RESEARCH ARTICLE

A Comparative Analysis of Recent Metaheuristic Algorithms for Image Segmentation Using the Minimum Cross-Entropy for Multilevel Thresholding

Omar Alvarez[1], Luis A. Beltran[1], Angel Casas-Ordaz[1,*], Jorge Ramos-Frutos[2], Mario A. Navarro-Velázquez[1], Oscar Ramos-Soto[1] and Diego Oliva[1]

[1] Depto. de Ingeniería Electro-Fotónica, Universidad de Guadalajara, CUCEI, Guadalajara 44430, Jalisco, México
[2] Depto. de Ingeniería Industrial, Tecnológico Nacional de México, Jiquilpan 59514, Michoacán, México

Abstract

Metaheuristic Algorithms (MAs) are commonly used in the scope of digital image processing, in particular, image segmentation processes. This is evident in Multilevel Thresholding (MTH) methods, where the optimal threshold configuration must be found to produce high-quality segmented images. Minimum Cross-Entropy (MCE) is one of the most prominent techniques for MTH due to its simplicity and efficiency. This article proposes a comparison of recent MAs that have not yet been implemented for image segmentation. Six recently published MAs were implemented and tested on nine complicated images selected from the BSDS300 dataset. Analyzing the results reveals the best algorithm when MCE is used as the objective function. Central tendency indicators, such as Standard Deviation and mean, are also used to analyze the five threshold values. Additionally, three quality indicators used in processing images are analyzed: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Feature Similarity (FSIM). The result of this analysis allows for the quality of the segmentation of each algorithm used in the comparison. The metrics with the highest values are indicative of the most effective algorithm in terms of segmentation performance.

Keywords: image segmentation, thresholding, minimum Cross-Entropy, metaheuristics.

1 Introduction

Image segmentation has been a significant challenge in the scope of computer vision and digital image processing. This process is crucial because it simplifies and facilitates the representation of an image. More specifically, image segmentation refers to the mechanism used to divide an image into multiple segments [1]. Since one of the main challenges of image processing is to extract the desired information from a given image without affecting the other properties of the image, several methods have been developed in recent years to achieve this goal [2, 16]. This segmentation process will represent

Submitted: 03 August 2025
Accepted: 17 September 2025
Published: 20 November 2025

Vol. 1, **No.** 2, 2025.
10.62762/TSEL.2025.417356

***Corresponding author:**
✉ Angel Casas-Ordaz
angel.casas@academicos.udg.mx

Citation
Alvarez, O., Beltran, L. A., Casas-Ordaz, A., Ramos-Frutos, J., Navarro-Velázquez, M. A., Ramos-Soto, O., & Oliva, D. (2025). A Comparative Analysis of Recent Metaheuristic Algorithms for Image Segmentation Using the Minimum Cross-Entropy for Multilevel Thresholding. *ICCK Transactions on Swarm and Evolutionary Learning*, 1(2), 50–82.

some type of information for the user in the form of color, intensity, or texture. Therefore, the selection of any image segmentation technique is made after observing the domain of the problem [3]. The image segmentation process can be performed using different techniques [4], including threshold-based techniques, which use a set of thresholds to separate objects from the background; region-based techniques, which consist of splitting the original image into separate regions according to predefined criteria such as color, intensity, or object; clustering-based techniques, in which image pixels with the same properties are divided into groups; and edge-based strategies, which separate objects from the image by searching for discontinuous edges. Among these methodologies, threshold-based segmentation has stood out for its ease of implementation [5, 17]. Threshold techniques work with the image histogram and use a threshold to divide the pixels into two groups. In this case, when a single threshold is used, it is known as a two-level threshold. On the other hand, when more than two thresholds are used to separate into several groups, it is known as Multilevel Thresholding (MTH) [6]. One of the main problems with these approaches is precisely finding the thresholds that divide the image into homogeneous regions. To do this, different strategies can be found, such as Otsu [7], Kapur [9], Tsallis [8, 10, 12], Renji [11, 18], and Minimum Cross-Entropy (MCE) [13, 19].

In recent decades, various studies based on MTH have been developed. For example, the one developed by [21] proposes an improved method of image segmentation with multiple thresholds based on the Whale Optimization Algorithm (WOA). Interestingly, two other proposals based on this algorithm (WOA) use a different approach [20, 22]. On the other hand, applying a different technique, there is the approach proposed by [23], in which the Firefly Algorithm (FA) is used to optimize fuzzy parameters and obtain optimal thresholds. A recent work by [24] shows another different approach that increases the search capacity of the artificial electric field algorithm and is then used with a multilevel threshold approach. Algorithm hybridization has been a very popular topic recently. A proposal developed by [25] shows an interesting hybridization between the Sine-Cosine Algorithm (SCA) and an Equilibrium Optimizer (EO). In [25], a proposal is presented that uses the Human Mental Search (HMS) algorithm in an improved version that uses the multilevel threshold approach with minimum cross-entropy. In [26], implementations of this type can also be found that use an improved version of the algorithm applied to a multilevel thresholding approach that uses cross-entropy as the objective function. A notable study presenting a frog leap algorithm mixed with multiple strategies and horizontal and vertical crossover search, designed for multilevel image segmentation, demonstrates a concrete implementation of the methodology [14, 15, 27]. These investigations promote satisfactory results in the implementation of different versions of known algorithms with different objective functions, such as Otsu's interclass variance, Kapur's entropy, Tsalli's entropy, and MCE cross-entropy. However, it remains important to further explore the capabilities of MAs in this area of image segmentation, focusing on MTH.

This study aims to demonstrate various evolutionary strategies employed to address the issue of image segmentation, with MCE serving as the objective function. The following algorithms were used: the Differentiated Creative Search (DCS) [39], the Escape Algorithm (ESC) [28], the Moss Growth Optimization (MGO) [29], the Mirage Search Optimization (MSO) [30], the Nutcracker Optimization Algorithm (NOA) [31], and the Polar Lights Optimizer (PLO) [32]. These algorithms were selected because they were recently published, and their implementation for image segmentation had not been done previously until this manuscript. With the goal of creating a stable reference framework that allows for an objective comparison of six selected algorithms, it was decided to include a widely recognized reference in the literature: Particle Swarm Optimization (PSO). This algorithm, inspired by the collective behavior of fish, has proven effective in a wide range of optimization problems, given its extensive research and balanced application in the solution area. In order to ensure an optimal comparative analysis, PSO was applied to the same data as the main experiments, considering exactly the same conditions and reference images. This ensures that the differences in observed performance between the algorithms are not due to changes in data or experimental parameters, but also to the inherent characteristics of each technique. Using PSO as a point of comparison not only provides the possibility of contextualizing the results obtained with the six evaluated algorithms but also creates a standard benchmark against a well-documented optimization method, providing greater stability and objectivity to the objective study. This is part of the novelty of the manuscript.

The subsequent sections of this paper are structured as

follows: Section 2 covers background information on the MAs implemented in the study. Section 3 describes the methodology used to adapt the algorithms to the image segmentation problem. Section 4 discusses the experimental development of the algorithms. Finally, Section 5 concludes the manuscript.

2 Preliminaries

2.1 Particle Swarm Optimization

Particle swarm Optimization is an algorithm inspired by an animal's social behavior in their natural environment, this behavior is constant in multiple species in different environments, animals communicate with each other when hunting or when being preyed on in order to confuse the predator or in order to obtain prey (as the case may be) and feed the group; this constant communication between members of the group has an enormous influence on its movement, this social behavior was abstracted from nature and was first captured in 1995 by Kennedy et al. [47]. Over time the proposed algorithm was modified, maintaining its social inspiration but improving how particles move through space to improve its performance and avoid stagnation, one of the most recognized improvements of the PSO is the implementation of an inertial weight in 1998 by Shi et al. [48], this inertial weight scales the velocity vector over the iterations, with this scaling the algorithm refines the search, in 1999 by Shi et al. [49], a dynamic inertia was implemented that decreases its value with the passing of the iterations with this strategy the search for a more appropriate solution is increasingly refined, the inertia balances the search of the algorithm when the inertia is high the algorithm explores more and when the inertia decreases the algorithm tends to exploit in a better way hence the importance of dynamic inertia in this particular algorithm. The strategy for updating the velocity vectors is described in Eq. 1.

$$v_i^{(t+1)} = \omega^{(t)} \cdot v_i^{(t)} + c_1 \cdot r_1(p_{best_i} - x_i^{(t)}) + c_2 \cdot r_2(g_{best} - x_i^{(t)}) \tag{1}$$

where the inertial adaptive weight is defined in Eq. 2, $v_i^{(t)}$ is the actual particle position in the first iteration is randomly selected, the coefficients social and cognitive are c_2 and c_1 each of these coefficients scales the velocity of each of the particles based on its own experience or based on the experience obtained by the most optimal particle in the entire swarm. This social influence determines how the particles will be explore, r_2 and r_1 are random numbers betwin $[0, 1]$, $x_i^{(t)}$ is the actual particle position, p_{best_i} and g_{best} represents the best position find by own particle and the best position find by all swarm respectively.

$$\omega^t = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{T} \cdot t \tag{2}$$

where ω_{max} and ω_{min} represent the maximum and minimum ω values, T and t are all the iterations and the actual iteration respectively.

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{3}$$

where $x_i^{(t)}$ is the current position and $v_i^{(t+1)}$ is the velocity update of each particle.

2.2 The differentiated creative search

Differentiated Creative Search (DCS), created by Duankhan et al. [39], is an innovative proposal in the field of algorithms with a population optimization approach. The DCS algorithm introduces an innovative technique based on differentiated knowledge acquisition and creative realism, with the aim of more effectively addressing decision-making in highly complex environments. Unlike conventional Differential Evolution (DE) methods, DCS structures the search process from the perspective of divergent and convergent thinking, applied in a differentiated manner according to the performance of each of the population's individuals. The algorithm's strategy has the following approach: starting from a population with a constant size, the individuals in the population are ordered according to their performance and classified with the following characteristics.

- High performance: members belonging to this category employ divergent thinking through the DCS/Xrand/Linnik(α, σ) strategy, using Linnik flights to generate creative solutions that allow exploring rarely visited regions of the space search by extending the algorithm's search with its best solution prospects. This strategy is defined in Eq.(4).

$$v_{i,d} = x_{i,d} + \text{Linnik}(\alpha, \sigma) \tag{4}$$

  where the Linnik distribution introduces significant perturbations that favor broad and innovative exploration.

- The solution refinement strategy: utilizes the remaining population members outside the main group, applying convergent thinking to each member using the DCS/Xbest/Current-to-2rand

strategy. This approach aims to refine existing knowledge by integrating details from the leading individual in the population and two randomly selected members. This strategy is expressed in Eq (5).

$$v_{i,d} = w \cdot x_{\text{best},d} + \lambda_t(x_{r2,d} - x_{i,d}) + \omega_{i,t}(x_{r1,d} - x_{i,d}) \tag{5}$$

where $w = 1$ represents the cognitive weight, λ_t is the social sensitivity that decreases its value over time, and $\omega_{i,t} \sim U(0,1)$ represents the intensity of individual learning influenced by stochastic factors.

One of the innovations of the DCS algorithm is the application of the Differentiated Knowledge Acquisition (DKA) mechanism. This mechanism is a variation of the DE that replaces the classic crossover operator. The main objective of this new mechanism is to adapt the degree of knowledge modification for every individual belonging to the population particle, ordered by their performance ranking. The quantified knowledge acquisition rate is shown in Eq. (6).

$$\eta_{i,t} = \frac{1}{2}\left([U(0,1)\cdot\phi_{i,t}] + \begin{cases} 1 \text{ if } U(0,1) \le \phi_{i,t} \\ 0 \text{ otherwise} \end{cases}\right) \tag{6}$$

where $\phi_{i,t}$ in the Eq.(7) represents knowledge imperfection. This can be seen below:

$$\phi_{i,t} = 0.25 + 0.55 \cdot \sqrt{\frac{R_{i,t}}{NP}}, \tag{7}$$

where $R_{i,t}$ is the position of the individual in the population and NP is the total population size. This approach allows low-performing individuals to experience more significant changes in their dimensions, fostering deeper and more personalized learning.

2.3 The escape or escape algorithm

The Escape or Escape Algorithm (ESC) created by Ouyang et al. [28], based on human behavioral segmentation during emergency evacuations (Calm, Herding, Panic), the metaheuristic algorithm design balances exploration and exploitation as it classifies solutions using human behavioral segmentation as follows:

- Calm: rational individuals who guide the group, and the behavior model is expressed as follows in Eq.(8):

$$x_{i,j}^{new} = x_{i,j} + m_1 \cdot (w_1 \cdot (C_j - x_{i,j}) + v_{c,j}) \cdot P(t) \tag{8}$$

where C_j denotes the group's collective decision and serves as the center of the calm group along the $j-th$ dimension. The $v_{c,j}$ vector is given by Eq. (9).

$$v_{c,j} = R_{c,j} - x_{i,j} + \epsilon_j \tag{9}$$

where $R_{c,j}$, denoting a random position in the quiet group, see in Eq. 10. Values $r_{min,j}^c$ and $r_{max,j}^c$ indicate the minimum and maximum $J-th$ dimension. The term ϵ_j introduces a small motion correction and m_1 regulates the amount of such movement.

$$R_{c,j} = r_{min,j}^c + r_{i,j} \cdot (r_{max,j}^c - r_{min,j}^c) \tag{10}$$

- Herding: They follow other individuals without a clear direction, and the behavior model is expressed in Eq. (11):

$$x_{i,j}^{new} = x_{i,j} + m_1 \cdot \Big(w_1 \cdot (C_j - x_{i,j}) + m_2 \cdot w_2 \cdot (x_{p,j} - x_{i,j}) + v_{h,j} \cdot P(t)\Big) \tag{11}$$

where $x_{p,j}$ represents a solution randomly selected from the panic group, w_2 is an adaptive Levy weight. $v_{h,j}$ as calculated with Eq.(12), $R_{h,j} = r_{min,j}^h + r_{i,j} \cdot (r_{max,j}^h - r_{min,j}^h)$ denotes generated random position, $r_{min,j}^h$ and $r_{max,j}^h$ are the minimum and maximum parameters of the $j-th$ dimensions, m_1 and m_2 are a binary variables.

$$v_{h,j} = R_{h,j} - x_{i,j} + \epsilon_j \tag{12}$$

- Panics: They move erratically, exploring unconventional exits, and the behavior model is expressed as follows in Eq.(13):

$$x_{i,j}^{new} = x_{i,j} + m_1 \cdot \Big(w_1 \cdot (E_j - x_{i,j}) + m_2 \cdot w_2 \cdot (x_{rand,j} - x_{i,j}) + v_{p,j} \cdot P(t)\Big) \tag{13}$$

where E_j represents an individual random from the Pool Elite, x_{rand}, j denotes any individual in

the population. The value of $v_{p,j}$ is defined by Eq. (14):

$$v_{p,j} = R_{p,j} - x_{i,j} + \epsilon_j \tag{14}$$

where the $R_{p,j} = r^p_{min,j} + r_{i,j} \cdot (r^p_{max,j} - r^p_{min,j})$ is a randomly generated position within the panic group's bounds.

Each x_i in the population is randomly initialized in domain of solutions as follows in Eq.(15):

$$x_{i,j} = lb_j + r_{i,j} \cdot (ub_j - lb_j), \quad r_{i,j} \sim U(0,1) \tag{15}$$

where lb_j and ub_j represent the lower and upper limits of the $j-th$ dimension, $r_{i,j}$ denotes a value in the interval 0 and 1, showing the randomness in the initial decision-making process during an evacuation. After initializing the population, the objective function value of $f_i = f(x_i)$ is calculated. The top individuals are stored in the Elite Pool E as follows in Eq.(16):

$$E = X_1, X_2, \ldots, X_{exist} \tag{16}$$

Individuals elite represent the potential best solutions (output) identified by the current population and become a reference for future iterations.

The panic parameter models the transition in crowd behavior at the beginning of each iteration t. The panic parameter follows Eq.(17):

$$P(t) = \cos\left(\frac{\pi t}{6T}\right) \tag{17}$$

When the panic parameter is higher, the individuals exhibit more disorder behavior. This parameter loses its value over time as t goes from 0 to iterations T. In the exploration phase, individuals are considered "calm" and move toward Elite Pool solutions as follows, Eq.(18):

$$\begin{aligned} x^{new}_{i,j} = {} & x_{i,j} + m_1 \cdot w_1 \cdot (E_j - x_{i,j}) \\ & + m_2 \cdot w_2 \cdot (x_{rand,j} - x_{i,j}) \end{aligned} \tag{18}$$

where $x_{i,j}$ denotes the position in $i-th$ individual in $j-th$ dimension, E_j is a position from Elite Pool, $x_{rand,j}$ is a random position in the population. The size step of individuals are controlled by Levy weights adaptive strategy, as shown in Eq.(19) and Eq.(20).

$$w_j = \frac{|\mu_j|}{|\nu_j|^{1/\beta}}, \quad \mu_j \sim N(0,\sigma^2),\ \nu_j \sim N(0,1) \tag{19}$$

$$\sigma = \left(\frac{\Gamma(\frac{1+\beta}{2}) \cdot \beta \cdot 2^{\frac{1}{\beta}-1}}{\Gamma(1+\beta) \cdot \sin(\frac{\pi \cdot \beta}{2})}\right)^{\frac{1}{\beta}} \tag{20}$$

where β denotes a dynamic parameter that models its behavior through Eq. (21).

$$\beta(t) = \beta_{\text{base}} + 0.5 \cdot \sin\left(\frac{\pi t}{2T}\right) \tag{21}$$

where β_{base} is the initial value of β, this parameter is found empirically and allows for a larger exploration phase at the beginning of the algorithm.

2.4 The moss growth optimization

The Moss Growth Optimization (MGO), created by Zheng et al. [29], a novel metaheuristic algorithm based on the biological behavior of moss. MGO combines exploration and exploitation by applying strategies observed in biological environments; the main strategies are extender spore, dual propagation, and cryptobiosis. These three properties seen in the moss growth model the behavior of the algorithm as follows:

- **Spore dispersion:** under steady and turbulent wind conditions, simulates global exploration. This behavior simulates random dispersion using long and short steps following Eq.(22)

$$D_{wind} = \frac{1}{\text{num}} \sum_{i=1}^{num} dM_i \in dirX \tag{22}$$

where D_{wind} denotes the angle wind and has the identical dimension as the members in the population. The variable num represents all the individuals in Eq. (23).

$$dirX = \{M_{best} - M_i \,|\, M_i \in divX\} \tag{23}$$

where $dirX$ are the separation between individuals within $divX$ concerning M_{best}. This strategy simulates the natural dispersion of the wind. To achieve this, a strategy of long and short steps is applied through the academic modeling Eq.(24)

$$M^{new}_i = \begin{cases} M_i + step_1 \cdot D_{wind}, & r_1 > d_1 \\ M_i + step_2 \cdot D_{wind}, & r_1 \leq d_1 \end{cases} \tag{24}$$

where M^{new}_i represents the new moss acquired by the dispersal of the member, M_i, r_1 denotes any number in (0,1), d_1 is a constant obtained

experimentally, $step1$ represents the distance traveled by the spore under constant wind conditions are in Eq.(25) and $step2$ represents the space traveled by the spore under rough wind conditions following the Eq.(26),

$$step1 = w \cdot (r_2 - .5) \cdot E \tag{25}$$

where w is a constant obtained experimentally, r_2 is a random vector in the interval (0,1), and E is the force of the wind.

$$step2 = 0.1 \cdot w \cdot (r_3 - 0.5) \cdot E \cdot \left[1 + \frac{E}{2} \cdot \left(1 + \tanh(\beta\sqrt{1-\beta^2})\right) \right] \tag{26}$$

where r_3 is a random vector in the interval (0,1) and β represents the proportion of the population

- **Dual propagation search** introduces new solutions based on local reproduction as shown in Eq.(27)

$$M_i^{new} = \begin{cases} (1 - act) \cdot M_i + act \cdot M_{best}, & \text{si } r_4 > d_2 \\ M_{best,j} + step_3 \cdot D_{wind,j}, & \text{si } r_4 \leq d_2 \end{cases} \tag{27}$$

where M_i^{new} represents the new individual in ith, $M_{i,j}^{new}$ represents the jth particle in M_i^{new} and j is a random number that does not exceed the maximum dimension of the individual, M_{best} the best individual solution found, $M_{best,j}$ represents the jth particle in M_{best}, D_{wind}, j is the jth particle in D_{wind}, r_4 is a random number in the interval (0,1), d_2 is a constant parameter obtained experimentally, act is expressed in Eq.(28) and $step3$ in Eq.(29).

$$act = \begin{cases} 1, \frac{1}{1.5 - 10 \cdot r_5} \geq 0.5 \\ 0, \frac{1}{1.5 - 10 \cdot r_5} < 0.5 \end{cases} \tag{28}$$

where r_5 is a random vector in (0,1).

$$step3 = 0.1 \cdot (r_6 - 0.5) \cdot \tag{29}$$

where r_6 is any number between the range (0,1) and E is the power of the wind.

- **Cryptobiosis mechanism** is the ability of moss to recover and thrive after a period of drought in nature; the moss at this stage starts a period of metabolic latency; when the circumstances are favorable with more water, the moss can revive; this biological survival strategy of moss is transferred to the MGO algorithm by storing up to ten previous versions of each individual, this strategy is represented in Eq. (30)

$$M_i \leftarrow r M_i^{\text{best}} \tag{30}$$

where the current individual M_i is replaced by its best historical version rM_i^{best}.

2.5 The mirage search optimization

The Mirage Search Optimization (MSO), created by Jiahao He et al. [30], is a novel metaheuristic technique that is inspired by atmospheric optical phenomena and mirages. Mirages are common natural phenomena that occur because of the refraction of light as a result of temperature gradients in the atmosphere. These temperature changes cause movement and change the density of the air, modifying the refractive index. Refraction occurs when light bends as it passes through different air temperatures. This causes optical aberrations because the human brain interprets light as coming in a straight line, creating a distorted image of the environment. Mirages are classified into two types:

- Superior mirages are generated when hot air is placed above layers of cold air in the natural environment. These physical phenomena allow the observation of distant objects and are common in cold environments on the planet.
- Inferior mirages are generated when warm air is near the surface and cold air is in the upper layers. A common example of this phenomenon is on the roads on hot days, where these types of mirages generate inverted and magnified images.

These phenomena have been extensively studied by extracting refractive indices. These experiments have led to the development of mathematical formulas that model these phenomena. These mathematical models are used in the MSO algorithm to guide the algorithm's behavior during the iterative process.

2.5.1 MSO mathematical foundations

The mathematical modeling of the algorithm is divided into two sections: the upper mirages are the global optimization strategy, and the lower mirages are the local optimization strategy.

2.5.2 Initialization

The first set of solutions is used to observe, and based on these observations of the initial positions, the new

positions are updated based on the upper or lower mirage, as the case may be. Each observation position is a vector of n as shown in Eq.(31)

$$X = \begin{bmatrix} x_{11} & \dots & x_d \\ \vdots & \ddots & \vdots \\ x_{p1} & \dots & x_{pd} \end{bmatrix} \tag{31}$$

where p is the upper limit and d is the upper limit of the dimension.

The generation of vectors for each individual is defined in Eq.(32)

$$x_i = lb_i + r \cdot (ub_i - lb_i), i = 1, 2, \dots d \tag{32}$$

where x_i is the i dimension of the current individual, lb_i is the lower bound, ub_i is the upper bound and r is a random number between 0 and 1.

2.5.3 Superior mirage strategy (global exploration)

The superior mirage strategy is divided into 3 cases, depending on how the light and temperature interact to generate the superior mirage in natural environments. The equations that model the physical phenomenon were modeled and combined, resulting in Eq.(33), which updates the initial positions.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \Delta x_{lower} \tag{33}$$

where $x_{i,j}^{t+1}$ is the new updated position of each individual, $x_{i,j}^{t}$ is the current position of each individual and Δx_{lower} is selected as appropriate.

- Case 1, expressed in Eq.(34), when the condition is satisfied that the incident light is to the left of the horizontal datum normal.

$$\Delta x_{lower} = \alpha \cdot \frac{\sin(\pi - 2\alpha - 2\beta) \cdot h \sin(\frac{\pi}{2} + \beta)}{\sin(\alpha - \beta) \cdot \sin(\alpha - 2\beta)} \tag{34}$$

- Case 2, expressed in Eq.(35), when the incident light conditions are to the right of the horizontal datum normal and $\beta < \alpha < \pi/2$ are satisfied.

$$\Delta x_{lower} = \alpha \cdot \frac{\sin(\pi - 2\alpha + 2\beta) \cdot h \sin(\frac{\pi}{2} - \beta)}{\sin(\pi - \alpha + \beta) \cdot \sin(\alpha - 2\beta)} \tag{35}$$

- Case 3, expressed in Eq.(36), when determining that the incident light is to the right of the horizontal datum normal and satisfying $\beta < \alpha < \pi/2$ is satisfied.

$$\Delta x_{lower} = \alpha \cdot \frac{\sin(-\pi + 2\alpha - 2\beta) \cdot h \sin(\frac{\pi}{2} - \beta)}{\sin(\pi - \alpha + \beta) \cdot \sin(\pi - \alpha + 2\beta)} \tag{36}$$

2.5.4 Inferior mirage strategy (local exploration)

The lower mirage strategy, due to its ability to enlarge images distorted by the effect of temperature and air, has excellent scanning performance, modeling Eq.(37)

$$x_{ij}^{t+1} = x_{ij}^{t} + D\Delta x_{upper} \tag{37}$$

where there are 2 possible cases for selecting parameter D as follows:

- Case 1, when the individual present is not the most optimal individual in the population, and the following parameters are in Eq.(38)

$$\begin{aligned} h_{ij} &= |gbest_j - x_{ij}| \cdot rand \\ D &= \frac{(gbest_j - x_{ij}) \cdot rand}{h_{ij}} \end{aligned} \tag{38}$$

- Case 2, when the present individual is an optimal individual in the population and the following parameters are in Eq.(39)

$$\begin{aligned} h_{ij} &= |\pm 0.05 \cdot rand| \\ D &= \frac{\pm 0.05 \cdot rand}{h_{ij}} \end{aligned} \tag{39}$$

and Δx_{upper} is modeled on Eq.(40)

$$\Delta x_{upper} = \frac{h}{\tan(\ell)} - \frac{(\frac{h}{\sin(\ell)} - \frac{h \cdot \sin(\phi)}{\cos(\phi + \ell)} \cdot \cos(\omega))}{\cos(\omega - \ell)} \tag{40}$$

2.5.5 Applications and resources

The MSO has proven to be a competitive algorithm for optimization in complex search spaces. Its robustness has been tested with CEC2017, CEC2014, and 21 classic benchmarking functions. It demonstrated its robustness by being in first place in the Friedman ranking, with 86 optimal results out of 116 in CEC2017 (100D). It showed good performance in unimodal and hybrid functions and gave worse results using multimodal functions.

2.6 The nutcracker optimization algorithm

The Nutcracker Optimizer Algorithm (NOA) [31] is an optimization algorithm inspired by the behavior of the nutcracker bird, which follows an annual strategy for collecting, storing, and recovering seeds. This biological pattern is modeled in two fundamental stages: the search and storage strategy (during summer and fall) and the cache search and recovery strategy (in winter and spring). The first of these

strategies, which constitutes the beginning of the optimization process, is detailed below.

During the initial phase, agents act individually in the collection zone, which represents the region of the search space where new solutions are explored. Each individual starts from a randomly generated position and seeks to move toward promising regions of the space. This process simulates the behavior of a nutcracker when exploring a forest in search of seeds. If it finds a suitable source, it records it; otherwise, it moves to another location. The following equation models this active exploration behavior as follows in Eq(41):

$$\overrightarrow{X}_i^{t+1} = \begin{cases} X_{i,j}^t, & \text{if } \tau_1 < \tau_2, \\ X_{m,j}^t + \gamma \cdot \left(X_{A,j}^t - X_{B,j}^t\right) & \\ +\mu \cdot \left(r^2 \cdot U_j - L_j\right) & \text{if } t \leq \frac{T_{max}}{2}, \\ X_{C,j}^t + \mu \cdot \left(X_{A,j}^t - X_{B,j}^t\right) & \\ +\mu \cdot \mathbf{1}(r_1 < \delta) \cdot \left(r^2 \cdot U_j - L_j\right), & \text{otherwise}, \end{cases} \tag{41}$$

where X^{t+1} i is the new position of the ith nutcracker in the current generation t; $X_{i,j}^t$ is the jth position of the ith nutcracker in the current generation; U_j and L_j are vectors, including the upper and lower bound of the jth dimension in the optimization problem; γ is a random number generated according to the levy flight; $X_{best,j}^t$ is the jth dimension of the best solution obtained even now; A, C, and B are three different indices randomly selected from the population to facilitate exploration of a high-quality food source; τ_1, τ_2, r, and r_1 are random real numbers in the range of $[0, 1]$; $X_{m,j}^t$ is the mean of the jth dimensions of all solutions of the current population in the iteration t; and μ is a number generated based on the normal distribution (τ_4), levy-flight (τ_5), and randomly between zero and one (τ_3) as shown in the following Eq.(42).

$$\mu = \begin{cases} \tau_3, & \text{if } r_1 < r_2 \\ \tau_4, & \text{if } r_2 < r_3 \\ \tau_5, & \text{if } r_1 < r_3 \end{cases} \tag{42}$$

where r_2 and r_3 are random real numbers in the range of $[0, 1]$.

This structure allows the algorithm to dynamically switch between local and global exploration dynamically, thereby increasing the coverage of the search space and reducing the risk of getting stuck in local optima. The variability introduced by μ also modulates the magnitude of the exploratory steps.

Once individuals identify potentially useful regions, the next step is the storage phase, which involves burying seeds in strategic locations. At this stage, the aim is to intensify exploitation around the best solutions found, guiding agents towards high-yield areas. The following Eq.(43) models this behavior:

$$\vec{X}_i^{t+1(new)} = \begin{cases} \vec{X}_i^t + \mu \cdot (\vec{X}_{\text{best}}^t - \vec{X}_i^t) \cdot |\lambda| & \\ \quad +r_1 \cdot (\vec{X}_A^t - \vec{X}_B^t), & \text{if } \tau_1 < \tau_2 \\ \vec{X}_{\text{best}}^t + \mu \cdot (\vec{X}_A^t - \vec{X}_B^t), & \text{if } \tau_1 < \tau_3 \\ \vec{X}_{\text{best}}^t \cdot l, & \text{otherwise} \end{cases} \tag{43}$$

where $\vec{X}_i^{t+1}$ represents the new position of the particles in the iteration $t+1$, $\vec{X}_i^t$ is the current position, $\vec{X}_{best}^t$ is the best solution found so far, $\vec{X}_A^t$ and $\vec{X}_B^t$ are randomly selected positions from the population, μ is a Levy-generated step size, λ is a Levy coefficient, l is a lineary decreasing factor (from 1 to 0), τ_1, τ_2, τ_3 and r_1 are random numbers in [0,1].

This model promotes more accurate convergence by progressively reducing the search range while maintaining the capacity for variation through the use of random population differences. The decision to accept or reject this new stored position is defined in Eq .(44):

$$\vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^{t+1(\text{new})}, & \text{if } \phi < Pa_1 \\ \vec{X}_i^t, & \text{otherwise} \end{cases} \tag{44}$$

where ϕ is a random number between zero and one, and Pa_1 represents a probability value that is linearly decreased from one to zero based on current generation. Finally, this first strategy culminates in the use of spatial memory. Once the seeds have been stored, the agents simulate the behavior of the nutcracker, which memorizes the location of its reserves using spatial references in the environment.

2.7 The polar lights optimizer

2.7.1 Overview

The *Polar Lights Optimizer* (PLO) is a novel physics-inspired metaheuristic algorithm introduced by Yuan et al. [32]. Inspired by the aurora phenomenon, PLO models the movement of high-energy particles under geomagnetic forces, combining strategies of "gyration motion", "aurora

oval walk", and "particle collision" to achieve a balance between exploration and exploitation.

PLO has demonstrated superior performance against classic and state-of-the-art optimizers on IEEE CEC2014 and CEC2022 benchmarks, and has found practical use in tasks such as multi-threshold image segmentation and feature selection, particularly in medical datasets [32, 33].

2.7.2 *Mathematical modeling of PLO operators*

- **Initialization**
  A population of N particles $X = \{x_1, x_2, \ldots, x_N\}$ is initialized randomly:

$$x_i^j = x_{\min}^j + r \cdot \left(x_{\max}^j - x_{\min}^j\right), \quad r \sim \mathcal{U}(0,1) \tag{45}$$

- **Gyration motion (local exploitation)**
  Simulating spiraling under Lorentz force and atmospheric damping:

$$v_i^{t+1} = v_i^t + \frac{qB}{m}\Delta t - \alpha v_i^t \tag{46}$$

  with $q = B = m = 1$, $\alpha \sim \mathcal{U}(1, 1.5)$, and $m = 100$; velocity is used to update positions locally [32].

- **Aurora oval walk (global exploration)**
  Is inspired by the phenomenon caused by energetic particles gradually converging on the poles and forming a luminous elliptical ring, the natural behavior of the energetically charged particles within the ring are free to move freely around the oval ring, this behavior of all particles causes constant collisions between the same particles, in addition the particles chaotically follow the changes in the Earth's magnetic field, this strategy of movement in the particles is mathematically introduced in the PLO with the following Eq.(47)

$$x_i^{t+1,j} = x_i^{t,j} + w_1 \cdot \text{Levy}(d) + w_2 \cdot \left(\bar{x}^j - x_i^{t,j}\right) + w_3 \cdot \left(ub^j - lb^j\right) \cdot r \tag{47}$$

$$w_1 = 1 - \frac{t}{T}, \quad w_2 = \frac{t}{T} \tag{48}$$

  where $\bar{x}$ is the population mean, t is current iteration, T is max iterations [32, 33].

- **Combined position update**
  Positions are updated by blending local and global strategies:

$$x_i^{t+1} = u \cdot \text{Gyration}(x_i^t) + (1-u) \cdot \text{OvalWalk}(x_i^t) \tag{49}$$

  with random $u \in [0, 1]$ [32].

- **Particle collision (stagnation avoidance)**
  Random pairwise collision to prevent local trapping:

$$x_i^{t+1,j} = x_i^{t,j} + r_1(x_i^{t,j} - x_k^{t,j}) + r_2 \sin(x_i^{t,j} - x_k^{t,j}) \tag{50}$$

  where $k \neq i$ is a random particle and $r_1, r_2 \in [0, 1]$. Collision probability increases over iterations [32].

2.7.3 *Algorithmic framework*

At each iteration, PLO performs:

1. Gyration motion update.
2. Aurora oval walk update.
3. Stochastic collision with another particle.
4. Objective function evaluation and archive of the best solution.

This continues until $t = T$ is satisfied.

2.7.4 *Applications and resources*

PLO has been validated on benchmark functions (IEEE CEC2014, CEC2022) and applied to "multi-threshold image segmentation" and "feature selection" in medical imaging. MATLAB code is openly available in the next repository:

- MATLAB toolbox: https://github.com/aliasgharheidaricom/Polar-Lights-Optimizer-Algorithm-and-Applications-in-Image-Segmentation-and-Feature-Selection

2.8 Minimum cross-entropy for thresholding

Minimum cross entropy for thresholding was created by Li et al. [19]. Minimum cross-entropy threshold selection created by Brink et al. [34] mentions the importance of selecting the correct thresholds for image segmentation; having errors in the initial thresholding stage would propagate to later stages of any process in which this technique is being used. Traditionally, methods such as Otsu have been used for their robustness in contexts where specific statistical assumptions are met. However, when images have distributions that present significant overlaps in the gray levels of objects and the background, unequal variations, or the underlying probabilistic model of the image is unknown, it can introduce biases in the estimation of the thresholds. Minimum Cross-Entropy (MCE) is another alternative to optimize segmentation.

2.8.1 Principle of cross entropy

Cross entropy, introduced by Kullback [35], measures the discrepancy between two probability distributions $P = p_1, p_2, \ldots, p_N$ and $Q = q_1, q_2, \ldots, q_N$. Cross entropy is modeled by the following Eq.(51)

$$D(Q, P) = \sum_{k=1}^{N} q_k \log \frac{q_k}{p_k} \tag{51}$$

where $D(Q, P)$ quantifies the loss of information when using Q to approximate P, in terms of image segmentation where P is the original image and Q is its thresholded version, the minimum cross entropy method can be seen as an extension of cross entropy by establishing initial estimates for all p_N when no prior information about the distributions is available.

3 Proposed approach

This section describes the methodology used for the approach proposed in this manuscript. Six novel evolutionary strategies were employed for the comparative analysis due to their recent publication. These strategies are: the Differentiated Creative Search (DCS) [39], the Escape Algorithm (ESC) [28], the Moss Growth Optimization (MGO) [29], the Mirage Search Optimization (MSO) [30], the Nutcracker Optimization Algorithm (NOA) [31], and the Polar Lights Optimizer (PLO) [32]. The development of the methodology involves adapting these algorithms to minimize the objective function values (in this case, the minimum cross-entropy for thresholding).

The methodology used to find each optimal Th is shown in Figure 1.

3.1 Problem definition

The Multilevel threshold image segmentation problem can be expressed mathematically as follows: the processed image is represented by the letter I, which has $K + 1$ classes. The main objective of multilevel threshold image segmentation problems is to find the optimal value k to divide I into k classes (i.e., $C_k, k = 1, 2, \ldots, k$), therefore, the definition of the optimization problem can be expressed as shown in Eq.(52):

$$\begin{aligned} C_0 &= I_{ij} \mid 0 \leq I_{ij} \leq t_1 - 1, \\ C_1 &= I_{ij} \mid t_1 \leq I_{ij} \leq t_2 - 1, \\ C_K &= I_{ij} \mid t_K \leq I_{ij} \leq L - 1 \end{aligned} \tag{52}$$

where L represents the maximum grayscale value of I and $t_1, t_2, ..., t_k$ represent the optimal thresholds found by the implemented metaheuristic algorithm.

3.1.1 Objective function

Data are often correlated in image segmentation, so treating image information as a coherent and dependent set is more appropriate. However, methods based on maximum entropy or cross entropy traditionally use only the gray-level histogram, treating each pixel as an independent experiment and ignoring the spatial correlation between them. In the minimum cross entropy segmentation, the image $f(x, y)$ is modeled as a positive distribution of gray levels, and the segmented image $g(x, y)$ is defined by a threshold t as expressed in Eq.(53)

$$g(x, y) = \begin{cases} \mu_1, & f(x, y) < t \\ \mu_2, & f(x, y) \geq t, \end{cases} \tag{53}$$

where μ_1 represents the group averages below the threshold t and μ_2 represents in the Eq.(54), the intensity averages for the groups that exceed or are equal to the threshold t.

$$\mu_1(t) = \frac{\sum_{j=1}^{t-1} j h_j}{\sum_{j=1}^{t-1} h_j}, \quad \mu_2(t) = \frac{\sum_{j=t}^{L} j h_j}{\sum_{j=t}^{L} h_j} \tag{54}$$

The cross entropy between $f(x, y)$ and $g(x, y)$ is expressed in the following Eq.(55).

$$\eta(t) = \sum_{j=1}^{t-1} j h_j \log(\frac{j}{\mu_1(t)}) + \sum_{j=t}^{L} j h_j \log(\frac{j}{\mu_2(t)}) \tag{55}$$

The optimal threshold is selected based on the following Eq.(56)

$$t_0 = \min_t(\eta(t)) \tag{56}$$

where t_0 is the required threshold, $\eta(t)$ is the cross-entropy.

4 Experimental study and analysis of results

This section presents the analysis of results obtained from experiments conducted to compare the different evolutionary strategies used in this manuscript. The section also presents the conditions of the experiments, the metrics used, and the qualitative and quantitative results obtained. Conclusively, the section undertakes a discussion of these results.

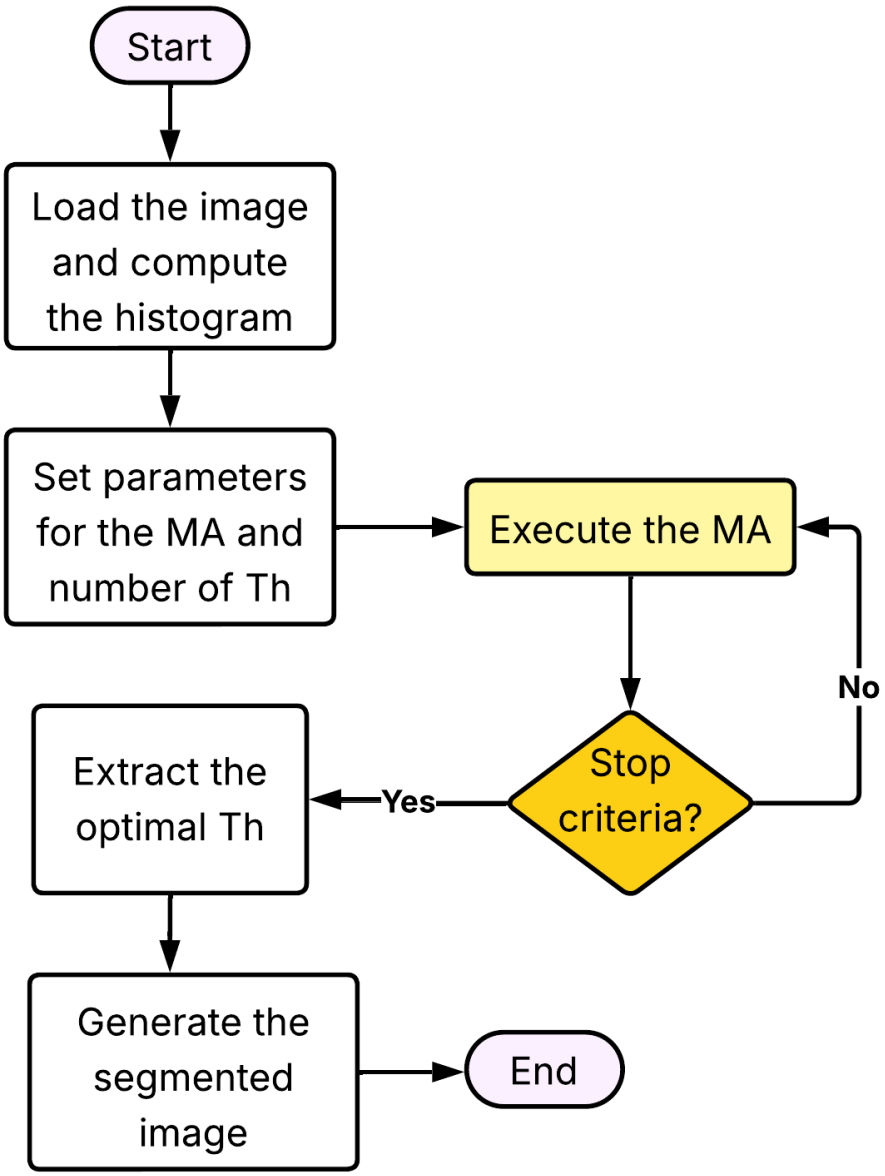

Figure 1. Generic flowchart of the proposed methodology for multilevel thresholding.

4.1 Experimental setup and dataset

The experiments were intentionally designed to provide a comprehensive evaluation of each algorithm; some complex images were selected from the Berkeley Segmentation Dataset 300 (BSD300) [38]. The BSD300 dataset has been frequently employed in state-of-the-art research to evaluate the performance of MAs implemented for image segmentation, edge detection, and other related applications. This dataset was proposed in 2001 by the University of Berkeley to provide a set of reference images that could be used to evaluate various MAs in terms of their performance in the segmentation process.

For experimental purposes, a set of nine representative images was meticulously selected for analysis, as shown in Figure 2: Starfish, Firefighters, Pyramid, Airplane, Owls, Bird, Ocean, Snow, and Kangaroo. The images in the dataset were selected to reflect the diversity, complexity, and key characteristics of the dataset. The dataset contains a variety of scenes and visual conditions, including lighting, textures, and colors. All algorithms were executed under identical conditions, with 30 runs and 1,000 iterations for each image in the dataset. The thresholds employed ranged from 2 to 5.

The experimental tests were conducted utilizing MATLAB 2024a programming software on a Windows 11 operating system. The computer is equipped with an AMD Ryzen 7 3700x processor, 8 cores, 32 GB of RAM, and an RTX 5070 graphics card. The configuration parameters for each algorithm were obtained from the original publications based on the

descriptions provided by each author. These details can be seen in Table 1. The general parameter settings are provided in the final column of the table. Np is the total population size, and $MaxIter$ is the maximum number of internal iterations for each algorithm.

Table 1. Parameter settings for experimental results.

Algorithm	Parameters
PSO	$\omega = .9$, $c_1 = 1.3$, $c_2 = 1.9$
DCS	$GoldenR = \frac{2}{1+\sqrt{5}}$, Pc=.5
ESC	$Prob. = .5$, $Elite = 5$, $Beta = 1.5$
MGO	$w = 2$, $DivNum = \frac{dim}{4}$
MSO	$SupMirage = [31, 34]$
NOA	$Alpha = .05$, $Pa2 = .2$, $Prob. = .2$
PLO	$w1 = [1.6E^{-11}, 1E^{-4}]$, $w2 = [1, 0.992]$
General parameters: $Np = 50$, $MaxIter = 500$	

4.2 Evaluation metrics

Since the image segmentation process involves reducing and simplifying the presentation of images for better analysis and processing, there is a possibility that significant information may be lost. Therefore, it is necessary to evaluate whether this process is conducted most effectively. To this end, some metrics [43] evaluate the quality of the image after it has been processed [41]. The metrics provided valuable information about the performance of image segmentation techniques. The metrics employed in this study included the Peak Signal-to-Noise Ratio (PSNR) [36, 40], the Structural Similatiry Index method (SSIM) [42], and the Feature Similarity Index method (FSIM) [37].

The PSNR uses the Root Mean Square Error (RMSE) to compare the pixel information in the original image and the segmented image, as shown below:

$$PSNR = 20 \log \frac{255}{RSME} \tag{57}$$

where:

$$RMSE = \sqrt{\frac{1}{Ro \times Co} \sum_{i=1}^{Ro} \sum_{j=1}^{Co} (I_{Gr}(i,j) - I_S(i,j))} \tag{58}$$

The size of the image is defined by Ro and Co. The original image in grayscale is denoted by I_{Gr}. The segmented image is denoted by I_S.

The Structural Similarity (SSIM) Index. SSIM compares information from the structures of two images.

$$SSIM = \frac{(2\mu I_{Gr}\mu I_S + C_1)(2\sigma I_{Gr}I_S + C_2)}{(\mu^2 I_{Gr} + \mu^2 I_S) + C_1)(\sigma^2 I_G r + \sigma_(I_S)^2 + C_2)} \tag{59}$$

In which

$$\sigma_{(I_{Gr}I_S)} = \frac{1}{Co-1} \sum_{i=1}^{Co} (I_{Gr_i} + \mu_{I_{Gr}})(I_S + \mu_{I_S}) \tag{60}$$

Finally, the feature similarity (FSIM) index measures the similarity between the segmented and original images in terms of their internal features, such as corners and borders. The FSIM is computed as follows:

$$FSIM = \frac{\sum_{\omega \in \Omega} S_{L(\Omega)} PC_{m(\omega)}}{\sum_{\omega \in \Omega} PC_{m(\omega)}} \tag{61}$$

where Ω represents the entire domain of the image, and their values are calculated on Eq. 62 defined as:

$$S_{L(\omega)} = S_{PC(\omega)} S_{G(\omega)} \tag{62}$$

$$S_{PC(\omega)} = \frac{2PC_{1(\omega)}2PC_{2(\omega)} + T_1}{PC^2_{1(\omega)}PC^2_{2(\omega)} + T_1} \tag{63}$$

$$S_{G(\omega)} = \frac{2G_{1(\omega)}G_{2(\omega)} + T_2}{G^2_{1(\omega)}G^2_{2(\omega)} + T_2} \tag{64}$$

T_1 and T_2 are constants, and the values chosen are $T_1 = 0.85$ and $T_2 = 160$. G is the magnitude of the gradient of a digital image, and PC is the phase congruence; both are defined in the following expressions:

$$G = \sqrt{G_x^2 + G_y^2} \tag{65}$$

where G_x is the magnitude of the gradient in the x direction and G_y is the magnitude of the gradient in the y direction.

$$PC(w) = \frac{E(w)}{(\varepsilon + \sum_n A_n(w))} \tag{66}$$

where $A_{n(\omega)}$ is the local amplitude on scale n, and $E_{(\omega)}$ is the magnitude of the response vector in w over n, ε is a small positive number, and $PCm(w)$ is the maximum value of $PC_1(w)$ and $PC_2(w)$ defined in the next expression:

$$PC_m(w) = (max(PC_1(w), PC_2(w))) \tag{67}$$

On the other hand, statistical analysis has recently been used to evaluate the performance of MAs.

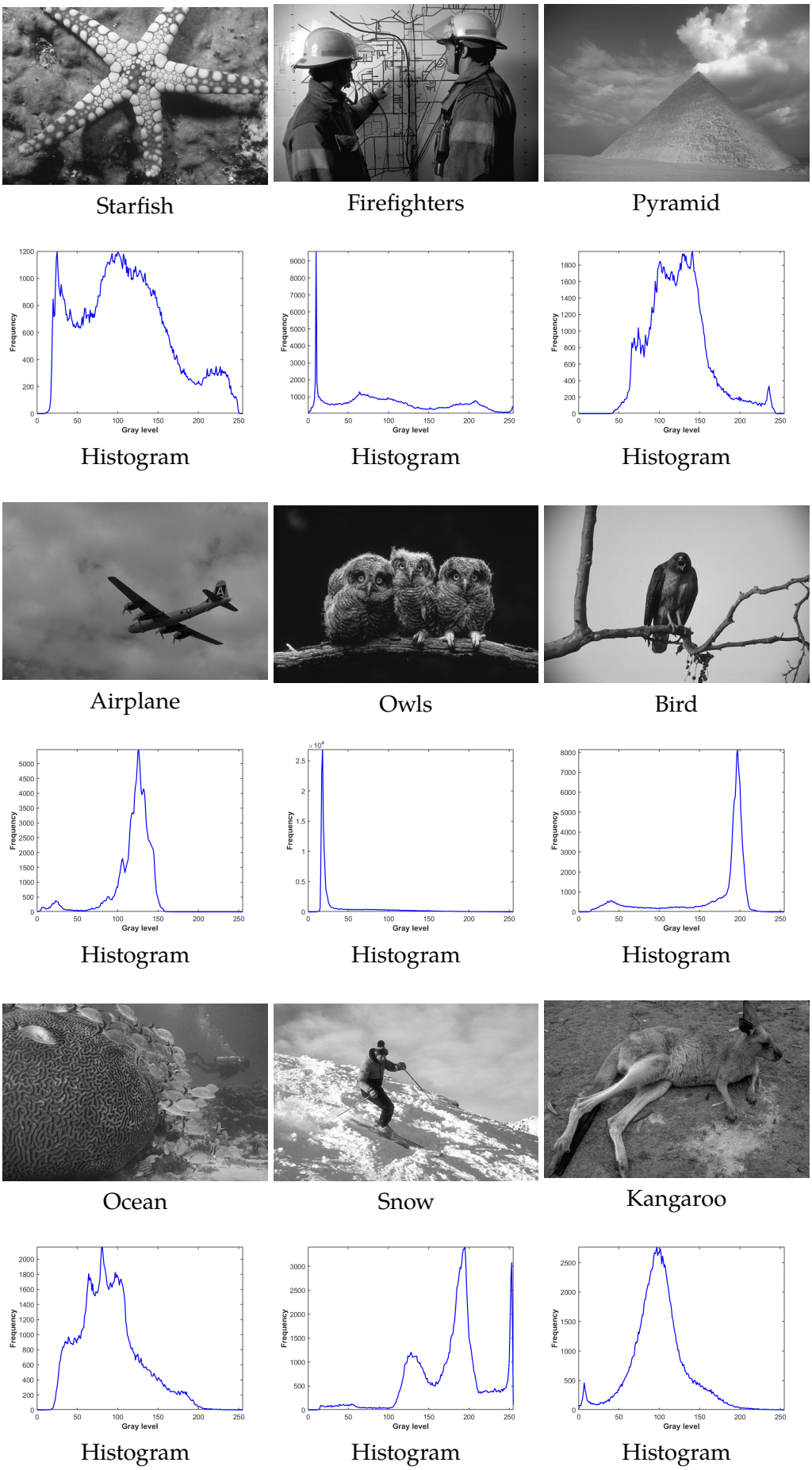

Figure 2. Set of images used for the experiments extracted from the BSD300 dataset.

Specifically, statistical analysis is necessary for fairly comparing stochastic metaheuristic techniques. One non-parametric statistical analysis method is the Friedman test, also known as Friedman's two-way ANOVA [46]. It is used to detect significant differences in the behavior of two or more algorithms. This statistical test was implemented in this manuscript to validate the experimental results of the comparison. The specific details of this approach can be found in [44, 45].

4.3 Results analysis and discussion

The experimental tests were conducted comprehensively, following the configuration and metrics previously described. The objective was to assess the performance of each algorithm on the selected dataset. The outcomes are organized in the following results sections. The objective function values are reported in Tables 2 and 3, while Table 4 provides the optimal thresholds identified for each algorithm, including all corresponding threshold values. The PSNR results are shown in Tables 5 and 6, followed by SSIM metrics in Tables 7 and 8, and FSIM outcomes in Tables 9 and 10. Each results section also includes the image names from the BSDS300 dataset used in the experiments, as well as the corresponding threshold values. The subsequent section presents a detailed analysis of the obtained results, emphasizing measures of central tendency (mean and standard deviation), with the best values highlighted in bold. Finally, Figures 3, 4, 5, 6 and 7 offer a visual comparison, illustrating both the optimal thresholds and the segmented images generated using those values.

As illustrated in Table 2, the results demonstrate that the PSO algorithm consistently achieves optimal values for the majority of threshold levels. Another similar case is that of the ESC algorithm, which shows significant values. This is particularly noteworthy given that the ESC algorithm is designed to address minimization problems, where the objective is to identify the lowest possible value. In certain cases, such as the image of the Bird, it is evident that the DCS is the optimal solution, with a margin of superiority measured in decimal points. This prompts the consideration of a second-best algorithm, which would be the DCS. A similar behavior can be observed in the snow image, in which the values differ by decimals in terms of the mean. Conversely, the NOA algorithm demonstrates an alternative behavioural pattern; it attains the maximum objective function values in comparison to all other algorithms for all images (see Table 3). It is interesting to note that the values obtained for each algorithm vary concerning each image, and the threshold value increases. This phenomenon is exemplified by the Ocean and Kangaroo images, where the variability of the results obtained by all algorithms is evident in both tables.

The optimal threshold values previously identified can be analyzed with the values presented in Table 4. This table demonstrates that, for threshold level 2, in the Starfish and Firefighters images, 69 and 138 would be considered, which are obtained by the ESC algorithm. Conversely, the NOA algorithm attains values of 121 and 59. A similar behavior can be observed in these images, with a tendency toward values that demonstrate maximum objective function values. As the threshold levels are increased, a similar behaviour of variations in the values obtained can be observed. To illustrate this, consider the Pyramid and Airplane images at threshold level 5. In this case, the DCS and ESC algorithms would maintain more stable values, while algorithms such as NOA and PLO would handle different values obtained.

The Starfish, Firefighters, Pyramid, and Airplane images show the optimal threshold values obtained for each thresholding level. Given that displaying all images with all thresholds for all algorithms would represent an overly exhaustive analysis and a large number of pages, it was decided to consider an analysis only for these images visually and only for one threshold value per image with its respective histogram, where the optimal threshold value obtained is shown with a red line. The results of this study are presented in Figures 3, 4, 5, 6 and 7. As demonstrated by the Airplane image for threshold level 5, the distinction between the DCS and NOA algorithms is readily apparent. Another intriguing case is that of the pyramid image for threshold level 4, where the MGO, NOA, and PLO algorithms exhibit marked variations in segmentation quality. For thresholds 2 and 3, the NOA algorithm is the most visually impacted in terms of segmentation quality. In contrast, the DCS and ESC algorithms demonstrate visually significant results in comparison to the other algorithms.

Table 5 shows metrics obtained by the PSO, DCS, ESC, and MGO algorithms when applied to different images from the BSD300 dataset, evaluated under different thresholds (Th = 2 to 5). PSNR is a widely used metric for quantifying the visual quality of a segmented

Table 2. Objective function value results using the BSDS300 dataset for PSO, DCS, ESC, and MGO algorithms.

Algorithm		PSO		DCS		ESC		MGO	
Image	Th	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Starfish	2	**2.27E+00**	0.00E+00	2.27E+00	6.38E-04	**2.27E+00**	1.44E-05	2.28E+00	1.83E-02
	3	**1.25E+00**	0.00E+00	1.25E+00	2.05E-03	**1.25E+00**	1.44E-05	1.29E+00	2.85E-02
	4	**7.84E-01**	8.69E-06	7.93E-01	5.35E-03	7.84E-01	1.44E-05	8.81E-01	5.36E-02
	5	**5.38E-01**	0.00E+00	5.50E-01	9.54E-03	5.38E-01	2.81E-04	6.53E-01	5.70E-02
Firefighters	2	**2.87E+00**	0.00E+00	2.87E+00	5.92E-04	**2.87E+00**	2.81E-04	2.87E+00	8.38E-03
	3	**1.51E+00**	0.00E+00	1.51E+00	1.40E-03	**1.51E+00**	2.45E-04	1.58E+00	5.85E-02
	4	**1.03E+00**	0.00E+00	1.04E+00	4.45E-03	1.03E+00	3.99E-05	1.14E+00	5.71E-02
	5	**7.29E-01**	2.31E-05	7.40E-01	6.56E-03	7.29E-01	2.45E-04	8.54E-01	4.49E-02
Pyramid	2	**9.90E-01**	0.00E+00	9.90E-01	3.39E-04	**9.90E-01**	2.43E-05	9.97E-01	9.74E-03
	3	**5.41E-01**	0.00E+00	5.43E-01	1.77E-03	**5.41E-01**	2.07E-03	5.80E-01	3.52E-02
	4	**3.49E-01**	0.00E+00	3.54E-01	2.42E-03	3.49E-01	6.06E-06	4.10E-01	3.75E-02
	5	2.53E-01	1.67E-02	2.64E-01	6.36E-03	**2.50E-01**	1.59E-04	3.26E-01	3.22E-02
Airplane	2	**5.08E-01**	0.00E+00	5.08E-01	1.48E-04	**5.08E-01**	9.54E-04	5.10E-01	3.48E-03
	3	3.24E-01	9.09E-07	3.25E-01	1.17E-03	**3.24E-01**	7.54E-07	3.48E-01	2.37E-02
	4	2.25E-01	1.72E-02	2.27E-01	2.69E-03	**2.22E-01**	8.63E-04	2.68E-01	1.69E-02
	5	1.77E-01	2.12E-02	1.78E-01	1.35E-02	**1.66E-01**	3.25E-05	2.26E-01	2.41E-02
Owls	2	**1.03E+00**	0.00E+00	1.03E+00	1.90E-04	**1.03E+00**	2.93E-04	1.03E+00	5.39E-03
	3	**5.91E-01**	0.00E+00	5.93E-01	1.44E-03	**5.91E-01**	5.17E-04	6.19E-01	2.90E-02
	4	**3.92E-01**	0.00E+00	3.97E-01	3.18E-03	3.92E-01	1.96E-05	4.42E-01	2.92E-02
	5	**2.86E-01**	3.49E-05	2.92E-01	3.19E-03	2.86E-01	8.39E-05	3.47E-01	2.47E-02
Bird	2	**7.92E-01**	0.00E+00	7.93E-01	2.97E-04	**7.92E-01**	5.25E-03	7.94E-01	1.99E-03
	3	**4.79E-01**	0.00E+00	4.80E-01	7.15E-04	4.79E-01	5.45E-06	4.94E-01	1.11E-02
	4	**3.16E-01**	0.00E+00	3.18E-01	1.24E-03	3.16E-01	4.41E-05	3.55E-01	2.10E-02
	5	**2.32E-01**	0.00E+00	2.38E-01	3.60E-03	2.32E-01	1.15E-04	2.83E-01	2.37E-02
Ocean	2	**1.33E+00**	0.00E+00	1.33E+00	4.86E-04	**1.33E+00**	2.86E-04	1.33E+00	1.40E-02
	3	**7.37E-01**	0.00E+00	7.41E-01	3.30E-03	**7.37E-01**	5.94E-03	8.04E-01	6.33E-02
	4	**4.98E-01**	0.00E+00	5.05E-01	5.29E-03	4.98E-01	4.93E-07	5.78E-01	5.02E-02
	5	**3.43E-01**	0.00E+00	3.52E-01	4.90E-03	3.43E-01	1.17E-04	4.62E-01	4.54E-02
Snow	2	**1.37E+00**	0.00E+00	1.37E+00	1.38E-04	**1.37E+00**	2.45E-01	1.37E+00	2.38E-03
	3	**5.55E-01**	0.00E+00	5.56E-01	7.96E-04	**5.55E-01**	3.41E-05	5.70E-01	1.16E-02
	4	**3.88E-01**	0.00E+00	3.93E-01	2.91E-03	3.88E-01	2.99E-05	4.29E-01	2.32E-02
	5	**3.01E-01**	6.33E-05	3.06E-01	4.23E-03	3.01E-01	1.57E-04	3.36E-01	1.84E-02
Kangaroo	2	**1.61E+00**	0.00E+00	1.61E+00	3.25E-04	**1.61E+00**	4.93E-05	1.61E+00	8.46E-03
	3	**9.02E-01**	0.00E+00	9.04E-01	2.37E-03	**9.02E-01**	2.17E-04	9.41E-01	3.47E-02
	4	5.84E-01	8.34E-06	5.92E-01	6.24E-03	**5.84E-01**	3.97E-06	6.62E-01	5.14E-02
	5	**4.18E-01**	3.22E-06	4.26E-01	4.07E-03	4.18E-01	2.17E-04	5.14E-01	4.02E-02

image concerning its original. Higher PSNR values indicate better preservation of visual information and less distortion introduced by the segmentation process. It can be observed that ESC presents the best overall performance among the evaluated algorithms, obtaining the highest PSNR values in the nine images. This behavior reflects its ability to generate accurate and high-quality segmentations in different visual contexts consistently. Meanwhile, MGO also demonstrates solid performance, particularly standing out in images such as Airplane, where it achieves competitive results across several thresholds. In contrast, DCS shows more irregular behavior, achieving good results in some specific images, but without maintaining the same consistency observed in ESC. Overall, the results suggest that ESC is the most effective algorithm in terms of preserving visual quality and maintaining stable and superior performance across a wide variety of conditions. This superiority positions it as a robust option for image segmentation applications that require high fidelity to the original content.

Table 6 shows the average PSNR values obtained by the MSO, NOA, and PLO algorithms when applied to the same set of images and thresholds. The results indicate that the NOA algorithm performs best overall, achieving the highest number of maximum PSNR values compared to MSO and PLO. This algorithm performed exceptionally well on images such as Starfish, Firefighters, Pyramid, and Ocean, where it consistently obtained higher values across

Table 3. Objective function value results using the BSDS300 dataset for MSO, NOA, and PLO algorithms.

Algorithm		MSO		NOA		PLO	
Image	Th	Mean	Std	Mean	Std	Mean	Std
Starfish	2	1.04E+02	1.04E+02	8.90E+01	9.10E+01	2.28E+00	9.65E-03
	3	1.12E+02	1.12E+02	1.24E+02	1.21E+02	1.30E+00	4.12E-02
	4	1.08E+02	1.07E+02	1.18E+02	1.35E+02	8.64E-01	6.27E-02
	5	1.09E+02	1.09E+02	1.42E+02	1.01E+02	6.60E-01	5.18E-02
Firefighters	2	8.80E+01	8.80E+01	9.10E+01	1.01E+02	2.88E+00	1.24E-02
	3	8.93E+01	8.93E+01	8.53E+01	7.93E+01	1.60E+00	7.13E-02
	4	8.30E+01	8.30E+01	9.90E+01	7.65E+01	1.14E+00	6.15E-02
	5	8.80E+01	9.48E+01	9.40E+01	7.26E+01	8.52E-01	5.30E-02
Pyramid	2	1.37E+02	1.37E+02	1.56E+02	1.38E+02	9.94E-01	5.17E-03
	3	1.29E+02	1.29E+02	1.32E+02	1.31E+02	5.74E-01	3.08E-02
	4	1.30E+02	1.30E+02	1.37E+02	1.22E+02	4.05E-01	3.47E-02
	5	1.28E+02	1.28E+02	1.16E+02	1.18E+02	3.19E-01	3.68E-02
Airplane	2	8.55E+01	8.55E+01	7.85E+01	8.55E+01	5.13E-01	6.88E-03
	3	9.60E+01	9.33E+01	1.26E+02	9.13E+01	3.59E-01	2.73E-02
	4	8.70E+01	9.68E+01	9.30E+01	6.15E+01	2.75E-01	2.75E-02
	5	8.36E+01	8.28E+01	7.92E+01	8.66E+01	2.33E-01	2.41E-02
Owls	2	7.25E+01	7.25E+01	6.70E+01	8.10E+01	1.04E+00	1.96E-02
	3	7.77E+01	7.77E+01	9.10E+01	7.30E+01	6.24E-01	2.91E-02
	4	8.18E+01	8.25E+01	8.38E+01	1.18E+02	4.50E-01	3.36E-02
	5	8.14E+01	8.14E+01	9.00E+01	1.24E+02	3.71E-01	3.68E-02
Bird	2	1.13E+02	1.13E+02	1.15E+02	1.10E+02	7.96E-01	4.27E-03
	3	1.13E+02	1.12E+02	1.18E+02	9.30E+01	4.97E-01	1.39E-02
	4	1.15E+02	1.15E+02	1.41E+02	1.08E+02	3.55E-01	2.85E-02
	5	1.04E+02	1.06E+02	1.07E+02	1.01E+02	2.75E-01	2.23E-02
Ocean	2	9.40E+01	9.40E+01	9.15E+01	8.00E+01	1.34E+00	2.36E-02
	3	9.10E+01	9.10E+01	9.87E+01	9.43E+01	7.91E-01	4.50E-02
	4	9.15E+01	9.13E+01	1.06E+02	9.63E+01	5.95E-01	6.59E-02
	5	9.70E+01	9.66E+01	1.21E+02	9.84E+01	4.71E-01	4.89E-02
Snow	2	1.26E+02	1.26E+02	1.32E+02	1.16E+02	1.37E+00	5.12E-03
	3	1.52E+02	1.52E+02	1.53E+02	1.46E+02	5.80E-01	1.59E-02
	4	1.55E+02	1.56E+02	1.33E+02	1.44E+02	4.26E-01	2.30E-02
	5	1.55E+02	1.38E+02	1.54E+02	1.59E+02	3.40E-01	1.86E-02
Kangaroo	2	9.10E+01	9.10E+01	1.01E+02	9.30E+01	1.61E+00	8.04E-03
	3	8.53E+01	8.53E+01	7.93E+01	8.70E+01	9.48E-01	3.24E-02
	4	8.58E+01	8.58E+01	7.73E+01	9.13E+01	6.76E-01	4.83E-02
	5	8.78E+01	8.82E+01	9.40E+01	8.10E+01	5.34E-01	5.36E-02

the different thresholds. This behavior suggests that the algorithm has a strong ability to preserve visual details in different segmentation scenarios. On the other hand, MSO shows competitive performance in images such as Bird and Snow, where it achieves some of the highest values at high thresholds. However, its performance is more inconsistent in other images. In contrast, PLO, although not dominating in terms of maximum values, maintains stable results close to the best performance in several images, such as Airplane, Owls, and Kangaroo, showing good overall consistency, although without reaching the same level of accuracy as NOA.

Table 7 shows the average values and standard deviations obtained using the SSIM metric for the PSO, DCS, ESC, and MGO algorithms, evaluated on nine

Table 4. The optimal threshold values for the PSO, DCS, ESC, MGO, MSO, NOA, and PLO algorithms, considering 2, 3, 4, and 5 thresholds.

Algorithm		PSO	DCS	ESC	MGO	MSO	NOA	PLO
Image	Th	Optimal Th	Optimal Th	Optimal Th	Optimal Th	Optimal Th	Optimal Th	Optimal Th
Airplane	2	69, 138	69, 139	69, 138	61, 131	69, 138	121, 59	70, 142
	3	59, 109, 167	59, 111, 166	59, 109, 167	62, 102, 173	59, 109, 167	56, 168, 104	67, 114, 159
	4	48, 84, 123, 176	51, 82, 123, 170	48, 84, 123, 176	51, 85, 129, 184	48, 84, 123, 176	140, 195, 51, 97	52, 96, 137, 195
	5	43, 73, 105, 138, 185	46, 76, 108, 136, 185	43, 73, 105, 138, 185	39, 72, 93, 130, 178	43, 73, 104, 138, 185	90, 135, 200, 59, 108	47, 77, 101, 141, 205
Starfish	2	45, 131	111, 161	111, 162	112, 163	111, 162	117, 159	112, 167
	3	34, 84, 150	93, 126, 168	92, 125, 170	98, 136, 179	92, 125, 170	107, 123, 195	91, 124, 166
	4	25, 57, 96, 156	88, 117, 139, 174	87, 114, 140, 180	88, 120, 137, 168	87, 114, 139, 179	130, 104, 1, 208	92, 125, 156, 189
	5	24, 54, 88, 131, 183	87, 109, 132, 156, 203	84, 107, 128, 152, 188	95, 120, 137, 165, 187	84, 107, 128, 151, 188	186, 120, 98, 168, 204	83, 104, 118, 152, 180
Owls	2	111, 162	64, 116	65, 117	64, 118	65, 117	50, 117	66, 119
	3	92, 125, 170	40, 88, 126	41, 88, 127	40, 88, 119	41, 88, 127	115, 35, 87	44, 94, 136
	4	87, 114, 140, 180	32, 71, 100, 136	33, 74, 101, 135	31, 71, 98, 134	31, 73, 100, 134	160, 35, 87, 109	27, 79, 102, 135
	5	84, 107, 128, 152, 189	29, 69, 90, 115, 149	29, 65, 90, 112, 144	38, 67, 88, 114, 158	29, 66, 90, 112, 144	133, 50, 83, 26, 211	12, 54, 77, 97, 124
Bird	2	55, 116	55, 115	55, 116	52, 112	55, 116	127, 46	57, 113
	3	52, 107, 129	51, 104, 127	52, 107, 129	40, 92, 119	52, 107, 129	131, 61, 108	40, 84, 129
	4	47, 93, 115, 132	42, 94, 118, 133	47, 93, 115, 132	56, 93, 115, 142	46, 93, 115, 131	89, 124, 106, 59	48, 94, 113, 135
	5	22, 53, 95, 116, 132	19, 54, 98, 119, 134	22, 53, 94, 116, 132	42, 99, 115, 131, 206	19, 50, 93, 115, 132	89, 180, 18, 65, 127	50, 100, 121, 131, 195
Ocean	2	41, 104	45, 131	45, 131	43, 126	45, 131	134, 37	45, 132
	3	34, 72, 127	34, 82, 149	34, 84, 150	33, 88, 162	34, 84, 150	80, 128, 38	26, 75, 142
	4	30, 58, 96, 146	27, 62, 98, 157	25, 57, 96, 156	38, 88, 136, 197	25, 57, 95, 155	103, 144, 28, 72	24, 69, 104, 156
	5	25, 42, 70, 105, 153	21, 50, 87, 127, 184	24, 54, 88, 131, 183	23, 44, 76, 121, 189	24, 53, 87, 130, 182	76, 23, 38, 164, 84	22, 52, 91, 133, 180
Kangaroo	2	75, 151	41, 104	41, 104	40, 104	41, 104	40, 93	41, 104
	3	59, 111, 169	32, 70, 126	34, 72, 127	35, 75, 121	33, 71, 126	92, 37, 117	34, 73, 129
	4	52, 91, 139, 181	31, 60, 98, 148	30, 58, 96, 146	33, 56, 92, 141	30, 57, 94, 144	193, 58, 122, 30	28, 68, 101, 162
	5	41, 66, 102, 144, 182	27, 43, 73, 106, 159	26, 45, 74, 108, 154	29, 51, 99, 144, 168	26, 45, 73, 108, 155	71, 38, 19, 111, 95	28, 44, 95, 145, 205
Snow	2	70, 118	76, 150	75, 151	75, 150	75, 151	76, 155	75, 152
	3	57, 89, 129	60, 112, 170	59, 111, 169	59, 122, 171	58, 109, 168	138, 170, 59	54, 107, 164
	4	51, 76, 101, 138	50, 88, 138, 178	52, 91, 139, 181	48, 86, 134, 183	52, 91, 139, 181	71, 128, 213, 164	56, 91, 137, 178
	5	49, 72, 93, 118, 153	45, 66, 108, 154, 186	41, 67, 102, 144, 182	49, 72, 118, 162, 197	41, 66, 102, 144, 182	41, 195, 173, 66, 128	44, 71, 113, 156, 178
Pyramid	2	84, 168	85, 168	84, 168	80, 168	84, 168	94, 168	91, 171
	3	82, 160, 215	83, 161, 215	82, 160, 215	92, 158, 217	82, 160, 215	72, 186, 142	68, 157, 212
	4	80, 147, 180, 218	81, 144, 178, 214	80, 147, 180, 218	86, 128, 163, 227	80, 147, 180, 218	176, 1, 207, 103	69, 144, 180, 219
	5	77, 133, 160, 187, 220	49, 99, 150, 182, 220	47, 96, 149, 181, 218	76, 133, 169, 190, 217	47, 92, 148, 180, 217	70, 26, 217, 108, 160	93, 143, 166, 180, 217
Firefighters	2	65, 117	70, 117	70, 118	69, 116	70, 118	50, 99	68, 121
	3	41, 88, 127	57, 90, 132	57, 89, 129	47, 79, 124	56, 88, 129	168, 47, 104	56, 78, 124
	4	33, 75, 102, 136	51, 79, 107, 145	51, 76, 101, 138	42, 66, 93, 129	51, 76, 101, 138	130, 63, 84, 90	37, 74, 98, 140
	5	29, 66, 90, 112, 144	50, 73, 95, 121, 159	49, 72, 93, 118, 153	50, 81, 105, 116, 142	48, 70, 92, 117, 151	125, 75, 47, 104, 26	56, 71, 88, 121, 145

Table 5. Results obtained for PSNR using the BSDS300 dataset for PSO, DCS, ESC, and MGO algorithms.

Algorithm		PSO		DCS		ESC		MGO	
Image	Th	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Starfish	2	1.52E+01	9.01E-15	1.52E+01	1.54E-02	1.52E+01	9.01E-15	1.52E+01	5.25E-02
	3	1.74E+01	0.00E+00	1.74E+01	9.49E-03	1.74E+01	0.00E+00	1.73E+01	8.62E-02
	4	1.93E+01	2.65E-03	1.93E+01	6.60E-02	1.93E+01	3.52E-03	1.90E+01	2.77E-01
	5	2.08E+01	1.80E-14	2.08E+01	9.77E-02	2.08E+01	1.32E-02	2.02E+01	3.85E-01
Firefighters	2	1.44E+01	5.41E-15	1.44E+01	3.20E-02	1.44E+01	5.41E-15	1.44E+01	1.04E-01
	3	1.70E+01	7.21E-15	1.70E+01	1.04E-01	1.70E+01	7.21E-15	1.70E+01	4.90E-01
	4	1.85E+01	3.60E-15	1.85E+01	1.73E-01	1.84E+01	2.92E-02	1.84E+01	7.82E-01
	5	2.09E+01	1.65E-02	2.07E+01	2.01E-01	2.08E+01	6.08E-02	1.99E+01	6.77E-01
Pyramid	2	1.25E+01	5.41E-15	1.25E+01	2.20E-02	1.25E+01	5.41E-15	1.25E+01	3.35E-01
	3	1.66E+01	7.21E-15	1.65E+01	2.29E-01	1.66E+01	7.21E-15	1.64E+01	9.06E-01
	4	1.80E+01	7.21E-15	1.79E+01	2.53E-01	1.80E+01	1.03E-03	1.78E+01	1.19E+00
	5	1.88E+01	1.44E-01	1.91E+01	7.85E-01	1.89E+01	8.80E-02	1.86E+01	1.46E+00
Airplane	2	1.92E+01	3.60E-15	1.93E+01	1.09E-01	1.92E+01	3.60E-15	1.92E+01	3.95E-01
	3	2.19E+01	1.24E-01	2.21E+01	3.30E-01	2.19E+01	1.03E-01	2.23E+01	1.01E+00
	4	2.53E+01	5.22E-01	2.52E+01	1.82E-01	2.54E+01	1.44E-14	2.42E+01	1.00E+00
	5	2.62E+01	3.84E-01	2.62E+01	5.86E-01	2.63E+01	2.20E-02	2.50E+01	1.04E+00
Owls	2	1.92E+01	7.21E-15	1.92E+01	7.28E-03	1.92E+01	7.21E-15	1.92E+01	4.12E-02
	3	2.07E+01	1.08E-14	2.07E+01	2.99E-02	2.07E+01	1.08E-14	2.07E+01	1.70E-01
	4	2.17E+01	3.60E-15	2.17E+01	5.64E-02	2.17E+01	3.24E-04	2.15E+01	1.74E-01
	5	2.24E+01	6.26E-03	2.24E+01	8.51E-02	2.24E+01	8.02E-03	2.21E+01	2.53E-01
Bird	2	1.52E+01	1.08E-14	1.52E+01	1.00E-01	1.52E+01	1.08E-14	1.53E+01	1.98E-01
	3	1.86E+01	3.60E-15	1.85E+01	1.62E-01	1.86E+01	2.47E-02	1.84E+01	8.37E-01
	4	2.14E+01	1.80E-14	2.12E+01	2.45E-01	2.13E+01	3.99E-02	2.06E+01	8.66E-01
	5	2.26E+01	7.21E-15	2.26E+01	3.55E-01	2.27E+01	3.08E-02	2.21E+01	9.13E-01
Ocean	2	1.66E+01	7.21E-15	1.66E+01	2.16E-02	1.66E+01	7.21E-15	1.66E+01	1.76E-01
	3	1.95E+01	7.21E-15	1.95E+01	5.42E-02	1.95E+01	7.21E-15	1.93E+01	5.15E-01
	4	2.12E+01	7.21E-15	2.11E+01	1.24E-01	2.12E+01	1.84E-03	2.07E+01	4.65E-01
	5	2.23E+01	1.08E-14	2.22E+01	1.83E-01	2.23E+01	4.04E-03	2.15E+01	7.19E-01
Snow	2	1.48E+01	3.60E-15	1.48E+01	5.58E-02	1.48E+01	3.60E-15	1.48E+01	1.46E-01
	3	1.65E+01	1.08E-14	1.66E+01	1.36E-01	1.65E+01	1.08E-14	1.65E+01	4.47E-01
	4	1.86E+01	3.60E-15	1.85E+01	2.20E-01	1.86E+01	1.00E-02	1.84E+01	7.82E-01
	5	2.04E+01	6.87E-02	2.04E+01	3.06E-01	2.04E+01	1.21E-01	2.03E+01	7.72E-01
Kangaroo	2	1.75E+01	3.60E-15	1.75E+01	4.84E-02	1.75E+01	3.60E-15	1.75E+01	1.58E-01
	3	1.97E+01	7.21E-15	1.97E+01	1.48E-01	1.97E+01	7.21E-15	1.96E+01	5.02E-01
	4	2.19E+01	2.46E-02	2.19E+01	2.00E-01	2.20E+01	1.17E-02	2.12E+01	8.58E-01
	5	2.36E+01	8.36E-03	2.35E+01	1.53E-01	2.36E+01	2.47E-02	2.26E+01	6.40E-01

images from the BSD300 dataset and with thresholds between 2 and 5. The results show that ESC ranks as the algorithm with the best overall performance, obtaining the highest average SSIM values, especially in images such as Starfish, Pyramid, Bird, and Kangaroo. This suggests that ESC is more effective at preserving the perceptual structure of the segmented images, maintaining greater structural similarity to the original image. On the other hand, MGO shows robust performance in images such as Airplane, Snow, and Ocean, where it obtains the highest SSIM values at multiple thresholds, highlighting its effectiveness in preserving details in images with high contrast or visual variability. Although MGO does not lead in the total number of combinations, it presents values close to the maximum with good stability, as evidenced by moderate standard deviations.

Table 8 shows the average values and standard deviations of the SSIM metric obtained by the MSO, NOA, and PLO algorithms, evaluated on the same set of images and thresholds as in the previous analysis. As can be seen, NOA is the algorithm with the best overall performance, obtaining the highest number of maximum SSIM values across the different images and thresholds. Its performance is particularly noteworthy in images such as Pyramid, Airplane, Bird, Ocean, and Snow, demonstrating its ability to effectively preserve the visual structure of the original image after segmentation. The MSO shows competitive performance in images such as Starfish and Kangaroo, achieving some of the best SSIM values. However, its

Table 6. Results obtained for PSNR using the BSDS300 dataset for MSO, NOA, and PLO algorithms.

Algorithm		MSO		NOA		PLO	
Image	Th	Mean	Std	Mean	Std	Mean	Std
Starfish	2	1.04E+02	1.04E+02	**1.05E+02**	1.10E+02	1.52E+01	3.94E-02
	3	**1.11E+02**	1.11E+02	1.07E+02	1.04E+02	1.73E+01	9.25E-02
	4	1.08E+02	1.08E+02	**1.31E+02**	1.45E+02	1.90E+01	2.28E-01
	5	1.09E+02	1.08E+02	**1.41E+02**	1.35E+02	2.02E+01	3.76E-01
Firefighters	2	**8.80E+01**	8.80E+01	8.45E+01	8.70E+01	1.44E+01	1.76E-01
	3	8.93E+01	8.93E+01	**9.97E+01**	8.07E+01	1.69E+01	4.09E-01
	4	8.35E+01	8.30E+01	**9.55E+01**	1.26E+02	1.85E+01	7.42E-01
	5	9.48E+01	9.60E+01	**1.10E+02**	9.56E+01	2.02E+01	8.21E-01
Pyramid	2	**1.37E+02**	1.37E+02	1.11E+02	1.21E+02	1.25E+01	2.40E-01
	3	1.29E+02	1.29E+02	**1.50E+02**	1.07E+02	1.65E+01	6.54E-01
	4	**1.30E+02**	1.29E+02	1.20E+02	1.11E+02	1.75E+01	1.06E+00
	5	**1.31E+02**	1.32E+02	1.25E+02	1.38E+02	1.84E+01	1.06E+00
Airplane	2	8.55E+01	8.55E+01	**1.00E+02**	8.85E+01	1.91E+01	6.65E-01
	3	9.47E+01	9.47E+01	**1.26E+02**	1.01E+02	2.20E+01	1.05E+00
	4	9.68E+01	9.65E+01	**1.38E+02**	1.06E+02	2.38E+01	1.10E+00
	5	8.24E+01	8.26E+01	**8.68E+01**	1.04E+02	2.49E+01	1.22E+00
Owls	2	7.25E+01	7.25E+01	**9.35E+01**	8.90E+01	1.92E+01	9.75E-02
	3	**7.77E+01**	7.77E+01	7.50E+01	8.30E+01	2.06E+01	1.38E-01
	4	8.25E+01	8.15E+01	**1.10E+02**	6.38E+01	2.15E+01	2.33E-01
	5	8.14E+01	7.48E+01	**9.10E+01**	7.04E+01	2.20E+01	3.04E-01
Bird	2	**1.13E+02**	1.13E+02	1.02E+02	1.13E+02	1.52E+01	3.44E-01
	3	**1.12E+02**	1.12E+02	8.53E+01	1.20E+02	1.85E+01	8.82E-01
	4	1.16E+02	1.16E+02	**1.18E+02**	8.15E+01	2.07E+01	9.95E-01
	5	**1.06E+02**	1.06E+02	9.64E+01	1.47E+02	2.22E+01	7.19E-01
Ocean	2	**9.40E+01**	9.40E+01	7.95E+01	1.01E+02	1.66E+01	2.01E-01
	3	**9.17E+01**	9.17E+01	8.57E+01	1.04E+02	1.93E+01	3.27E-01
	4	9.13E+01	9.15E+01	**1.13E+02**	1.04E+02	2.05E+01	6.12E-01
	5	9.58E+01	9.50E+01	**1.11E+02**	1.01E+02	2.15E+01	6.23E-01
Snow	2	1.26E+02	1.26E+02	**1.35E+02**	1.12E+02	1.48E+01	2.93E-01
	3	1.52E+02	1.52E+02	**1.60E+02**	1.69E+02	1.64E+01	7.26E-01
	4	1.56E+02	1.55E+02	**1.60E+02**	1.26E+02	1.86E+01	8.19E-01
	5	**1.54E+02**	1.36E+02	1.29E+02	1.43E+02	1.97E+01	9.90E-01
Kangaroo	2	9.10E+01	9.10E+01	**9.20E+01**	8.20E+01	1.74E+01	1.88E-01
	3	8.53E+01	8.53E+01	**9.20E+01**	8.67E+01	1.96E+01	5.91E-01
	4	8.58E+01	8.58E+01	**9.90E+01**	9.18E+01	2.13E+01	6.90E-01
	5	8.66E+01	8.78E+01	**8.80E+01**	9.02E+01	2.24E+01	7.26E-01

performance is more variable in other images, and in some instances, it falls below the value obtained by NOA. Nevertheless, it maintains very low standard deviations, which indicates high consistency between runs. The PLO is the algorithm with the fewest maximums achieved in the table, and in general, its SSIM values are lower than those of NOA and MSO. However, its behavior is stable and without drastic fluctuations, which could make it suitable for tasks that are less demanding in terms of visual fidelity but require operational stability.

Table 9 shows the average values and standard deviations obtained with the FSIM metric for the PSO, DCS, ESC, and MGO algorithms, evaluated on nine images from the BSD300 dataset and with thresholds between 2 and 5. The results show that PSO

consistently achieves the best values in several images, such as Starfish, Firefighters, and Airplane, reflecting a high capacity to preserve perceptual similarity with the original image after segmentation. The ESC algorithm also exhibits competitive behavior, especially in images such as Pyramid and Bird, where it obtains results close to the best with good stability, as observed in its low standard deviations. For its part, MGO shows isolated cases of favorable results, such as in Snow and Kangaroo, although in general, its performance is less consistent than that of PSO or ESC. Overall, PSO emerges as the algorithm with the best balance between performance and stability in this set of experiments.

Table 10 shows the FSIM results for the MSO, NOA, and PLO algorithms, evaluated on the same set of images and thresholds. In this case, NOA stands out as the algorithm with the best overall performance, achieving the most favorable values in most images and thresholds. Its superiority is evident in images such as Pyramid, Airplane, Bird, and Ocean, where it consistently achieves better results than the other algorithms. MSO also presents competitive behavior, especially in Starfish, Snow, and Kangaroo, where it obtains several of the best values, although with slightly greater variability than NOA. In contrast, the PLO algorithm tends to produce higher values, reflecting lower perceptual fidelity in the segmented images; however, its results are stable, with minor deviations, suggesting consistent but less effective performance. In summary, NOA is confirmed as the most effective algorithm in terms of FSIM, while MSO shows competitive performance, and PLO maintains stability at the expense of perceptual accuracy.

The comparison performed on the same set of representative images is shown in Table 3. All images were segmented using the PSO algorithm. This set consists of four different images, each using a different threshold (Th) to study the algorithm's behavior in various segmentation scenarios. The primary objective of this visual comparison is to establish a reference framework for PSO performance, facilitating later comparison with results obtained from other algorithms applied in segmentation, including DCS, ESC, MGO, MSO, NOA, and PLO. In this way, the table provides a first visual approach to evaluating the similarities, differences, and potential benefits of each method based on the quality of the segmentation obtained, analyzing it through a subjective visual interpretation.

Table 4 shows a comparison of the performance of several segmentation algorithms used on the same test image called "Starfish". Two thresholds were considered in this analysis, allowing each algorithm to be evaluated under different segmentation conditions. The algorithms arranged in the table were determined according to the fitness value obtained during the selection process for each threshold. According to this criterion, the order obtained was: ESC, DCS, PLO, MGO, NOA, and MSO. From a visual point of view, it is observed that the NOA algorithm has a significantly worse segmentation compared to the other images. However, it is not the one that achieved the worst performance in terms of fitness. On the other hand, the ESC algorithm showed the best overall performance, which emphasized both the visual quality of the segmentation and the fitness value. In contrast, the MSO algorithm demonstrated the worst overall performance, resulting in the least favorable among the analyzed methods.

Table 5 shows a performance comparison of several segmentation algorithms used on the same test image called "Firefighters". Three thresholds are considered in this analysis, allowing each algorithm to be evaluated under different segmentation conditions. The algorithms arranged in the table were determined according to the fitness value obtained during the selection process for each threshold. According to this criterion, the following order was obtained: ESC, DCS, MGO, PLO NOA, and MSO. From a visual and subjective perspective, it is observed that the NOA algorithm has a significantly worse segmentation compared to the other images. The thresholded image is observed to be darker than the rest, despite all images having the same number of threshold levels. The background of the object can be observed with less clarity. However, this algorithm is not the one that achieved the worst performance in terms of fitness. On the other hand, the algorithm that showed the best overall performance was ESC, in which the objects in the image are more clearly appreciated, in this case, the firefighters, and the background of the image is more distinguishable. In contrast, the algorithm that showed the worst overall performance was MSO, which resulted in the least favorable outcome among the analyzed methods in terms of fitness. However, visually, it is not a particularly unappealing image, as the content can still be appreciated.

Table 6 shows a comparison of the performance of several segmentation algorithms used on the same test image called "Pyramid." Four thresholds were

Table 7. Results obtained for SSIM using the BSDS300 dataset for PSO, DCS, ESC, and MGO algorithms

Algorithm		PSO		DCS		ESC		MGO	
Image	Th	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Starfish	2	6.28E-01	4.51E-16	6.28E-01	4.57E-04	6.28E-01	4.51E-16	6.27E-01	1.66E-03
	3	7.12E-01	2.25E-16	7.11E-01	1.01E-03	7.12E-01	2.25E-16	7.07E-01	5.52E-03
	4	7.69E-01	7.92E-05	7.67E-01	3.12E-03	7.69E-01	1.61E-04	7.57E-01	7.88E-03
	5	8.12E-01	7.89E-16	8.10E-01	3.47E-03	8.13E-01	2.48E-04	7.94E-01	1.03E-02
Firefighters	2	7.64E-01	3.38E-16	7.64E-01	7.59E-04	7.64E-01	3.38E-16	7.64E-01	1.45E-03
	3	8.14E-01	1.13E-16	8.14E-01	9.39E-04	8.14E-01	1.13E-16	8.10E-01	4.91E-03
	4	8.48E-01	3.38E-16	8.45E-01	1.92E-03	8.48E-01	3.00E-05	8.35E-01	7.13E-03
	5	8.60E-01	3.87E-05	8.58E-01	3.33E-03	8.60E-01	4.99E-04	8.52E-01	7.67E-03
Pyramid	2	7.13E-01	1.13E-16	7.13E-01	1.28E-04	7.13E-01	1.13E-16	7.12E-01	2.32E-03
	3	7.48E-01	4.51E-16	7.49E-01	2.54E-03	7.48E-01	4.51E-16	7.44E-01	7.99E-03
	4	7.75E-01	2.25E-16	7.75E-01	2.59E-03	7.75E-01	1.44E-04	7.68E-01	8.56E-03
	5	7.95E-01	3.42E-03	7.94E-01	2.72E-03	7.96E-01	3.19E-04	7.85E-01	8.55E-03
Airplane	2	8.43E-01	5.63E-16	8.45E-01	1.77E-03	8.43E-01	5.63E-16	8.45E-01	3.03E-03
	3	8.46E-01	5.94E-04	8.45E-01	2.23E-03	8.46E-01	4.93E-04	8.55E-01	1.01E-02
	4	8.58E-01	2.55E-03	8.60E-01	3.24E-03	8.59E-01	4.51E-16	8.57E-01	8.55E-03
	5	8.60E-01	2.36E-03	8.62E-01	4.29E-03	8.59E-01	2.16E-04	8.61E-01	7.75E-03
Owls	2	8.05E-01	0.00E+00	8.05E-01	1.36E-04	8.05E-01	0.00E+00	8.04E-01	1.08E-03
	3	8.46E-01	2.25E-16	8.46E-01	1.14E-03	8.46E-01	2.25E-16	8.45E-01	4.07E-03
	4	8.69E-01	7.89E-16	8.69E-01	2.10E-03	8.69E-01	1.74E-05	8.63E-01	6.04E-03
	5	8.89E-01	4.70E-03	8.88E-01	9.30E-03	8.93E-01	5.35E-03	8.75E-01	9.86E-03
Bird	2	8.30E-01	2.25E-16	8.30E-01	1.97E-04	8.30E-01	2.25E-16	8.30E-01	6.80E-04
	3	8.53E-01	5.63E-16	8.53E-01	2.64E-04	8.53E-01	3.91E-05	8.52E-01	1.94E-03
	4	8.67E-01	5.63E-16	8.67E-01	1.17E-03	8.67E-01	1.62E-04	8.64E-01	3.16E-03
	5	8.78E-01	6.76E-16	8.79E-01	1.89E-03	8.78E-01	3.21E-04	8.69E-01	1.16E-02
Ocean	2	7.30E-01	5.63E-16	7.30E-01	4.54E-04	7.30E-01	5.63E-16	7.31E-01	3.72E-03
	3	8.14E-01	5.63E-16	8.13E-01	1.93E-03	8.14E-01	5.63E-16	8.06E-01	1.05E-02
	4	8.48E-01	4.51E-16	8.46E-01	3.33E-03	8.48E-01	1.11E-05	8.39E-01	9.25E-03
	5	8.76E-01	5.63E-16	8.72E-01	4.21E-03	8.75E-01	4.69E-04	8.55E-01	1.59E-02
Snow	2	7.02E-01	2.25E-16	7.02E-01	5.23E-04	7.02E-01	2.25E-16	7.01E-01	1.24E-03
	3	8.09E-01	0.00E+00	8.08E-01	2.09E-03	8.09E-01	0.00E+00	8.07E-01	9.92E-03
	4	8.18E-01	0.00E+00	8.13E-01	5.34E-03	8.18E-01	7.47E-05	8.10E-01	1.73E-02
	5	8.32E-01	4.57E-03	8.33E-01	6.89E-03	8.29E-01	5.40E-03	8.26E-01	1.68E-02
Kangaroo	2	6.75E-01	4.51E-16	6.75E-01	1.25E-03	6.75E-01	4.51E-16	6.73E-01	4.19E-03
	3	7.62E-01	5.63E-16	7.61E-01	3.59E-03	7.62E-01	5.63E-16	7.58E-01	1.26E-02
	4	8.26E-01	6.33E-05	8.26E-01	5.08E-03	8.26E-01	3.01E-05	8.07E-01	1.89E-02
	5	8.68E-01	8.00E-05	8.67E-01	5.21E-03	8.68E-01	8.93E-04	8.43E-01	1.68E-02

considered in this analysis, allowing each algorithm to be evaluated under different segmentation conditions. The algorithms arranged in the table were determined according to the fitness value obtained during the selection process for each threshold. According to this criterion, the following order was obtained: ESC, DCS, PLO, MGO, MSO, and NOA. From a visual and subjective perspective, no superior performance was observed among the algorithms. On the other hand, the algorithm that showed the best overall performance was ESC. In contrast, the algorithm that demonstrated the worst overall performance was NOA, resulting in the least favorable among the methods analyzed based on fitness.

Table 7 shows a comparison of the performance of several segmentation algorithms used on the same test image called "Airplane." Five thresholds were considered in this analysis, allowing each algorithm to be evaluated under different segmentation conditions. The algorithms organized in the table are determined by the fitness score obtained during the selection process for each threshold. According to this criterion, the following order was obtained: ESC, DCS, MGO, PLO, NOA, MSO.

From a visual and subjective perspective, the image that lost the most detail despite having the highest number of thresholds in these experiments was the

Table 8. Results obtained for SSIM using the BSDS300 dataset for MSO, NOA, and PLO algorithms.

Algorithm		MSO		NOA		PLO	
Image	Th	Mean	Std	Mean	Std	Mean	Std
Starfish	2	**1.04E+02**	1.04E+02	1.03E+02	1.33E+02	5.88E-01	5.11E-03
	3	1.11E+02	1.12E+02	**1.18E+02**	1.29E+02	6.80E-01	1.02E-02
	4	1.08E+02	1.08E+02	**1.19E+02**	1.25E+02	7.46E-01	1.99E-02
	5	**1.08E+02**	1.09E+02	1.00E+02	1.03E+02	7.81E-01	2.05E-02
Firefighters	2	8.80E+01	8.80E+01	**9.80E+01**	8.95E+01	7.08E-01	4.12E-03
	3	**8.93E+01**	8.93E+01	8.87E+01	1.20E+02	7.84E-01	1.05E-02
	4	8.30E+01	8.28E+01	**9.10E+01**	1.05E+02	8.30E-01	1.29E-02
	5	9.26E+01	9.60E+01	**1.01E+02**	9.78E+01	8.50E-01	1.10E-02
Pyramid	2	**1.37E+02**	1.37E+02	1.19E+02	1.22E+02	5.12E-01	1.71E-02
	3	**1.29E+02**	1.29E+02	1.28E+02	1.02E+02	7.25E-01	3.22E-02
	4	1.30E+02	1.29E+02	**1.40E+02**	1.50E+02	7.59E-01	4.04E-02
	5	**1.31E+02**	1.30E+02	1.08E+02	1.10E+02	7.92E-01	3.35E-02
Airplane	2	**8.55E+01**	8.55E+01	7.85E+01	7.80E+01	8.74E-01	1.58E-02
	3	**9.47E+01**	9.47E+01	7.60E+01	1.04E+02	9.02E-01	1.60E-02
	4	9.45E+01	9.68E+01	**1.30E+02**	6.53E+01	9.22E-01	1.16E-02
	5	8.30E+01	8.24E+01	**1.16E+02**	9.02E+01	9.32E-01	1.08E-02
Owls	2	7.25E+01	7.25E+01	**7.35E+01**	6.30E+01	5.26E-01	1.92E-03
	3	**7.77E+01**	7.77E+01	6.07E+01	9.23E+01	5.70E-01	4.86E-03
	4	8.13E+01	8.25E+01	**8.58E+01**	8.48E+01	5.92E-01	1.01E-02
	5	8.14E+01	8.14E+01	**1.00E+02**	1.05E+02	6.07E-01	1.61E-02
Bird	2	**1.13E+02**	1.13E+02	9.90E+01	1.26E+02	8.58E-01	2.24E-03
	3	**1.12E+02**	1.13E+02	1.03E+02	1.21E+02	8.91E-01	5.63E-03
	4	1.15E+02	1.15E+02	**1.20E+02**	1.06E+02	9.08E-01	1.52E-02
	5	1.06E+02	1.07E+02	**1.07E+02**	8.36E+01	9.24E-01	1.46E-02
Ocean	2	9.40E+01	9.40E+01	**1.01E+02**	9.85E+01	6.55E-01	1.75E-02
	3	9.10E+01	9.10E+01	**1.08E+02**	9.13E+01	7.82E-01	2.22E-02
	4	9.15E+01	8.93E+01	**1.04E+02**	9.93E+01	8.17E-01	2.52E-02
	5	9.48E+01	9.48E+01	**9.94E+01**	7.76E+01	8.46E-01	2.48E-02
Snow	2	**1.26E+02**	1.26E+02	1.19E+02	1.26E+02	8.00E-01	5.80E-03
	3	**1.52E+02**	1.52E+02	1.46E+02	1.55E+02	8.62E-01	1.54E-02
	4	1.55E+02	1.56E+02	**1.61E+02**	1.19E+02	8.73E-01	1.62E-02
	5	1.38E+02	1.38E+02	**1.69E+02**	1.57E+02	8.82E-01	1.85E-02
Kangaroo	2	**9.10E+01**	9.10E+01	7.95E+01	7.25E+01	6.85E-01	7.02E-03
	3	8.53E+01	8.53E+01	**8.90E+01**	1.02E+02	7.71E-01	2.50E-02
	4	8.53E+01	8.58E+01	**9.45E+01**	9.00E+01	8.25E-01	2.51E-02
	5	8.48E+01	8.52E+01	**9.10E+01**	7.18E+01	8.56E-01	2.44E-02

NOA, showing an image with an extremely black background, losing a large amount of visual detail that is perceived in the other images with the same threshold level. Despite this, the NOA was not the worst performing algorithm; the algorithm with the worst overall performance was MSO, which resulted in the least favorable among the analyzed methods in terms of fitness. And the one that performed the best in terms of fitness was ESC.

4.4 Statistical analysis

A recent trend in the field involves the application of statistical analysis as a comparative method for MAs performance. The present study employs the nonparametric statistical analysis of the Friedman test. The purpose of this evaluation is to determine the

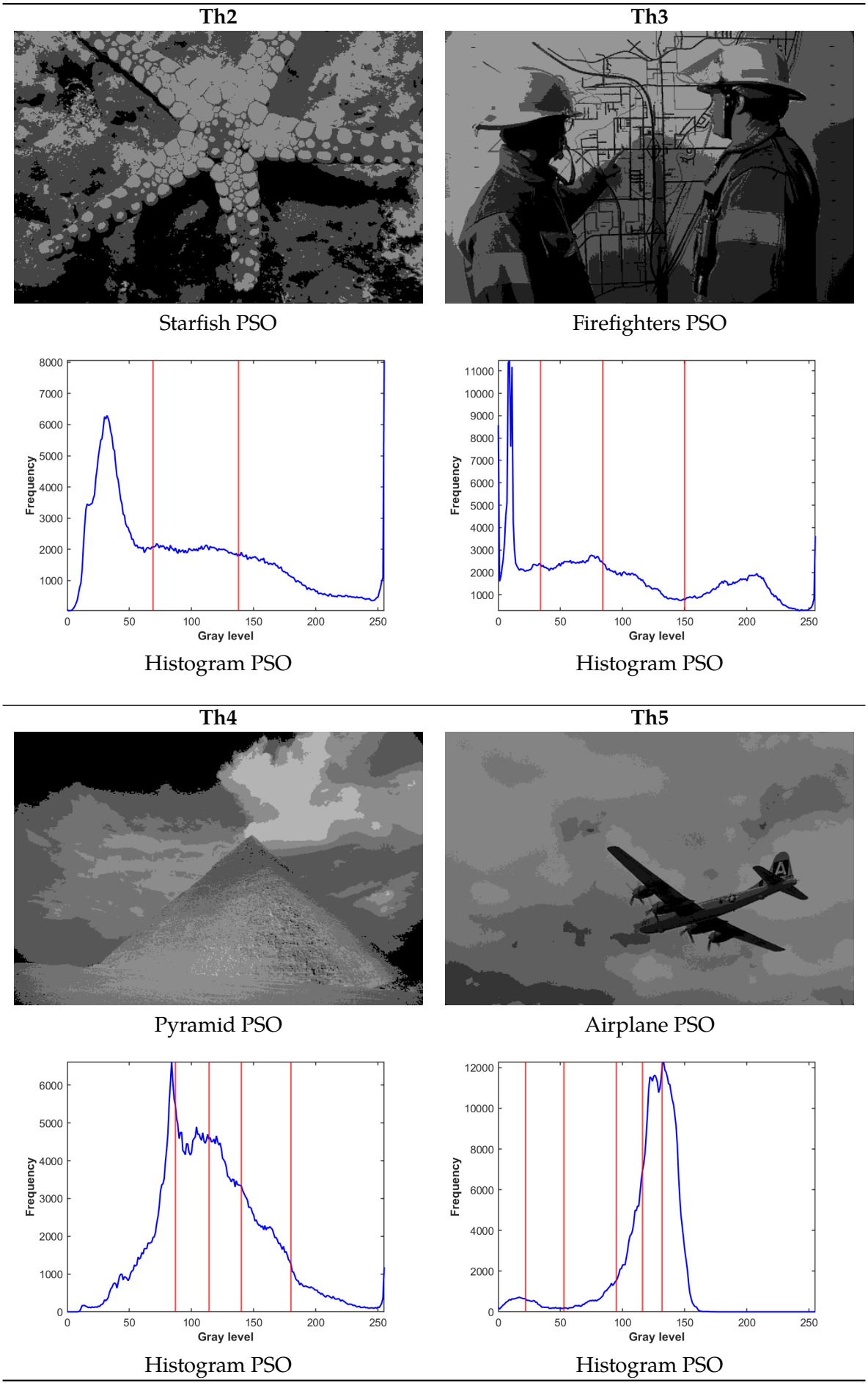

Figure 3. Applying PSO to segmentation as a baseline for comparison.

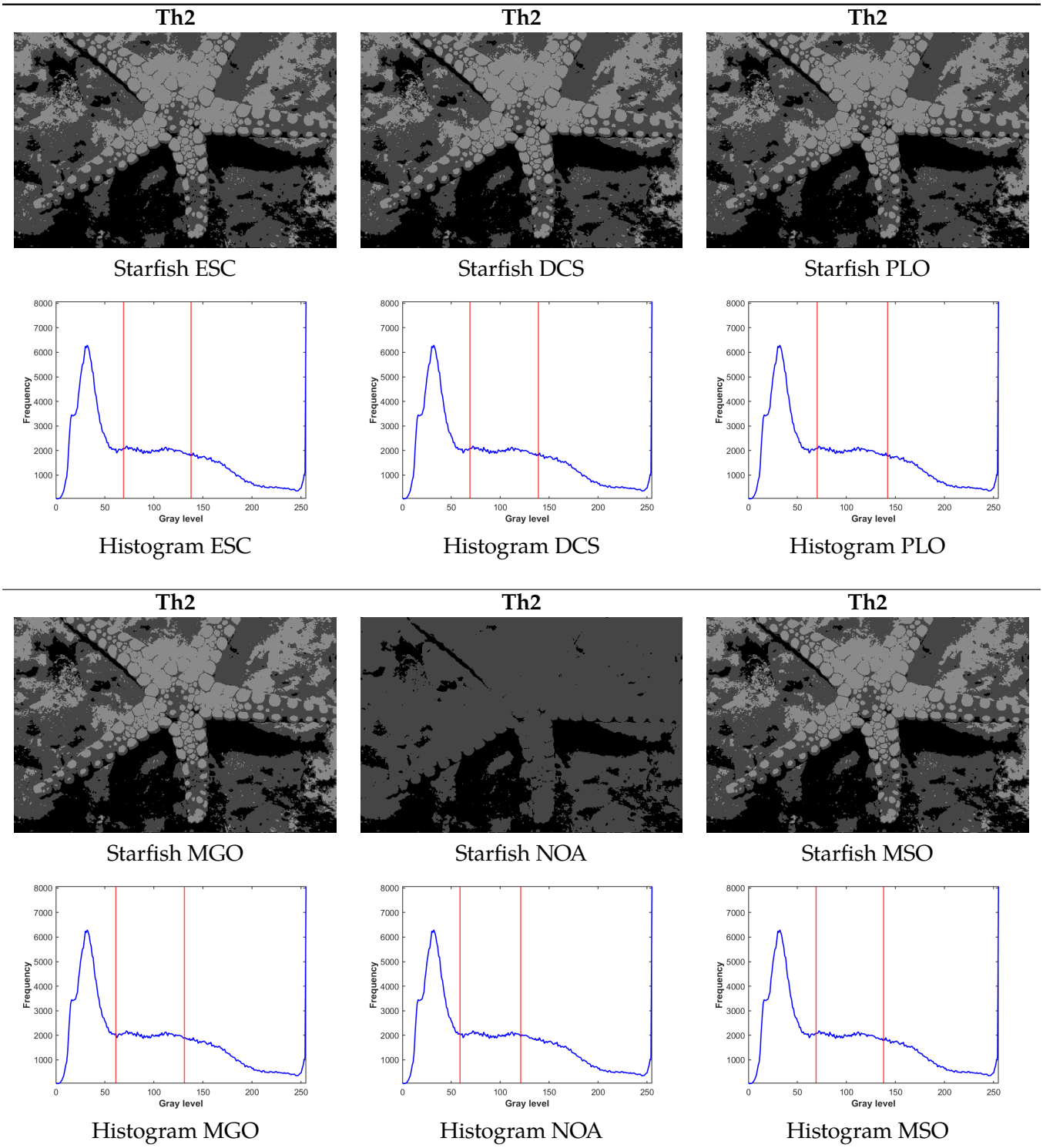

Figure 4. A comparison of the segmentation performed with the best thresholds obtained by each algorithm for Th=2.

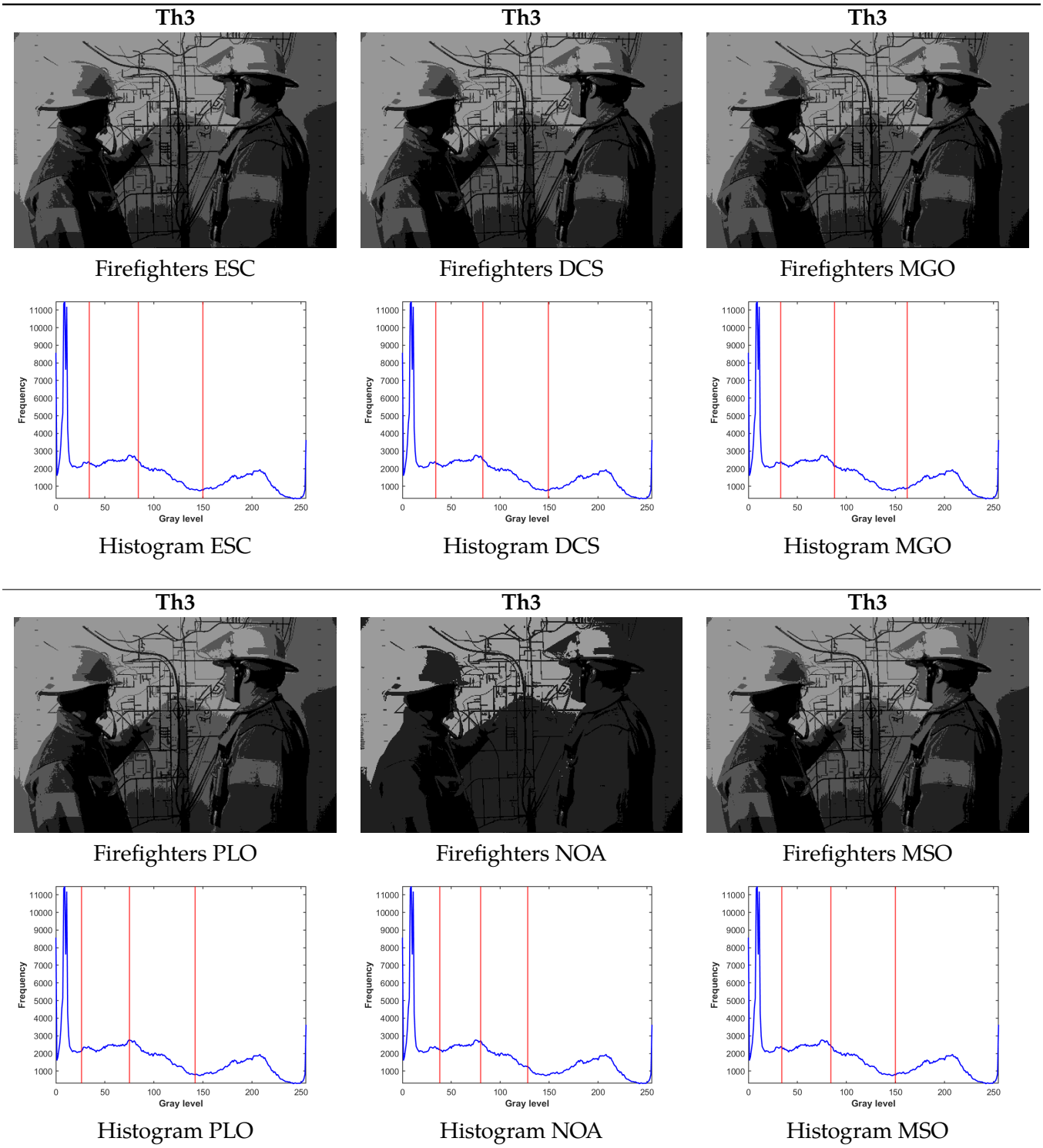

Figure 5. A comparison of the segmentation performed with the best thresholds obtained by each algorithm for Th=3

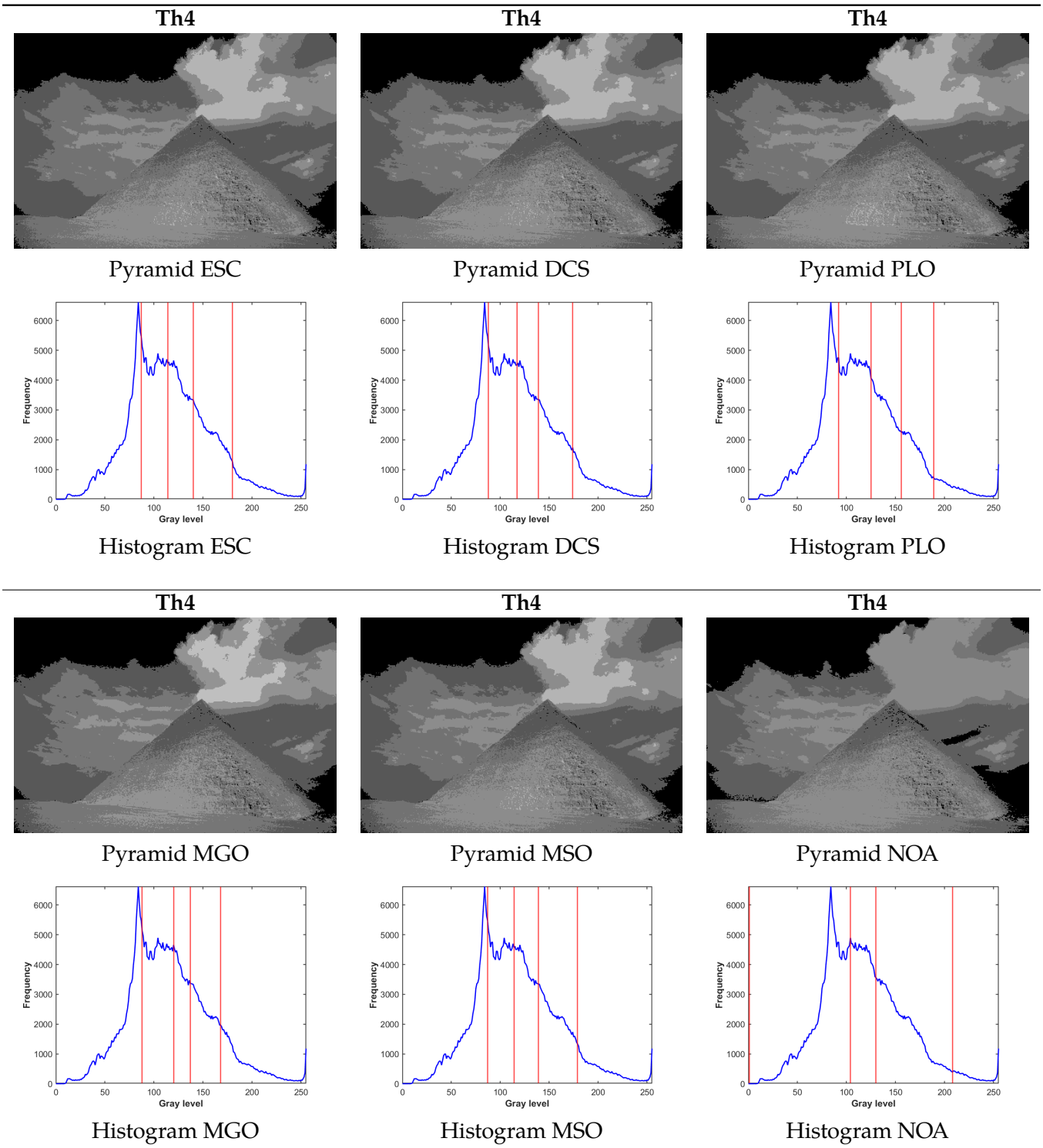

Figure 6. A comparison of the segmentation performed with the best thresholds obtained by each algorithm for Th=4.

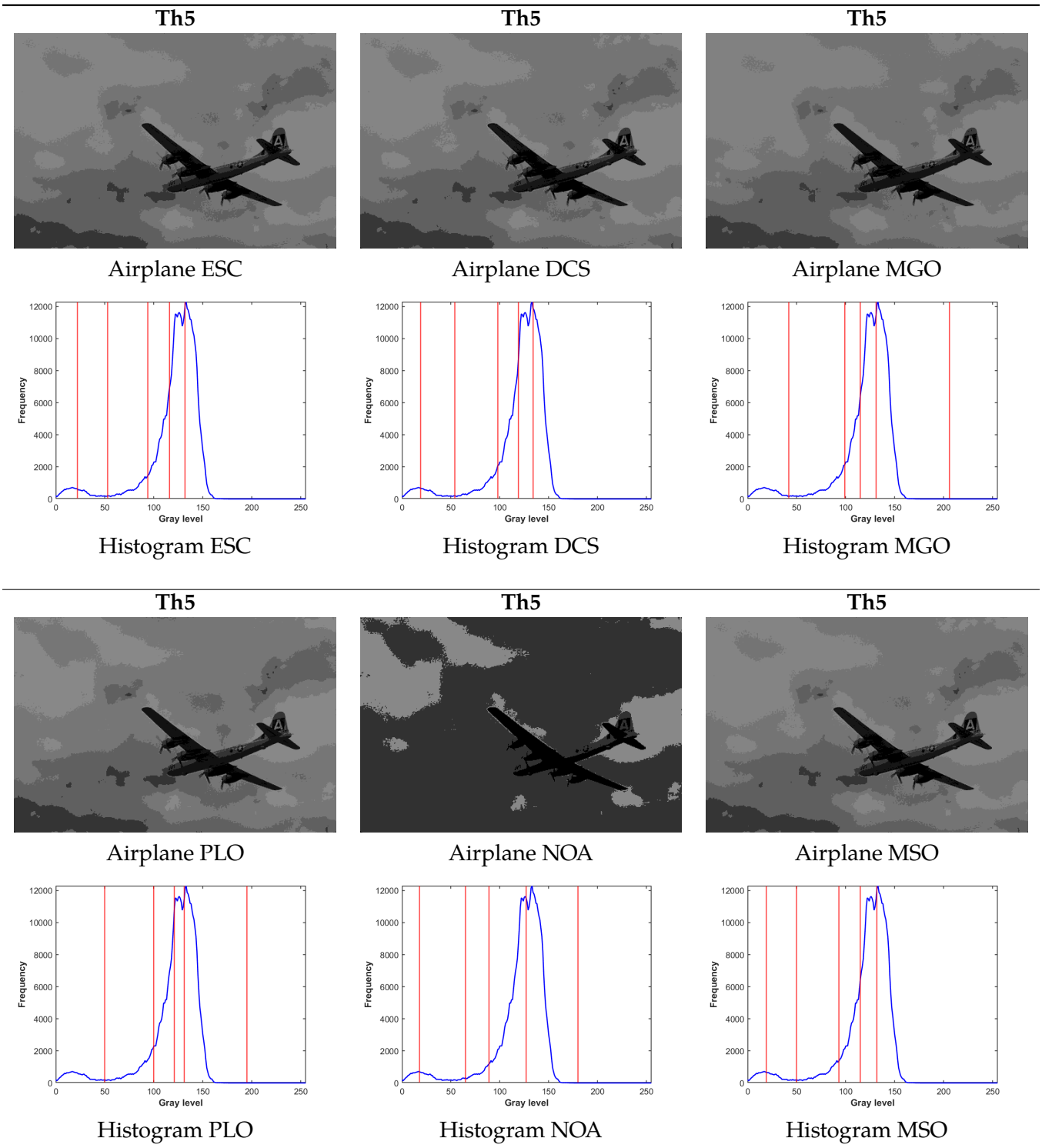

Figure 7. A comparison of the segmentation performed with the best thresholds obtained by each algorithm for Th=5.

Table 9. Results obtained for FSIM using the BSDS300 dataset for PSO, DCS, ESC, and MGO algorithms.

Algorithm		PSO		DCS		ESC		MGO	
Image	Th	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Starfish	2	6.28E-01	4.51E-16	6.28E-01	4.57E-04	6.28E-01	4.51E-16	6.27E-01	1.66E-03
	3	7.12E-01	2.25E-16	7.11E-01	1.01E-03	7.12E-01	2.25E-16	7.07E-01	5.52E-03
	4	7.69E-01	7.92E-05	7.67E-01	3.12E-03	7.69E-01	1.61E-04	7.57E-01	7.88E-03
	5	8.12E-01	7.89E-16	8.10E-01	3.47E-03	8.13E-01	2.48E-04	7.94E-01	1.03E-02
Firefighters	2	7.64E-01	3.38E-16	7.64E-01	7.59E-04	7.64E-01	3.38E-16	7.64E-01	1.45E-03
	3	8.14E-01	1.13E-16	8.14E-01	9.39E-04	8.14E-01	1.13E-16	8.10E-01	4.91E-03
	4	8.48E-01	3.38E-16	8.45E-01	1.92E-03	8.48E-01	3.00E-05	8.35E-01	7.13E-03
	5	8.60E-01	3.87E-05	8.58E-01	3.33E-03	8.60E-01	4.99E-04	8.52E-01	7.67E-03
Pyramid	2	7.13E-01	1.13E-16	7.13E-01	1.28E-04	7.13E-01	1.13E-16	7.12E-01	2.32E-03
	3	7.48E-01	4.51E-16	7.49E-01	2.54E-03	7.48E-01	4.51E-16	7.44E-01	7.99E-03
	4	7.75E-01	2.25E-16	7.75E-01	2.59E-03	7.75E-01	1.44E-04	7.68E-01	8.56E-03
	5	7.95E-01	3.42E-03	7.94E-01	2.72E-03	7.96E-01	3.19E-04	7.85E-01	8.55E-03
Airplane	2	8.43E-01	5.63E-16	8.45E-01	1.77E-03	8.43E-01	5.63E-16	8.45E-01	3.03E-03
	3	8.46E-01	5.94E-04	8.45E-01	2.23E-03	8.46E-01	4.93E-04	8.55E-01	1.01E-02
	4	8.58E-01	2.55E-03	8.60E-01	3.24E-03	8.59E-01	4.51E-16	8.57E-01	8.55E-03
	5	8.60E-01	2.36E-03	8.62E-01	4.29E-03	8.59E-01	2.16E-04	8.61E-01	7.75E-03
Owls	2	8.05E-01	0.00E+00	8.05E-01	1.36E-04	8.05E-01	0.00E+00	8.04E-01	1.08E-03
	3	8.46E-01	2.25E-16	8.46E-01	1.14E-03	8.46E-01	2.25E-16	8.45E-01	4.07E-03
	4	8.69E-01	7.89E-16	8.69E-01	2.10E-03	8.69E-01	1.74E-05	8.63E-01	6.04E-03
	5	8.89E-01	4.70E-03	8.88E-01	9.30E-03	8.93E-01	5.35E-03	8.75E-01	9.86E-03
Bird	2	8.30E-01	2.25E-16	8.30E-01	1.97E-04	8.30E-01	2.25E-16	8.30E-01	6.80E-04
	3	8.53E-01	5.63E-16	8.53E-01	2.64E-04	8.53E-01	3.91E-05	8.52E-01	1.94E-03
	4	8.67E-01	5.63E-16	8.67E-01	1.17E-03	8.67E-01	1.62E-04	8.64E-01	3.16E-03
	5	8.78E-01	6.76E-16	8.79E-01	1.89E-03	8.78E-01	3.21E-04	8.69E-01	1.16E-02
Ocean	2	7.30E-01	5.63E-16	7.30E-01	4.54E-04	7.30E-01	5.63E-16	7.31E-01	3.72E-03
	3	8.14E-01	5.63E-16	8.13E-01	1.93E-03	8.14E-01	5.63E-16	8.06E-01	1.05E-02
	4	8.48E-01	4.51E-16	8.46E-01	3.33E-03	8.48E-01	1.11E-05	8.39E-01	9.25E-03
	5	8.76E-01	5.63E-16	8.72E-01	4.21E-03	8.75E-01	4.69E-04	8.55E-01	1.59E-02
Snow	2	7.02E-01	2.25E-16	7.02E-01	5.23E-04	7.02E-01	2.25E-16	7.01E-01	1.24E-03
	3	8.09E-01	0.00E+00	8.08E-01	2.09E-03	8.09E-01	0.00E+00	8.07E-01	9.92E-03
	4	8.18E-01	0.00E+00	8.13E-01	5.34E-03	8.18E-01	7.47E-05	8.10E-01	1.73E-02
	5	8.32E-01	4.57E-03	8.33E-01	6.89E-03	8.29E-01	5.40E-03	8.26E-01	1.68E-02
Kangaroo	2	6.75E-01	4.51E-16	6.75E-01	1.25E-03	6.75E-01	4.51E-16	6.73E-01	4.19E-03
	3	7.62E-01	5.63E-16	7.61E-01	3.59E-03	7.62E-01	5.63E-16	7.58E-01	1.26E-02
	4	8.26E-01	6.33E-05	8.26E-01	5.08E-03	8.26E-01	3.01E-05	8.07E-01	1.89E-02
	5	8.68E-01	8.00E-05	8.67E-01	5.21E-03	8.68E-01	8.93E-04	8.43E-01	1.68E-02

existence of a substantial discrepancy between the algorithms in question.

Table 11 shows the results of the Friedman test carried out on the fitness of the PSO, DCS, ESC, MGO, MSO, NOA, and PLO algorithms, considering thresholds from 2 to 5. The values presented correspond to the average ranges obtained and the final position assigned to each algorithm based on its relative performance. In all cases, the $p - value$ obtained is less than 0.05, confirming the existence of statistically significant differences between the evaluated algorithms.

In the four scenarios examined, PSO consistently achieves the lowest average ranges, with values of (1.50, 1.44, 1.28, and 1.22), respectively, for thresholds ranging from 2 to 5. The ESC also performs well, ranking second for thresholds of 3 to 5 with values close to those of PSO, reflecting its competitiveness. DCS maintains a stable third-place position across all thresholds, establishing itself as an intermediate algorithm in terms of performance.

On the other hand, MGO and PLO exhibit similar behavior, ranking in the middle (fourth and fifth place, respectively), with values ranging from 4.2 to 4.7, indicating less efficient performance compared to the leading algorithms. Finally, MSO and NOA consistently rank last, with ranges above 6.0, indicating

Table 10. Results obtained for FSIM using the BSDS300 dataset for MSO, NOA and PLO algorithms.

Algorithm		MSO		NOA		PLO	
Image	Th	Mean	Std	Mean	Std	Mean	Std
Starfish	2	1.04E+02	1.04E+02	**1.05E+02**	1.02E+02	6.28E-01	1.23E-03
	3	1.11E+02	1.11E+02	**1.36E+02**	1.24E+02	7.07E-01	6.15E-03
	4	1.06E+02	1.08E+02	**1.14E+02**	1.10E+02	7.61E-01	8.76E-03
	5	**1.07E+02**	1.07E+02	1.05E+02	1.25E+02	7.94E-01	1.01E-02
Firefighters	2	8.80E+01	8.80E+01	**9.25E+01**	8.80E+01	7.63E-01	2.13E-03
	3	**8.93E+01**	8.90E+01	7.13E+01	1.15E+02	8.09E-01	5.58E-03
	4	8.35E+01	8.30E+01	**8.95E+01**	7.83E+01	8.37E-01	7.13E-03
	5	9.14E+01	9.44E+01	**9.24E+01**	9.00E+01	8.51E-01	7.90E-03
Pyramid	2	1.37E+02	1.37E+02	**1.46E+02**	1.20E+02	7.13E-01	1.31E-03
	3	**1.29E+02**	1.29E+02	1.27E+02	1.46E+02	7.46E-01	6.65E-03
	4	**1.30E+02**	1.30E+02	1.16E+02	1.46E+02	7.66E-01	7.51E-03
	5	1.29E+02	1.33E+02	**1.41E+02**	1.01E+02	7.85E-01	9.88E-03
Airplane	2	**8.55E+01**	8.55E+01	7.55E+01	8.55E+01	8.45E-01	2.92E-03
	3	**9.33E+01**	9.33E+01	5.83E+01	6.07E+01	8.52E-01	1.03E-02
	4	**9.65E+01**	9.58E+01	9.03E+01	8.58E+01	8.57E-01	1.10E-02
	5	8.18E+01	8.20E+01	**1.02E+02**	1.10E+02	8.60E-01	9.09E-03
Owls	2	7.25E+01	7.25E+01	**7.55E+01**	8.70E+01	8.04E-01	1.54E-03
	3	7.77E+01	7.77E+01	**8.00E+01**	7.60E+01	8.43E-01	5.42E-03
	4	**8.23E+01**	8.15E+01	7.88E+01	7.60E+01	8.64E-01	7.03E-03
	5	8.14E+01	8.14E+01	**1.30E+02**	1.32E+02	8.76E-01	1.22E-02
Bird	2	**1.13E+02**	1.13E+02	1.11E+02	1.10E+02	8.30E-01	8.24E-04
	3	1.13E+02	1.12E+02	**1.36E+02**	1.16E+02	8.52E-01	2.93E-03
	4	**1.15E+02**	1.16E+02	1.04E+02	8.85E+01	8.61E-01	1.05E-02
	5	**1.05E+02**	1.07E+02	9.66E+01	1.23E+02	8.72E-01	9.28E-03
Ocean	2	**9.40E+01**	9.40E+01	9.20E+01	8.85E+01	7.30E-01	4.78E-03
	3	**9.10E+01**	9.17E+01	8.17E+01	9.60E+01	8.05E-01	1.07E-02
	4	9.03E+01	9.15E+01	**9.15E+01**	1.10E+02	8.34E-01	1.50E-02
	5	9.62E+01	8.86E+01	**1.13E+02**	8.10E+01	8.53E-01	1.31E-02
Snow	2	1.26E+02	1.26E+02	**1.36E+02**	1.38E+02	7.01E-01	2.37E-03
	3	**1.52E+02**	1.52E+02	1.46E+02	1.54E+02	8.02E-01	1.11E-02
	4	**1.56E+02**	1.56E+02	1.23E+02	1.14E+02	8.16E-01	1.46E-02
	5	1.38E+02	1.54E+02	**1.73E+02**	1.45E+02	8.23E-01	1.66E-02
Kangaroo	2	**9.10E+01**	9.10E+01	8.05E+01	8.70E+01	6.72E-01	5.21E-03
	3	**8.53E+01**	8.53E+01	7.53E+01	6.90E+01	7.57E-01	1.22E-02
	4	**8.58E+01**	8.58E+01	6.58E+01	9.63E+01	8.08E-01	1.57E-02
	5	8.68E+01	8.70E+01	**8.84E+01**	1.13E+02	8.40E-01	1.85E-02

significantly lower performance compared to the other methods.

5 Conclusion

This study introduces six novel evolutionary strategies employing multilevel thresholding with minimum cross-entropy as objective function. These strategies were developed to identify optimal thresholds by minimizing the minimum cross-entropy. Experiments were conducted using the BSDS300 dataset and the nine most complicated images. Image segmentation was performed at five threshold levels (e.g., 2–5). Three different indicators were considered in image processing to evaluate important characteristics of the segmented images. These indicators are the PSNR, SSIM, and FSIM.

Table 11. Friedman Rank Test with 2–5 Th.

	2 Th		3 Th		4 Th		5 Th	
Algorithm	**Rank**	**Final**	**Rank**	**Final**	**Rank**	**Final**	**Rank**	**Final**
PSO	1.50	1	1.44	1	1.28	1	1.22	1
DCS	3.00	2	3.00	3	3.00	3	3.00	3
ESC	1.50	1	1.56	2	1.72	2	1.78	2
MGO	4.22	3	4.22	4	4.44	4	4.33	4
MSO	6.44	5	6.22	6	6.22	6	6.33	6
NOA	6.56	6	6.78	7	6.78	7	6.67	7
PLO	4.78	4	4.78	5	4.56	5	4.67	5
p-value	1.58E-09		1.45E-09		2.13E-09		2.22E-09	

The significance of this study lies in the implementation of evolutionary strategies applied to the problem of MTH using MCE as the objective function. A comprehensive evaluation of the results obtained led to the conclusion that the optimal algorithm for this implementation is the ESC algorithm, considering the objective function values.

Conversely, it is acknowledged that this study allows extensive scope for further implementation. In subsequent studies, it is planned to implement a variety of multilevel thresholding techniques to validate the efficacy of the algorithms utilized in this comparison. Furthermore, it is intended to conduct experiments with other test sets, not only in image segmentation but also in single-objective optimization or even multi-objective optimization.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Oliva, D., Hinojosa, S., Osuna-Enciso, V., Cuevas, E., Pérez-Cisneros, M., & Sanchez-Ante, G. (2019). Image segmentation by minimum cross entropy using evolutionary methods. *Soft Computing, 23*(2), 431–450. [CrossRef]

[2] Wang, Z., Wang, E., & Zhu, Y. (2020). Image segmentation evaluation: a survey of methods. *Artificial Intelligence Review, 53*(8), 5637–5674. [CrossRef]

[3] Kumar, M. J., Kumar, D. G. R., & Reddy, R. V. K. (2014). Review on image segmentation techniques. *International Journal of Scientific Research Engineering & Technology, 3*(6), 993–997.

[4] Kaur, D., & Kaur, Y. (2014). Various image segmentation techniques: a review. *International journal of computer science and Mobile Computing, 3*(5), 809–814.

[5] Sahoo, P. K., Soltani, S. A. K. C., & Wong, A. K. C. (1988). A survey of thresholding techniques. *Computer vision, graphics, and image processing, 41*(2), 233–260. [CrossRef]

[6] Bhargavi, K., & Jyothi, S. (2014). A survey on threshold based segmentation technique in image processing. *International Journal of Innovative Research and Development, 3*(12), 234–239.

[7] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics, 9*(1), 62–66. [CrossRef]

[8] Jena, B., Naik, M. K., Panda, R., & Abraham, A. (2021). Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta Ray foraging optimization. *Engineering Applications of Artificial Intelligence, 103*, 104293. [CrossRef]

[9] Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing, 29*(3), 273–285. [CrossRef]

[10] Agrawal, S., Panda, R., Bhuyan, S., & Panigrahi, B. K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm and Evolutionary Computation, 11*, 16–30. [CrossRef]

[11] Sahoo, P., Wilkins, C., & Yeager, J. (1997). Threshold selection using Renyi's entropy. *Pattern recognition, 30*(1), 71–84. [CrossRef]

[12] Chao, Y., Dai, M., Chen, K., Chen, P., & Zhang, Z. (2016). Fuzzy entropy based multilevel image thresholding using modified gravitational search

algorithm. In *2016 IEEE international conference on industrial technology (ICIT)* (pp. 752–757). [CrossRef]

[13] Fang, S. C., Peterson, E. L., & Rajasekera, J. R. (1992). Minimum cross-entropy analysis with entropy-type constraints. *Journal of computational and applied mathematics, 39*(2), 165–178. [CrossRef]

[14] Horng, M. H., & Liou, R. J. (2011). Multilevel minimum cross entropy threshold selection based on the firefly algorithm. *Expert Systems with Applications, 38*(12), 14805–14811. [CrossRef]

[15] Horng, M. H. (2010). Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. *Expert Systems with Applications, 37*(6), 4580–4592. [CrossRef]

[16] Pare, S., Kumar, A., Singh, G. K., & Bajaj, V. (2020). Image segmentation using multilevel thresholding: a research review. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 44*(1), 1–29. [CrossRef]

[17] Karakoyun, M., Baykan, N. A., & Hacibeyoglu, M. (2017). Multi-level thresholding for image segmentation with swarm optimization algorithms. *International Research Journal of Electronics & Computer Engineering, 3*(3), 1.

[18] Merzban, M. H., & Elbayoumi, M. (2019). Efficient solution of Otsu multilevel image thresholding: A comparative study. *Expert Systems with Applications, 116*, 299–309. [CrossRef]

[19] Li, C. H., & Lee, C. K. (1993). Minimum cross entropy thresholding. *Pattern recognition, 26*(4), 617–625. [CrossRef]

[20] Abdel-Basset, M., Mohamed, R., & Abouhawwash, M. (2022). A new fusion of whale optimizer algorithm with Kapur's entropy for multi-threshold image segmentation: Analysis and validations. *Artificial intelligence review, 55*(8), 6389–6459. [CrossRef]

[21] Ma, P., & Geng, Y. (2024). An improved whale optimization algorithm based on multi-populations and multi-strategies for multilevel threshold image segmentation. In *2024 4th International Conference on Neural Networks, Information and Communication Engineering (NNICE)* (pp. 826–831). [CrossRef]

[22] Shi, J., Chen, Y., Cai, Z., Heidari, A. A., Chen, H., & He, Q. (2024). Multi-threshold image segmentation using a boosted whale optimization: case study of breast invasive ductal carcinomas. *Cluster Computing, 27*(10), 14891–14949. [CrossRef]

[23] Rahkar Farshi, T., & K. Ardabili, A. (2021). A hybrid firefly and particle swarm optimization algorithm applied to multilevel image thresholding. *Multimedia Systems, 27*(1), 125–142. [CrossRef]

[24] Song, S., Jia, H., & Ma, J. (2019). A chaotic electromagnetic field optimization algorithm based on fuzzy entropy for multilevel thresholding color image segmentation. *Entropy, 21*(4), 398. [CrossRef]

[25] Thapliyal, S., & Kumar, N. (2024). ASCAEO: accelerated sine cosine algorithm hybridized with equilibrium optimizer with application in image segmentation using multilevel thresholding. *Evolving Systems, 15*(4), 1297–1358. [CrossRef]

[26] Hammouche, K., Diaf, M., & Siarry, P. (2008). A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision and Image Understanding, 109*(2), 163–175. [CrossRef]

[27] Ma, M., & Zhu, Q. (2017). Multilevel thresholding image segmentation based on shuffled frog leaping algorithm. *Journal of Computational and Theoretical Nanoscience, 14*(8), 3794–3801. [CrossRef]

[28] Ouyang, K., Fu, S., Chen, Y., Cai, Q., Heidari, A. A., & Chen, H. (2024). Escape: an optimization method based on crowd evacuation behaviors. *Artificial Intelligence Review, 58*(1), 19. [CrossRef]

[29] Zheng, B., Chen, Y., Wang, C., Heidari, A. A., Liu, L., & Chen, H. (2024). The moss growth optimization (MGO): concepts and performance. *Journal of Computational Design and Engineering, 11*(5), 184–221. [CrossRef]

[30] He, J., Zhao, S., Ding, J., & Wang, Y. (2025). Mirage search optimization: Application to path planning and engineering design problems. *Advances in Engineering Software, 203*, 103883. [CrossRef]

[31] Abdel-Basset, M., Mohamed, R., Hezam, I. M., Sallam, K. M., & Hameed, I. A. (2024). An improved nutcracker optimization algorithm for discrete and continuous optimization problems: Design, comprehensive analysis, and engineering applications. *Heliyon, 10*(17). [CrossRef]

[32] Yuan, C., Zhao, D., Heidari, A. A., Liu, L., Chen, Y., & Chen, H. (2024). Polar Lights Optimizer: Algorithm and Applications in Image Segmentation and Feature Selection. *Neurocomputing, 607*, 128427. [CrossRef]

[33] Heidari, A. (2024). Polar lights optimizer: Algorithm and applications in image segmentation and feature selection [Computer software]. *GitHub*. https://github.com/aliasgharheidaricom/Polar-Lights-Optimizer-Algorithm-and-Applications-in-Image-Segmentation-and-Feature-Selection

[34] Brink, A. D., & Pendock, N. E. (1996). Minimum cross-entropy threshold selection. *Pattern Recognition, 29*(1), 179-188. [CrossRef]

[35] Kullback, S. (1997). *Information theory and statistics*. Courier Corporation.

[36] Horé, A., & Ziou, D. (2010). Image Quality Metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition* (pp. 2366-2369). [CrossRef]

[37] Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing, 20*(8), 2378-2386. [CrossRef]

[38] Martin, D. R., Fowlkes, C. C., Tal, D., & Malik, J. (2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proc. 8th Int'l Conf. Computer Vision* (Vol. 2, pp. 416–423). [CrossRef]

[39] Duankhan, P., Sunat, K., Chiewchanwattana, S., & Nasa-ngium, P. (2024). The Differentiated Creative Search (DCS): Leveraging differentiated knowledge-acquisition and creative realism to address complex optimization problems. *Expert Systems with Applications, 252*, 123734. [CrossRef]

[40] Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics letters, 44*(13), 800–801. [CrossRef]

[41] Avcibas, I., Sankur, B., & Sayood, K. (2002). Statistical evaluation of image quality measures. *Journal of Electronic imaging, 11*(2), 206–223. [CrossRef]

[42] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing, 13*(4), 600–612. [CrossRef]

[43] Sara, U., Akter, M., & Uddin, M. S. (2019). Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study. *Journal of Computer and Communications, 7*(3), 8–18. [CrossRef]

[44] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association, 32*(200), 675–701. [CrossRef]

[45] Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92. [CrossRef]

[46] Scheff, S. W. (2016). Chapter 8 - Nonparametric Statistics. In S. W. Scheff (Ed.), *Fundamental Statistical Principles for the Neurobiologist* (pp. 157-182). Academic Press. [CrossRef]

[47] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942-1948 vol.4). [CrossRef]

[48] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 69-73). [CrossRef]

[49] Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* (Vol. 3, pp. 1945-1950 Vol. 3). [CrossRef]

Omar Alvarez was born in Guadalajara, Jalisco, Mexico, in 1998. He received a B.S. degree in Communications and Electronics Engineering from the University of Guadalajara, Mexico, in 2023. He is currently pursuing an M.Sc. degree in Electronic Engineering and Computer Sciences at the University of Guadalajara. His research interests include machine learning and image processing.

Luis A. Beltran received his B.S. degree in Mechatronic Engineering from the Universidad Politécnica de Sinaloa in 2020. In 2024, he obtained his M.Sc. degree in Electronics and Computer Engineering from the Universidad de Guadalajara, at the Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI), where he is currently pursuing a Ph.D. in Computational Intelligence. Throughout his postgraduate studies, he has focused on Evolutionary Computation and metaheuristic algorithms. His research interests lie in the intersection of Artificial Intelligence and optimization techniques, particularly in the development and application of metaheuristics and hyper-heuristics. He is an active member of the Automatic Control and Intelligent Systems Research Group, where he explores intelligent optimization strategies for complex systems.

Angel Casas-Ordaz received a B.S. degree in Electronic Engineering from the Instituto Tecnológico de Ciudad Juárez in 2018, and later the M.Sc. degree in Electronics and Computer Engineering from the Universidad de Guadalajara at the Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) in 2022; He is currently a Ph.D. student in Electronics and Computer Science at the same campus. During his postgraduate research, he specialized in the field of Evolutionary Computation and Image Segmentation. His current research interests include Evolutionary Computation and Artificial Intelligence, specializing in Metaheuristic Algorithms in the Automatic Control and Intelligent Systems Research Group.

Jorge Ramos-Frutos received a B.S. degree in industrial engineering from the Instituto Tecnológico de Jiquilpan, Michoacán, Mexico, in 2018 and an M.Sc. degree in Science and Technology from the Centro Innovación Aplicada en Tecnologías Competitivas (CIATEC), León, Guanajuato, Mexico, in 2021. He is studying for his doctorate at CIATEC under the supervision of Ph.D. Israel Miguel and Ph. D. Diego Oliva. His research interests are related to digital image processing, evolutionary computation, and machine learning.

Mario A. Navarro-Velázquez In 2019, he completed a Master of Science in Electronic Engineering and Computer Science, focusing his research on the design of metaheuristic algorithms and applications in image segmentation. In 2023, he obtained his PhD in Electronic and Computer Engineering at the Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) in Guadalajara, Mexico, concentrating on the coevolution of metaheuristic strategies to solve various optimization problems. His research interests include artificial intelligence, specifically the design and hybridization of evolutionary algorithms, the development of operators and hyperheuristics to solve high-dimensional problems, and the integration of evolutionary algorithms and machine learning.

Oscar Ramos-Soto received the B.S. degree in Communications and Electronics Engineering from the National Polytechnic Institute, Mexico, in 2019, and the M.Sc. degree in Electronic Engineering and Computer Sciences from the University of Guadalajara, Mexico, in 2021. Currently, he is pursuing a Ph.D. degree in Electronics and Computer Science at the University of Guadalajara, Mexico. His research interests primarily involve areas such as computer vision, digital image processing, machine learning, deep learning, biomedical engineering, and vision science.

Prof. Diego Oliva received a B.S. degree in Electronics and Computer Engineering from the Industrial Technical Education Center (CETI) of Guadalajara, Mexico, in 2007 and an M.Sc. degree in Electronic Engineering and Computer Sciences from the University of Guadalajara, Mexico, in 2010. He obtained a Ph. D. in Informatics in 2015 from the Universidad Complutense de Madrid. Currently, he is an Associate Professor at the University of Guadalajara in Mexico. He is a member of the Mexican National Research System (SNII), a Senior member of the IEEE, and a member of the Mexican Academy of Computer Sciences (AMEXCOMP). His research interests include evolutionary and swarm algorithms, hybridization of evolutionary and swarm algorithms, and computational intelligence.