(95)00111-5

Rocha AMAC, Fernandes EMGP (2009a) Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. Int J Comput Math 86:1932–1946. doi:10.1080/00207160902971533

Rocha AMAC, Fernandes EMGP (2009b) Modified movement force vector in an electromagnetism-like mechanism for global optimization. Optim Methods Softw 24:253–270

Sarkar S, Das S, Chaudhuri SS (2015) A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution. Pattern Recognit Lett 54:27–35. doi:10.1016/j.patrec.2014.11.009

Sarkar S, Patra GR, Das S (2011) A differential evolution based approach for multilevel image segmentation using minimum cross entropy thresholding. In: Swarm, evolutionary, and memetic computing, pp 51–58

Sathya PD, Kayalvizhi R (2011) Optimal multilevel thresholding using bacterial foraging algorithm. Expert Syst Appl 38:15549–15564. doi:10.1016/j.eswa.2011.06.004

Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. doi:10.1023/A:1008202821328

Tang K, Yuan X, Sun T et al (2011) An improved scheme for minimum cross entropy threshold selection based on genetic algorithm. Knowl Based Syst 24:1131–1138. doi:10.1016/j.knosys.2011.02.013

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612. doi:10.1109/TIP.2003.819861

Wu P, Yang W-H, Wei N-C (2004) An electromagnetism algorithm of neural network analysis-an application to textile retail operation. J Chin Inst Ind Eng 21:59–67. doi:10.1080/10170660409509387

Yang X-S (2014) Cuckoo search and firefly algorithm: overview and analysis. In: Yang X-S (ed) Cuckoo search and firefly algorithm. Springer, Berlin, Heidelberg, pp 1–26

Yin P-Y (2007) Multilevel minimum cross entropy threshold selection based on particle swarm optimization. Appl Math Comput 184:503–513. doi:10.1016/j.amc.2006.06.057

Yu JJQ, Li VOK (2015) A social spider algorithm for global optimization. Appl Soft Comput 30:614–627. doi:10.1016/j.asoc.2015.02.014

Yurtkuran A, Emel E (2010) A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. Expert Syst Appl 37:3427–3433. doi:10.1016/j.eswa.2009.10.005

Zhang L, Zhang L, XuanqinMou DZ (2011) FSIM: a feature similarity index for image. IEEE Trans Image Process 20:2378–2386