Research Reports on Computer Science
https://ojs.wiserpub.com/index.php/RRCS

UNIVERSAL WISER PUBLISHER

Article

A Novel Optimization Algorithm for Otsu's Entropy-Based Multi-Level Thresholding for Image Segmentation

Karri Chiranjeevi[1*], M.S.R. Naidu[2], G.S.S.S.S.V. Krishna Mohan[2], Vuppula Manohar[3], Santosh Kumar Gottapu[4], Anil Kumar Indugupalli[1]

[1] Department of Electronics and Communication Engineering, A.U. College of Engineering, Andhra University, Visakhaptnam, India
[2] Department of Electronics and Communication Engineering, Aditya Institute of Technology and Management, Tekkali, India
[3] Department of Electronics and Communication Engineering, Vaagdevi Engineering College, Warangal, India
[4] Department of Civil, GVP College of Engineering, Visakhapatnam, India
E-mail: chiru404@gmail.com

Received: 15 May 2024; **Revised:** 1 July 2024; **Accepted:** 12 July 2024

Abstract: In applications involving image processing, segmentation is an essential stage. This procedure divides the image's pixels into various classes, enabling the examination of the scene's objects. Finding the ideal collection of thresholds to correctly segment each image is a challenge that multilevel thresholding solves with ease. The optimal thresholds can be found using methods like Otsu's between-class variance or kapur's entropy, but they are computationally costly when there are more than two thresholds. This study presents a novel meta-heuristic algorithm, Election-Based Optimization Algorithm (EBOA) to discover the optimal threshold configuration with Otsu as the objective function, to solve this kind of problem. The obtained results proved better in WPSNR, PSNR, SSIM, FSIM and misclassification error and segmented image quality when compared with existing algorithms.

***Keywords*:** image segmentation, Otsu's entropy, multi-level thresholding, optimization, image

1. Introduction

Thresholding is a popular segmentation technique [1] since it is irrespective of image size as well as being simple to use. It is separated into two categories: multi-level and bi-level [2]. The image is divided into two parts using bi-level and Multilevel thresholding. These are used to divide complicated images and can generate various thresholds like tri or quad level, which separates pixels into multiple related portions according to intensity. When the image has only two primary grey levels, bi-level thresholding can yield appropriate results; nevertheless, the main limitation is the time-consuming computation, which is frequently high when expanded to multilevel thresholding. The first step involves some statistical parameters (such as time consumption and accuracy) for the classes in the image must be computed. In the second level, optimal thresholds are chosen by maximizing some criterion such as Otsu' criteria [3] as well as the Kapur's entropy [4]. The optimal threshold in the Otsu-based method is the one that maximizes the between-class variance of the areas. By maximizing the entropy, Kapur's entropy determines the homogeneity of the classes. As a result, selecting optimal thresholds in multilevel is an NP-hard task that has long been regarded as challenge [5]. Classical procedures are appropriate when the threshold number is modest; however, as this number rises, the segmentation accuracy suffers. As a result,

DOI: https://doi.org/10.37256/rrcs.3220244958

to circumvent this weakness, many swarm AI approaches have been used and integrated with thresholding algorithms, such as the Genetic Algorithm [6]. Hybrid meta-heuristics are the most intriguing contemporary concepts in optimization and memetic algorithms [7]. Depending on the Whale Optimization Algorithm (WOA), two hybridization models are employed to construct various feature selection approaches. In the first model, the Simulated Annealing (SA) method is included in the WOA algorithm, but in the second model, it is used to optimize the optimal solution discovered after every iteration of the WOA process [8]. The Gravitational Search Algorithm (GSA) is a novel meta-heuristic method that relies on Newtonian gravity, although it offers the advantages of speedy convergence and reduced computational cost and it can occasionally become trapped in a local optimum which resulting in poor solution precision. As a result, a variety of variations of GSA had been described throughout the literature by adjusting its parameter's such as 'position, velocity, gravitational constant, and *Kbest*' [9]. *Kbest* is a vital function which balances both exploitation and exploration in GSA. In practice, it is a linearly declining task that shows the number of objects at a given iteration by imparting gravitational effect. As a result, it is accountable for the GSA's straight shift from exploration towards exploitation. After some rounds, the *Kbest* function was adjusted to improve exploitation gradually throughout exploration. On the other hand, the improved *Kbest* is also a linearly declining basis. Currently, chaotic *Kbest* GSA ('*cKGSA*') has been presented, which modifies *Kbest's* linear diminishing to chaotic behavior. As *cKGSA* has superior precision which has faster convergence rate and a stronger global search capability. Nevertheless, the chaotic behavior's reliance on initial conditions may have an impact on the outcomes.

Finally, the highlights of the paper are as follows:

- We introduced a new optimization algorithm, Election-Based Optimization Algorithm (EBOA), for optimal thresholds which helps to get good image segmentation results.
- EBOA algorithm was introduced based on the leader selection in general elections
- Efficiency of algorithm was tested on five standard images such as Camera, Goldhill, Lena, Pirate and Starfish.

The rest of the paper is organized as follows: Section 2 provides a literature review on image segmentation specially focusing on multilevel image thresholding with optimization algorithms. Section 3 describes the detailed explanation about the EBOA with mathematical representations. Section 4 gives the results, discussions and finally last section ends with conclusions.

2. Related Work

The Whale Optimization Algorithm (WOA) and the Moth-Flame Optimization (MFO) are two novel swarm algorithms that have been recently developed. Mirjalili and Lewis [10] propose the WOA tackle the global optimization problem by imitating the behaviors of humpback whales. Bubble-net feeding is a distinctive hunting tactic used by humpback whales. This behavior occurs in three stages: coral loop, lobtail, and capture loop. Mirjalili [11], on either hand, has developed MFO, which is a new meta-heuristic algorithm based on moth behavior. The WOA and MFO algorithm performance has been assessed based on two measures, PSNR and SSIM. Eight benchmark images are used in experiments. Paper [12] investigated two important meta-heuristic algorithms, namely the Quantum Genetic Method (QGA) as well as the Differential Evolution algorithm (DE) were used. Regarding image segmentation, two-dimensional R' enyi entropy and Tsallis entropy are employed with various forms of gray-level histograms. Extensively evaluated and refined the two-dimensional histogram computation and precisely describes the two-dimensional R' enyi and Tsallis entropies within the context of multilevel thresholding.

According to recent research, when it comes to bi-level thresholding, the outcomes of 2D histogram-oriented methods outperform those of 1D histogram-based methods. Using the Tsallis entropy, [13] proposed an approach for incorporating 2D histogram-related metadata for generalized multilevel thresholding. To enhance the processing efficiency of the approach, differential evolution (DE) which is a basic efficient evolutionary algorithm of recent interest is used. DE's functionality is thoroughly explored by comparing it to other well-known nature-inspired global optimization approaches including the genetic algorithm, particle swarm optimization, artificial bee colony and simulated annealing. Furthermore, the suggested method is tested against a well-known benchmark, the Berkley segmentation data set (BSDS300) which contains 300 different images. Genetic algorithms (GAs) are meta-heuristic optimization techniques that fall within the evolutionary algorithm category (EAs) [14]. They are motivated by the parallel between both the optimization process and organism evolution. When there is no deterministic technique or the deterministic method is computationally costly, a GA is employed to explore global and optimal solutions. Obtaining more than one threshold equals multiple outcomes. Every solution is symbolized by a chromosome, which is made up of genes and the responses created by all iterations are referred as the population. The population size refers to the number of possibilities of every iteration. Let 'n' be the number of arbitrarily produced individuals in the population. The genetic algorithm begins using 'n' randomly generated solutions. The best member solutions are then chosen to produce new solutions and the finest produced solutions are included in the subsequent iteration, while all poor solutions are excluded. The optimal solution is chosen within every generation based on the best fitness evaluation values of everyone in the population to generate a new population. The halting criterion is determined by the number of generations developed or by a suitable fitness value achieved for the population. The four stages of a genetic algorithm are typically population initialization, fitness assessment, reproduction and termination criterion.

We also suggested using advanced optimization algorithms like quantum-based avian navigation [15] and starling murmuration optimizer [16] for the same problem dependence on their performance in terms of metrics. In literature, image segmentation with optimized multilevel thresholds were obtained with different optimization algorithms and their hybridizations [17, 18]. For example, An improved African vulture's optimization algorithm with different cost functions [19], Snake Optimization [20], Hybrid optimization algorithms such as Arithmetic and Harris Hawks [21], A hybrid golden jackal and sine cosine algorithm [22].

3. Methods and materials

Image segmentation is a very important area for several applications, especially in the application of medical images for disease diagnosis, satellite images segmentation for floods detection etc. Image segmentation with image thresholding is very crucial because of the selection of appropriate thresholds. In this paper, these thresholds are selected based on optimizing the Otsu´s entropy fitness function by means of newly developed EOBA optimization algorithm.

3.1 *Election-based optimization algorithm*

This fragment will initiate the hypothesized Election-Based Optimization Algorithm (EBOA) and thereafter analyze it mathematically.

3.2 *Motivation*

An election is a procedure through which members of a community choose one of several nominees. The person elected as the community's leader has an impact on all the individuals, along with those who didn't cast votes for him. As more people in the community become aware, the better they will be able to select and vote for the most qualified candidate. The EBOA was designed with these defined principles of the election as well as voting procedure in mind.

3.3 *Algorithm initialization*

EBOA is indeed a population-based meta-heuristic algorithm including key stakeholders. Every individual from the population in the EBOA shows a possible answer to the problem. The EBOA population is mathematically portrayed by a matrix, which is described by Equation (1).

$$F=\begin{bmatrix}F_1\\ \vdots\\ F_i\\ \vdots\\ F_P\end{bmatrix}_{P\times q}=\begin{bmatrix}f_{1,1} & \cdots & f_{1,j} & \cdots & f_{1,q}\\ \vdots & \ddots & \vdots & \cdots & \vdots\\ f_{i,1} & \cdots & f_{i,j} & \ddots & f_{i,q}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ f_{P,1} & \cdots & f_{P,j} & \cdots & f_{P,q}\end{bmatrix}_{P\times q} \tag{1}$$

wherein F is EBOA population matrix, and F_i refers to i^{th} EBOA member (i.e., the proposed solution), The values $f_{i,j}$ represent the j^{th} problem variable supplied by the i^{th} EBOA member, P represents the size of the EBOA population, q represents the number of decision variables and Individuals' beginning positioning in the search space is decided at randomness as stated by Equation (2)

$$f_{i,j}=k\cdot b_j+z\cdot[vb_j-kb_j]\quad i=1,2,\ldots,P,j=1,2,\ldots,q \tag{2}$$

while $k{\cdot}b_j$ and $v{\cdot}b_j$ denote the lower and upper bounds of the j^{th} variable, and z is a random number in between [0, 1]. A value for the optimization problem can be calculated using the values recommended by each EBOA member for the problem variables. A vector is used to specify these estimated quantities for the issue's objective functions stated by (3).

$$O_bF_n=\begin{bmatrix}O_bF_{n1}\\ \vdots\\ O_bF_{n2}\\ \vdots\\ O_bF_{nP}\end{bmatrix}_{P\times 1}=\begin{bmatrix}O_bF_n\left(F_{n1}\right)\\ \vdots\\ O_bF_n\left(F_{n2}\right)\\ \vdots\\ O_bF_n\left(F_{nP}\right)\end{bmatrix}_{P\times 1} \tag{3}$$

where O_bF_n, O_bF_{ni} are the vectors of EBOA populations acquired objective function values, derived objective function value for the i^{th} EBOA individual. The results of the objective function serve as a criterion for assessing the quality of recommended responses along with the optimum values of objective function indicating both the highest and least member.

3.4 *Mathematical modeling of EBOA*

The main distinction between metaheuristic algorithms is how individuals of the population are upgraded and offered solutions improve with each iteration. The method of updating the algorithm population in EBOA is divided into two steps, as explained below.

3.4.1 *Phase-1*

3.4.1.1 *Voting procedure and holding elections (exploration)*

Depending on their level of understanding, EBOA members vote for one of the nominees. The validity and fairness of the objective function's value can be thought of as influencing people's consciousness. As a result, community members' consciousness is simulated using Equation (4). Individuals who have higher levels of the objective function are far more conscious during simulation process.

$$R_i=\begin{cases}\frac{O_bF_n-O_bF_{n(least)}}{O_bF_{n(highest)}-O_bF_{n(least)}} & O_bF_{n(highest)}\neq O_bF_{n(least)}\\ 1 & otherwise\end{cases} \tag{4}$$

whereas R_i is the consciousness of the i^{th} EBOA member, $O_bF_{n(highest)}$ and $O_bF_{n(least)}$ are the objective function's highest and least values, accordingly. It needs to be remembered that in minimization issues, $O_bF_{n(highest)}$ represents to the objective function's minimum value and $O_bF_{n(least)}$ represents to the objective function's maximum value, in maximizing issues, whereas $O_bF_{n(highest)}$, $O_bF_{n(least)}$ represents the highest and least values of the objective function.

Among the members of society, 10% of the most aware individuals are considered election contenders. In EBOA, the least number of candidates (P_D) is 2 (i.e., $P_D \geq 2$), implying that as a minimum two competitors will enroll for the election. In EBOA, the voting process is implemented in such a way that everyone's consciousness level is contrasted to a random value; if a person's consciousness level is larger than that of random number, the individual can vote for the better candidate (called as D_1). Simultaneously, that individual will vote for another contender. This procedure has been quantitatively modelled in Equation (5).

$$U_i = \begin{cases} D_1 & R_I > Z \\ D_l & else \end{cases} \tag{5}$$

whereas U_i denotes to the vote of the i^{th} person in the community, D_I denotes to the better candidate, and D_k signifies to the k^{th} candidate, where l is a randomly selected number from the set $\{2, 3..., P_D\}$.

After the end of the election process, depending upon the total number of votes cast, the candidate with the majority votes is declared as the leader who influences all the individuals in society and those who didn' t cast a vote.

Individual positions in the EBOA are revised under the supervision and instruction of the selected leader. This leader leads the algorithm population to several sections of the search space, raising the EBOA's capability to explore in global search. The leader leads the procedure of upgrading the EBOA population, starting with the creation of a new position for every individual. If the newly created location enhances the value of the target function, it is suitable for revising. Simultaneously, the preceding location is kept by the matching individual. The EBOA's updating mechanism is utilized by Equations (6) and (7)

$$f_{i,j}^{latest,P_e1} = \begin{cases} f_{i,j} + z \cdot (H_j - J \cdot f_{i,j}), & O_bF_{nH} < O_bF_n \\ f_{i,j} + z \cdot (f_{i,j} - H_j), & otherwise \end{cases} \tag{6}$$

$$F_i = \begin{cases} F_i^{latest,P_e1}, & O_bF_n^{latest,P_e1} < O_bF_n \\ F_i, & otherwise \end{cases} \tag{7}$$

whereas F_i^{latest,P_e1} refers to a newly generated position for the i^{th} EBOA member, $f_{i,j}^{latest,P_e1}$ is its j^{th} dimension, $O_bF_n^{latest,P_e1}$ is its worth of the objective function, J is the integer chose at random from the values 1, 2. H referred to the elected leader, H_j is its j^{th} dimension, and O_bF_{nH} is its objective function value.

3.4.2 *Phase 2*

3.4.2.1 *A public awareness campaign (exploitation)*

Increase in awareness had a significant effect on their effective conclusions in the elections and casting method. Aside from the leader's impact on public consciousness, any individual's attitude and activity can raise that individual's consciousness. A superior response could be mathematically calculated by doing a local search close to several contributed solution. As a result of community members' efforts to raise awareness, the EBOA's ability to utilize the local search and find better approaches to the issue that grows. To replicate this local search process, a random place in the search space's neighborhood is evaluated. The objective function of the problem is then assessed considering this new circumstance to decide whether it is superior to the member's current condition. The local search is effective if the new role has a greater quality for the objective function, and the location of the relevant member is upgraded. Enhancing the value of the goal functions that will raise the person's consciousness, allowing them to make better decisions during the afterwards election (iteration). This method is about the awareness of EBOA by using Equations (8) and (9).

$$f_{i,j}^{latest,P_e2} = f_{i,j} + (1-2z) \cdot G \cdot \left(1 - \frac{t_{ic}}{T_{maximum}}\right) \cdot f_{i,j} \tag{8}$$

$$F_i = \begin{cases} F_i^{latest,P_e2}, & O_bF_n^{latest,P_e2} < O_bF_n \\ F_i, & otherwise \end{cases} \tag{9}$$

whereas F_i^{latest,P_e2} referred to a new generated position for the i^{th} EBOA member, $f_{i,j}^{latest,P_e2}$ is its j^{th} dimension, $O_bF_n^{latest,P_e2}$ is its value of the objective function, G is the constant which equals 0.02, t_{ic} describes to iteration contour, and $T_{maximum}$ denotes to highest number of iterations.

3.4.2.2 *The EBOA process is as follows*

Step-1: Define the following details about the optimizer: objective function, restrictions and number of choice-variables.

Step-2: Modify the number of algorithm iterations (T) and Population size (N).

Step-3: Randomly seed the EBOA population and estimate the objective function.

Step-4: Revise EBOA population's finest and worst individuals.

Step-5: Compute the community's consciousness vector.

Step-6: Choose individuals from the EBOA community.

Step-7: Keep the Voting process going.

Step-8: Depending on voting results, ascertain the elected representative.

Step-9: Modify EBOA members' positions in the search space depending on elected leader direction.

Step-10: To increase awareness, modify the status of EBOA members according to the notion of local search and public activity.

Step-11: As the best qualified solution thus far, keep the best EBOA member.

Step-12: If algorithm's iterations are complete, proceed to the subsequent or else return to step-4.

Step-13: In the result, display the most suitable candidate solution.

End.

The flowchart of deployment of the EBOA is listed and shown in Figure 1 and its pseudo-code is represented in Algorithm 1.

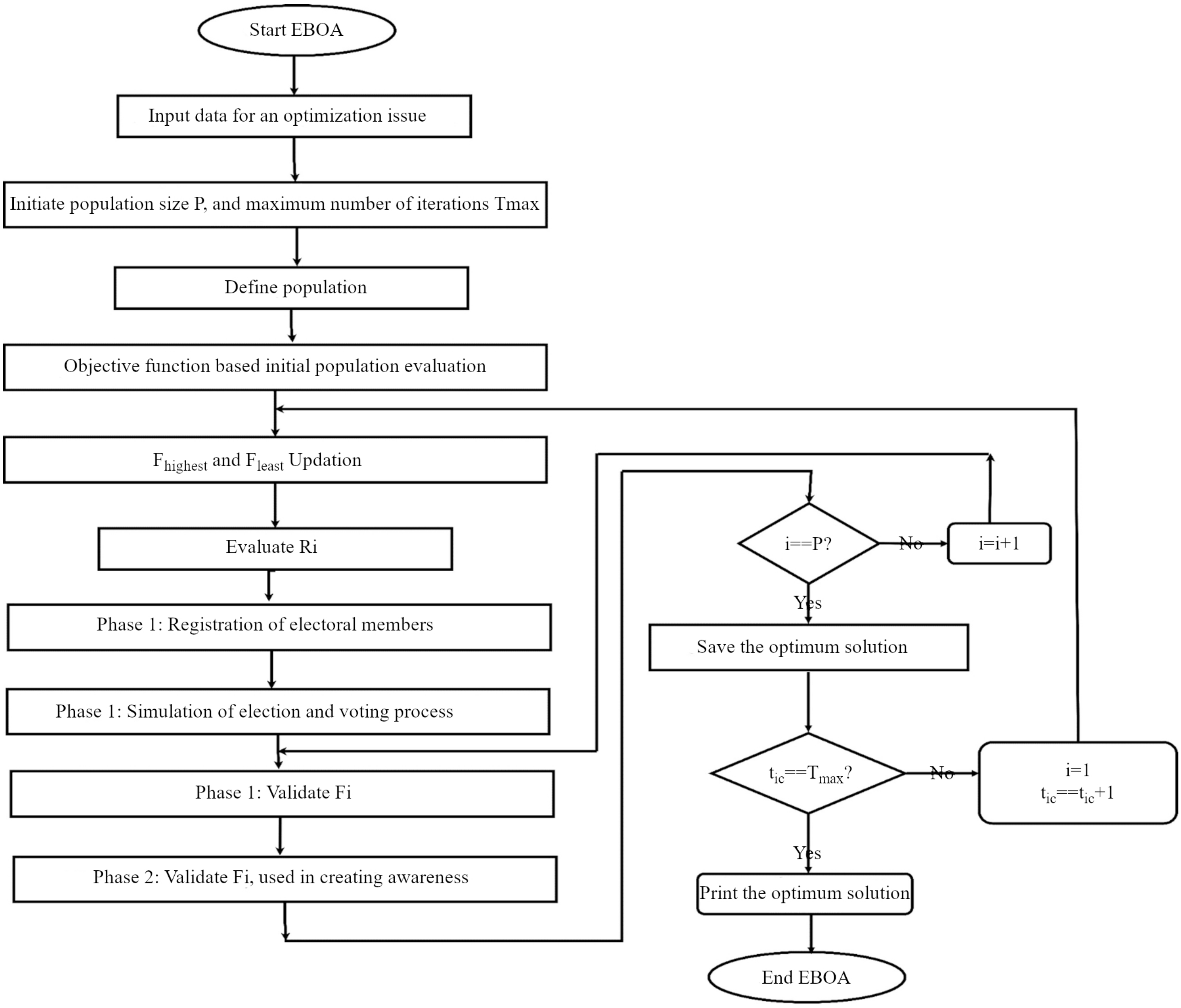

Start EBOA
Input data for an optimization issue
Initiate population size P, and maximum number of iterations Tmax
Define population
Objective function based initial population evaluation
$F_{highest}$ and F_{least} Updation
Evaluate Ri
Phase 1: Registration of electoral members
Phase 1: Simulation of election and voting process
Phase 1: Validate Fi
Phase 2: Validate Fi, used in creating awareness
i==P?
No
i=i+1
Yes
Save the optimum solution
t_{ic}==T_{max}?
No
i=1
t_{ic}==t_{ic}+1
Yes
Print the optimum solution
End EBOA

Figure 1. Flowchart for EBOA implementation

3.5 *Repetition process, pseudo-code, and flowchart of EBOA*

On modifying the position of all population members, an EBOA iteration is completed. The EBOA initiates further iteration with the freshly updated values and the population update process is repeated in accordance with Equations (4)–(9) until the final repetition. EBOA offers the most proposed outcome attained throughout the algorithm iterations as the solution to the difficulty after the full implementation of algorithm is finished.

Algorithm 1

```
Initiate EBOA process:
Enter the input-problem information: Objective function, variables and constraints.
Decide EBOA population size (P) and iterations (Tmaximum).
First, create a random population matrix.
Calculate the fitness function.
for tc = 1 to Tmaximum do
    Upgrade the optimum and least population partners.
    Level-1: Voting procedure and holding elections i.e., exploration.
    Find out R value utilizing Equation (4).
    Find the individuals depending on consciousness factors.
    Using Equation (5), simulate holding elections and voting.
    Count the ballots and choose the victor of the elections to be the leadership.
    for i = 1 to P do
        Evaluate $f_{i,j}^{latest,P_e1}$ using Equation (6)
        Update $F_i$ making use of Equation (7)
    end for
    Level-2: Public movement to increase attention i.e., exploitation.
    Evaluate $F_i^{latest,P_e2}$ using Equation (8)
    Update $F_i$ using Equation (9)
end for
Terminate
Store the optimum suggested result so far.
Terminate.
Output best quasi-optimal solution acquired through the EBOA.
Terminate EBOA.
```

3.6 *Computational—complexity of EBOA*

The computational cost of EBOA commencement, comprising arbitrary population creation and first assessment of the objective function is equal to O_b (Pq).

Where P = size of the EBOA population, q = the number of problem variables.

The computational complexity of holding election and upgrading EBOA population in the first level is equivalent to O_b $(PqT_{maximum})$, where $T_{maximum}$ refers the number of iterations. The population upgrade depends on second level of EBOA to raise awareness is equal to O_b $(PqT_{maximum})$. As a result, the total computational complexity of EBOA is O_b $(Pq(1+2T_{maximum}))$.

4. Results and discussions

4.1 *Performance of Otsu-thresholding technique*

The Otsu method says on thresholding 1-D intensity data that is taken from a 2-D image [23, 24]. The configuration yields straightforward calculation that is reliable and versatile for use in a variety of computer vision applications [25]. We considered a real-world scenario where 2D images are divided into two-pixel groups that represent the segmented object and backdrop. The Otsu-technique consist of four fields, which effect Otsu thresholding [26, 27] and are frequently seen in scenes. These four parameters are:

(a) The contrast in object-background intensity reflects the scene's lighting conditions.

(b) Size and placement of the object defining the location and separation of the object from the camera

(c) Noise estimation used to assess the technical performance of cameras.

Other factors including geometric characteristics of the object which are anticipated to give no impact on threshold performance because this technique executes threshold which depends on data intensity. In the investigation, Otsu thresholding approach was applied using a publicly accessible implementation [25]. The Otsu-thresholding approach is used as metric to distinguish both images (the segmented and original gray-scale images). The performance of this thresholding approach should be comparable to that of previous lower-level thresholding algorithms. When modeling an image,

$$I = f\left(I^{OR}\right) = f\left(I_b^{OR} + I_f^{OR}\right) \tag{10}$$

where: I^{OR} signifies the original image,
I_b^{OR} signifies the background image,
I_f^{OR} signifies the foreground image,
$f\left(I^{OR}\right)$ denotes the acquisition transformation.
Noise may be described as:

$$I(x,y) = f\left(I^{OR}(x,y)\right) = Ns(x,y)\left[\left(I_b^{OR}(x,y) + I_f^{OR}(x,y)\right)\right] \tag{11}$$

$$Ns\left((x,y)\left(I_b^{OR}(x,y)\right)\right) + Ns\left((x,y)\left(I_f^{OR}(x,y)\right)\right) \tag{12}$$

where: $Ns(x,y)$ is the noise, assuming the pixels of a specified image (I) denoted in 'L' Gray levels [1, 2, 3, 4, ……, L]. The number of pixels at stage *ip* is represented in n_{ip} and the overall number of pixels $N = n_1 + n_2 + n_3 + \ldots + n_L$.

The obtained *probability-distribution* is.

$$P_{ip} = \frac{n_{ip}}{N}; P_{ip} \geq 0; \sum_{ip=1}^{L} P_{ip} = 1 \tag{13}$$

The pixel is classified into two categories C_0 and C_1 by a threshold at level k_T. The C_0 represents pixels at level [1, …, k_T] similarly C_1 represents pixels at level [k_T + 1, ……, L].

The probabilities that a class will occurrence are determined as

$$W_0 = \sum_{ip=1}^{k_T} P_{ip} = \omega\left(k_T\right) \tag{14}$$

$$W_1 = \sum_{ip=k_T+1}^{L} P_{ip} = 1 - \omega\left(k_T\right) \tag{15}$$

where: W_0 and W_1 are the chances of C_0 and C_1 occurrence.

The class means levels are given by

$$\mu_0 = \sum_{ip=1}^{k_T} \frac{ip_1}{\omega_0} \tag{16}$$

$$\mu_1 = \sum_{ip=k_T+1}^{L} \frac{ip_{ip}}{\omega_1} \tag{17}$$

where, $\mu_0 = meanC_0$, $\mu_1 = meanC_1$

$$\omega(k_T) = \sum_{ip=1}^{k_T} P_{ip} \tag{18}$$

$$\mu(k_T) = \sum_{ip=1}^{k_T} ipP_{ip} \tag{19}$$

are the 0^{th} and 1^{st} order cumulative moments from the histogram is up to the k^{th} level.

The overall mean-level k_T of the original image is denoted as Equation (20)

$$\mu_T = \sum_{ip=1}^{L} ipP_{ip} \tag{20}$$

The class variance depends on Equations (21) and (22)

$$\sigma_0^2 = \sum_{ip=1}^{k_T} (1-\mu_0)^2 \frac{P_{ip}}{\omega_0} \tag{21}$$

$$\sigma_1^2 = \sum_{ip=k_T+1}^{L} (1-\mu_1)^2 \frac{P_{ip}}{\omega_1} \tag{22}$$

To evaluate the threshold the discriminant criterion is used.

$$\lambda = \frac{\sigma_B^2}{\sigma_\omega^2}; k_T = \frac{\sigma_T^2}{\sigma_\omega^2}; \eta = \frac{\sigma_B^2}{\sigma_T^2} \tag{23}$$

where,

$$\sigma_\omega^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \tag{24}$$

$$\sigma_B^2 = \omega_0\omega_1(\mu_1-\mu_0)^2 \tag{25}$$

$$\sigma_T^2 = \sum_{ip=1}^{L} (1-\mu_T)^2 P_{ip} \tag{26}$$

are within between class and total variance of levels correspondingly. η denotes the simplest measure concerning k_T, thus, it is implemented to assess the threshold (Th) at level k_T. The picture containing a foreground and background was created using sets of synthetic images that were used as inputs for the Otsu thresholding method. During the experiment, the scene and measurement parameters are changed by changing the features of the synthetic images, such as:

1) I stand for the intensity differential between pixels pertaining to the object and the backdrop. When the intensity of the pixels corresponding to the object and backdrop are identical, the result is I = 0%. Conversely, I = 100% denotes a black background and a white object or vice versa.

2) The dimensions of the object indicated by S = l × l, represented the object dimensions as it appeared on the image plane. For convenience of analysis, we used a 50 × 50-pixel square image. The entire image is covered by 50 × 50 pixels, as shown by the Object Size, S = 100%. In contrast, S = 20% means that, out of a 50 × 50 square image, the object size l × l is 10 × 10 pixels.

3) P stands for the object location that was visible on the image plane. We adjusted the object's position at various points in the artificial image. These objects can be found in the image plane's center as well as the first, second, third, and fourth quadrants.

4) Measurement of noise is represented by σ^2. A typical normal distribution noise included to the image plane with 50 × 50 pixels. It is assumed that every point or pixel in the noise-contaminated input image plane has an independent, identical distribution with a mean of $\tilde{x} = 0$ and a variance of $\sigma^2 = 1$. Next the Otsu was used to introduce the synthetic image into the thresholding process. Thresholding execution was also calculated by the segmentation ratio ζ.

$$\zeta = \frac{no.\ of\ segmented\ points}{no.\ of\ true\ points} \tag{27}$$

If the segmentation ratio is equivalent to 1, it signifies that entire pixels related to objects that have been effectively segregated, which is the ideal state of segmentation. Meanwhile, occurrences of under or over-segmentation are indicated by values of ζ that are less than or greater than 1. Over segmentation means that some background pixels were segmented as objects, however under segmentation means that some pixels belonging to the object were not segmented.

$$segmentation\ ratio\ \zeta = \begin{cases} \zeta < 1, & under\ segmentation \\ \zeta = 1, & good\ segmentation \\ \zeta > 1, & over\ segmentation \end{cases} \tag{28}$$

Three sections contain the analysis of Otsu thresholding performance: the consistency analysis, the verification and demonstration using real picture data, and the feasibility analysis of thresholding.

4.1.1 *Feasibility analysis of the Otsu thresholding*

This study concentrates on the background-intensity, Object size and its position and noise during the first part of the threshold segmentation analysis. Since these are the fundamental parameters that would impact the performance of segmentation, they are proposed. In terms of object-position 'P' is at first fixed at the center of the image then changed at different quadrants. For back-ground intensity parameters, 'I' is shifting from value bit [0 to 1] with a rise of 20%. For object size, S is first fixed at [10 × 10] pixels and it is modified to [50 × 50] pixels with a rise of 20%. A normal allocated noise is introduced to the synthetic images used in theoretical experiments to study the conditions for effective image-segmentation in the noise factor. The noise density is shifting from σ^2 = 0.54 to assess the limitations of the proposed parameters at various noise levels. Without failure of originality, quadruplet factors will be verified using the Otsu-thresholding process to ensure robust estimation for real-world image applications. The purpose of the investigation is to investigate the theoretical bounds of Otsu thresholding as it relates to segmenting images using fixed parameters and known scenarios. The following results show how feasible each of the suggested factors such as back-ground intensity, object size, object position and noise. Synthetic reference image for background intensity parameter, pre-threshold image and thresholding result image are shown in Figure 2.

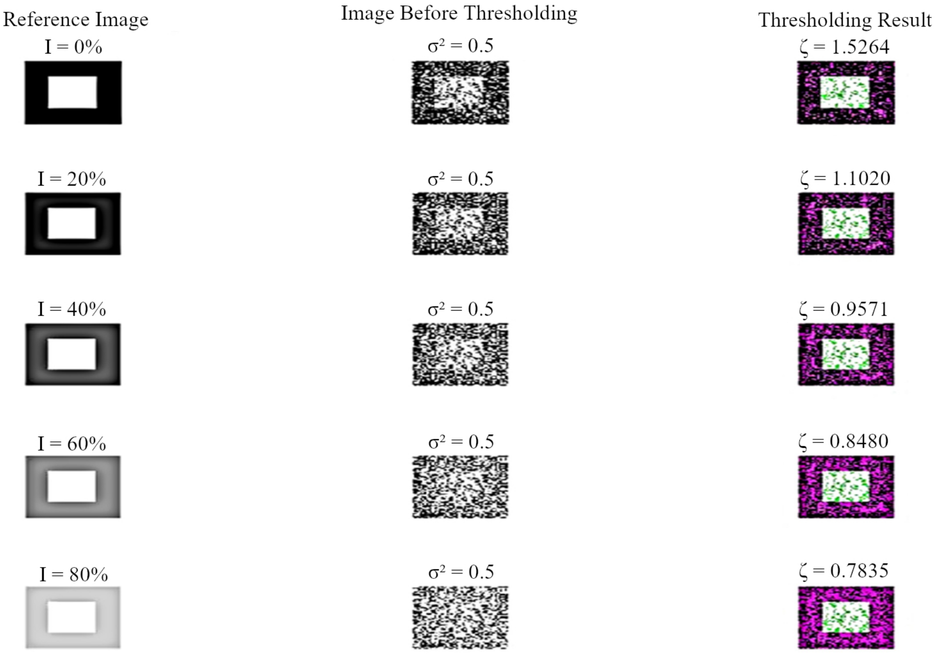


Figure 2. Synthetic-reference images for intensity factor, pre-thresholding image and thresholding result image which includes segmentation ratio ζ, *noise* σ^2, Intensity (I), Object size (S) and location (P).

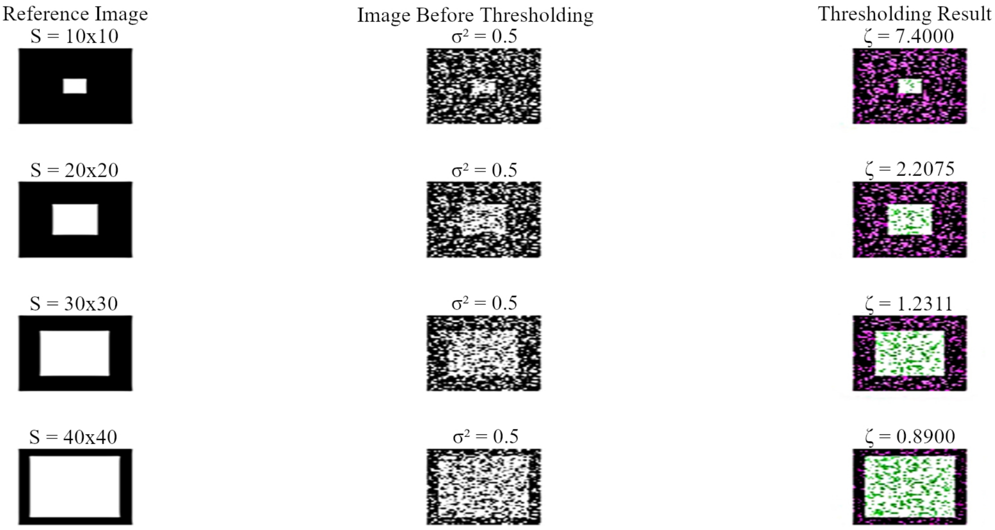


Figure 3. Synthetic-reference images for object size parameter, pre-threshold image and thresholding result image contains Segmentation ratio (ζ), *noise* σ^2, Intensity (I), object size (S) and location (P).

Theoretically, as the intensity difference increases, segmenting the image will become easy. Results shown in Figure 3 indicate that noise is controlled at $\sigma^2 = 0.5$. It is also clear that the segmentation percentage increases as it experiences over-segmentation, when the background intensity (I) is less than 40%. This shows that the inliers and outliers in appropriate to categorize as object and background due to improper segmentation. Following Otsu segmentation, some of the outlier points may unexpectedly become inlier points. In the same way, when background intensity (I) exceeds 40%, the portion of segmentation that experiences under-segmentation will also increase. Based on the outcomes shown in Figure 3, it can be observed that noise is controlled at $\sigma^2 = 0.5$ and that a higher percentage of over-segmentation occurs when the object size (S) is smaller than [40 × 40] pixels. This stated that the inlier and outlier could not be correctly separated as background and object due to inaccurate segmentation. This indicates that it generally becomes difficult to segment an object correctly when it looks smaller in the image plane. Because some of the outlier points might turn into inlier points during Otsu segmentation, this could result in over-segmentation.

Similarly, when the object (S) size exceeds [40 × 40] pixels, the percentage of segmentation that experiences under-segmentation will also be improved. This is because, in comparison to background points, object points have grown and snowballed. As a result, the possibility of inlier points at object sizes larger than [40 × 40] pixels will increase since improper segmentation. The synthetic reference image for the object position parameter, the image before to thresholding and the thresholding result image are represented in Figure 4.

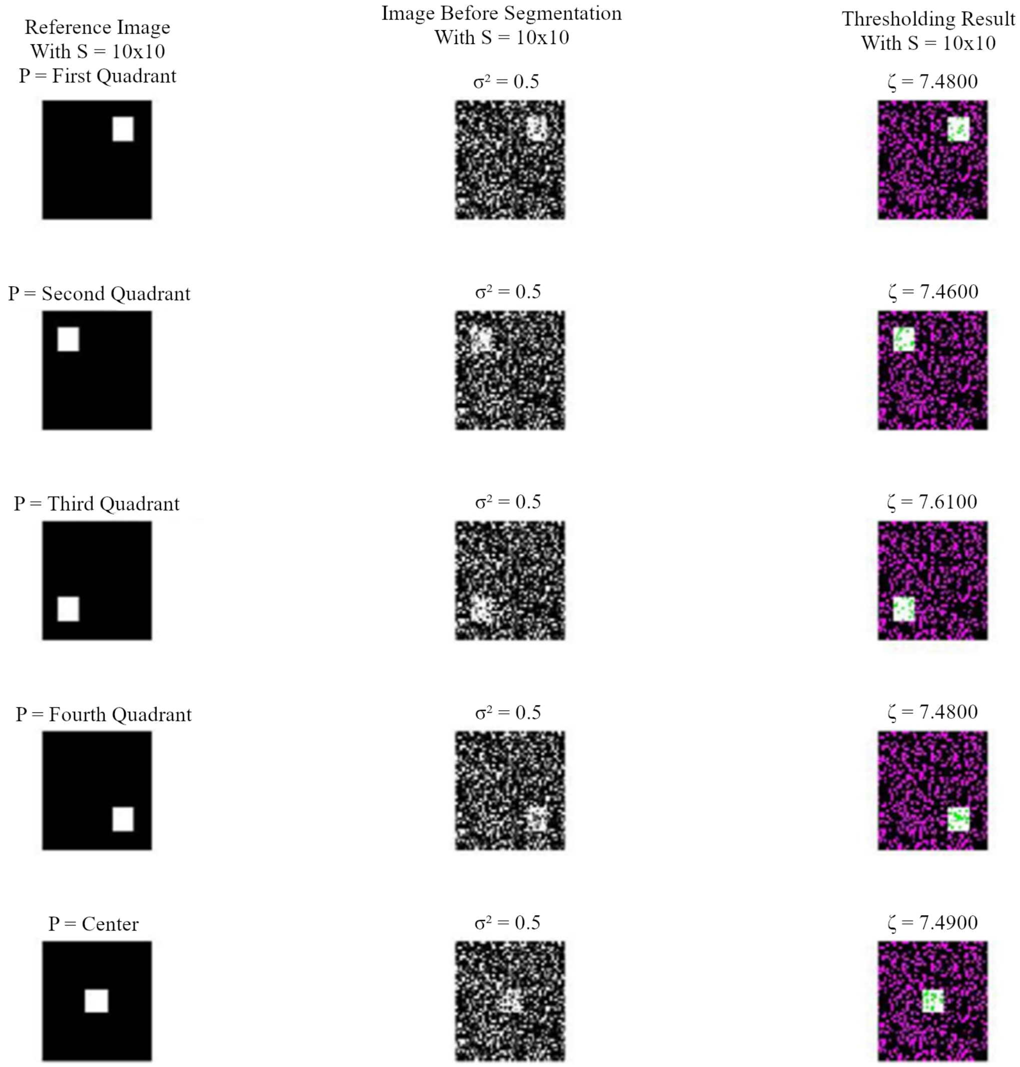


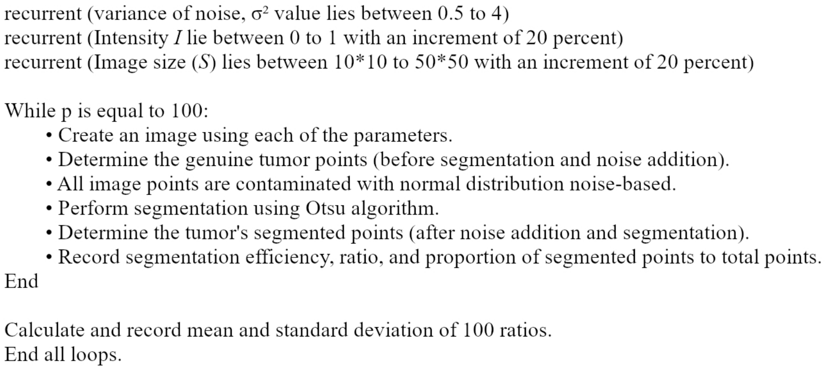
Figure 4. Synthetic-reference images for object position parameter, pre threshold image and thresholding result image includes Segmentation ratio (ζ), *noise* σ^2, Intensity (I), Object size (S) and location (P).

Theoretically, the object position in each image has no effect on thresholding performance unless the present object's position has various stages of light intensity. Figure 4 shows that when noise is controlled at $\sigma^2 = 0.5$, the object size (S) of 10 × 10 pixels gives consistent results across various positions.

Similar to this, the segmentation ratio outcomes are consistent at various object position parameters varied at below mentioned positions as (i) primary section ($\zeta = 7.4800$), (ii) secondary section ($\zeta = 7.4600$), (iii) tertiary section ($\zeta = 7.6100$), (iv) quaternary section ($\zeta = 7.4800$) and center ($\zeta = 7.4900$) of the synthetic images for object size 'S' equal to [20 × 20], [30 × 30] and [40 × 40] pixels respectively.

4.1.2 *Consistency analysis of the Otsu thresholding*

To check the accuracy of the feasibility analysis, consistency analysis is conducted. The experiment's design was based on statistical Monte Carlo methods. At random generated 100 synthetic image cases are given to Otsu method for each study. 100 values of ζ were recorded during the thresholding process and the consistency was calculated using the standard deviation and mean, represented by σ_ζ and $\tilde{\zeta}$ respectively. During the evaluation, the image segmentation generated by Otsu thresholding can be recognized as reliable, accurate only if the values of σ_ζ and $\tilde{\zeta}$ ≈ 0.01 and 1 respectively. Then, consistent analysis was carried out again for every parameter. Figure 5 includes a pseudo-code that summarizes the analysis setup.



```
recurrent (variance of noise, σ² value lies between 0.5 to 4)
recurrent (Intensity I lie between 0 to 1 with an increment of 20 percent)
recurrent (Image size (S) lies between 10*10 to 50*50 with an increment of 20 percent)

While p is equal to 100:
    • Create an image using each of the parameters.
    • Determine the genuine tumor points (before segmentation and noise addition).
    • All image points are contaminated with normal distribution noise-based.
    • Perform segmentation using Otsu algorithm.
    • Determine the tumor's segmented points (after noise addition and segmentation).
    • Record segmentation efficiency, ratio, and proportion of segmented points to total points.
End

Calculate and record mean and standard deviation of 100 ratios.
End all loops.
```

Figure 5. Pseudo-code for consistency analysis.

Using synthetic images, the consistency analysis results under various image parameters are presented in Figure 6. To ensure a successful image segmentation, Figure 6 presents various large images of the necessary conditions for the suggested parameters. The optimal requirements for every parameter are decided from the graphs.

The segmentation presentation for the back-ground intensity factors modified at various noise stage factors is shown in Figure 7a. Based on the graphical result, segmentation is most accurate at $\sigma^2 = 1.0$ and background intensity I = 40%, as illustrated by the segmentation ratios of $\tilde{\zeta} \approx 1$ and $\sigma_\zeta \approx 0.01$. This represents that the image segments effectively when the background's intensity is about 40%. Furthermore, Figure 7a shows that, except for I = 40%, then the segmentation is correct, entire back-ground intensity variations from 0 to 1 with a 20% increment are incorrectly segmented. Further, we have tested how noise variance affected the intensity parameter-based image segmentation. When compared to other noise variances, the mean segmentation ratio value of the graphical outcomes in Figure 7a at I = 40% indicated that all segmentation is true. As their intensity values are individualistic of object size, image size and object position, we can conclude from the analysis that the intensity impacted target segmentation.

Figure 7b shows the segmentation effectiveness for object size factor changed at various noise stage parameters. According to the graphical results, segmentation is best accurate when noise variance is equal to one and object size S = [40 × 40] pixels with the segmentation ratios of $\tilde{\zeta} = 1$ and $\sigma_\zeta \approx 0.01$. This implies that the image has efficient segmentation when the object size is about [40 × 40] pixels. Figure 7b shows that, except for S = [40 × 40] pixels, for which the segmentation is accurate, all object sizes from [10 × 10] to [50 × 50] pixels with a 20% increment show improper segmentation. We then looked at how noise variance affected image segmentation using the object size parameter. At S = [40 × 40] pixels, the graphic results in Figure 7b demonstrate that overall segmentation is inaccurate. Since, its average segmentation ratio value is nearer to 1 than among noisy variances. As a result of the investigation, we can say that target segmentation has been influenced by object size because these objects' sizes are separate of object position, image size and intensity. Through Figure 7c, it is possible to see the state of the segmentation performance for the object location parameter. According to the graphical results, the segmentation with object-position values result has no effect on the mean of the segmentation ratio for different object sizes when the noise variance is equal to 1. Additionally, it can be observed and noticed that the data results are nearly unchanged by shifting the position (primary, secondary, tertiary and quaternary quadrants and center) throughout the experiment. Therefore, we can conclude from a preliminary theoretical analysis that the segmentation performance is independent of the object location parameter.

According to the graphic results shown in Figure 7a–c, a Segmentation-ratio of one denotes successful separation of the foreground and background. For each parameter, these optimum values act as the prerequisites to provide accurate segmentation.

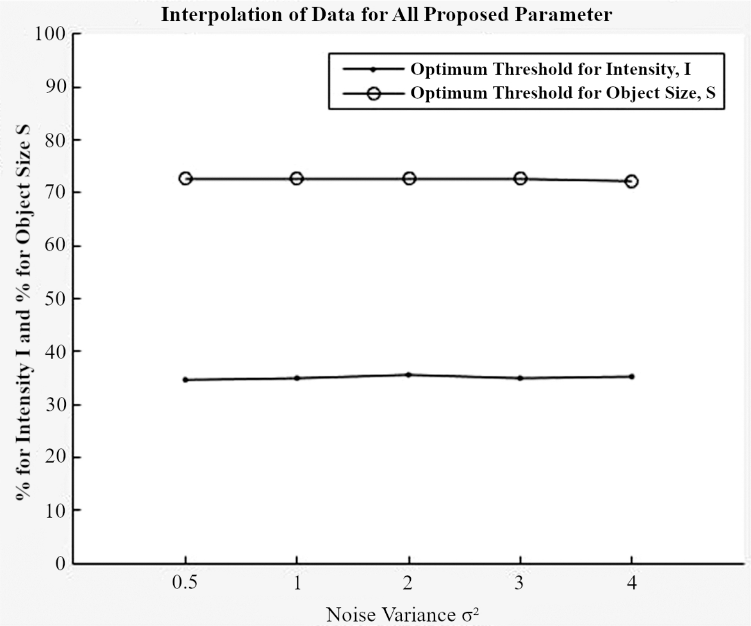


Figure 6. Data interpolation for every proposed parameter

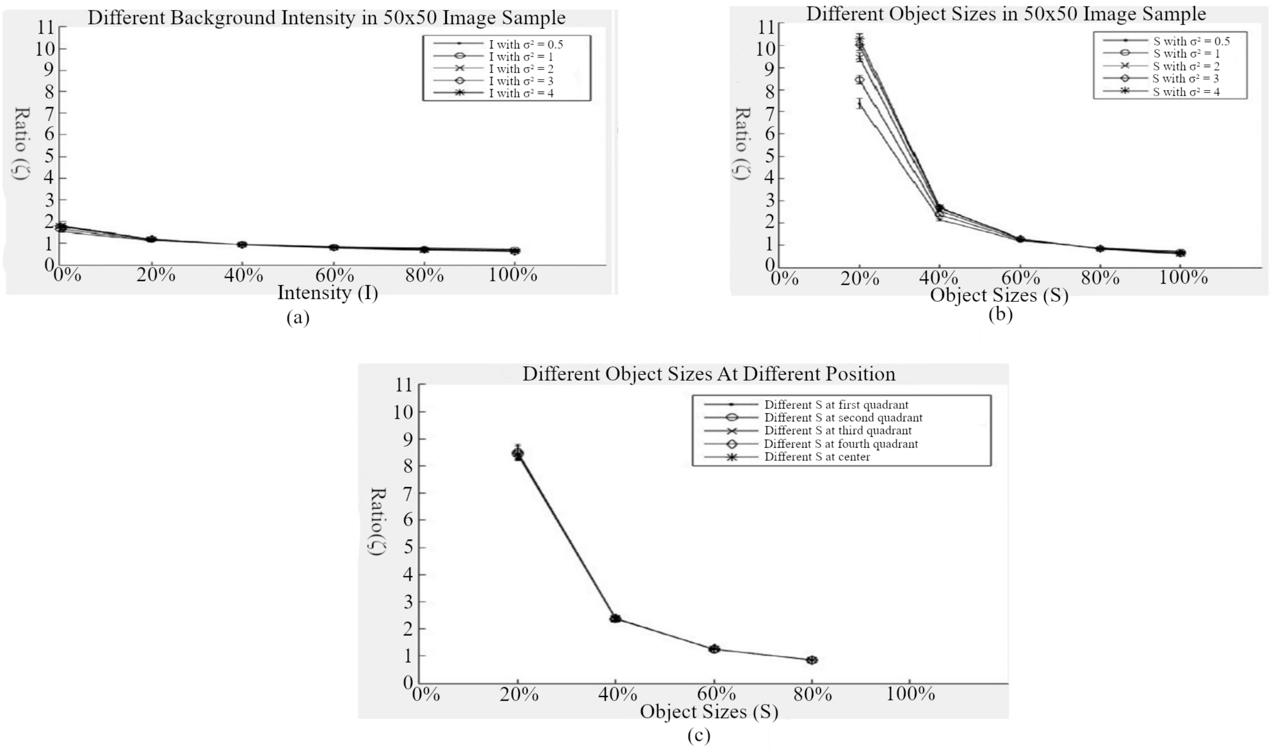


Figure 7. Indicates the segmentation performance for each of the proposed values. (**a**) The performance of segmentation $\widetilde{\zeta}$ and σ_{ζ} for back-ground intensity parameter at various levels of noise begin ($\sigma^2 = 0.5\sigma^2 = 4$), (**b**) Segmentation operation $\widetilde{\zeta}$ and σ_{ζ} for object size parameter at various noise levels from ($\sigma^2 = 0.5\sigma^2 = 4$), (**c**) Performance of segmentation $\widetilde{\zeta}$ and σ_{ζ} for position of the object parameter at various sizes of object from (S = 20% to S = 80%).

Figure 6 interpolated from Figure 7 and illustrates the requirements for successful Otsu thresholding with $\widetilde{\zeta} = 1$ and $\sigma_{\zeta} \approx 0.01$. It can be shown that for various noise levels, the intensity parameter and object size parameter ought to be adjusted at about 35% and 72%, respectively, to ensure successful thresholding. The object location parameter has no effect on the segmentation performance, as seen in Figure 7c. Since the object location parameter has little effect on the segmentation's overall performance, the remaining parameters like object size, background intensity and noise are involved in data interpolation process. Figure 6 can be used as a reference to ensure that Otsu thresholding is effective and efficient.

4.2 *Performance metrics*

The quality of Image segmentation can be measured in Peak Signal to Noise Ratio (PSNR), Structure Similarity Index Measure (SSIM) and Featured Similarity Index Measured (FSIM). The PSNR can be obtained with the following Equations (29) and (30).

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\mathrm{dB} \tag{29}$$

where MSE is the Root Mean Square Error calculated by

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I(i,j) - s(i,j)^2 \tag{30}$$

M, N represents the size of image, $I(i,j)$ is the unsegmented image and $S(i,j)$ is the segmented image with a given thresholding level. When the PSNR is lower, the segmentation quality is lower because there is less difference between the original image and the processed image. A higher value of PSNR indicates better quality of threshold image or reconstructed image. The SSIM is calculated from [27]. A SSIM near one indicates better results in segmentation and a value closer to zero indicates poor image quality.

Another metric is the Featured Similarity Index (FSIM) evaluates the importance of the local structure between the non-segmented and segmented image. FSIM is a method used to map the features and measures the similarity between images (i.e., the stored and the original image). It is used in quality image assessment. FSIM only considers the luminance component of the image. The maximum FSIM value that can be possible is 1 [27].

4.2.1 *Weighted peak signal to noise ratio (WPSNR)*

It is a method used for calculating weight factors to obtain visual sensitivity of the image regions. It is used to determine those sensitivity weights with higher spatial accuracy. The WPSNR was developed to solve the drawback of PSNR which exhibits weak performance in reflecting human discernment [28]. This method is computationally inexpensive and outperforms PSNR and SSIM when it comes to approximate subjective rating of JPEG2000. WPSNR can be calculated as.

$$\mathrm{WPSNR} = 10 \times \log_{10}\left(\frac{\left(2^{\mathrm{BD}}-1\right)^2}{\mathrm{MSE}^w}\right) \tag{31}$$

where BD = bit depth, MSE^w is weighted MSE

4.2.2 *Misclassification*

It is a phenomenon which occurs when individual data are assigned to a category other than one, they should be in. Misclassification can also be called incorrect classification [29]. On the other hand, misclassification rate indicates the percentage of observed outcomes that were incorrectly classified by a model. Larger value of misclassification rate indicates more incorrect classification done by the classification model. In simple form the misclassification rate can be calculated as.

$$Misclassification\ rate = \frac{False\ classifications\ (i.e.,\ false\ positive + false\ negative)}{Total\ predictions} \tag{32}$$

However, the opposite of misclassification rate is accuracy rate. Therefore, accuracy rate can be calculated as; Accuracy rate = 1 − Misclassfication rate.

4.3 *Discussion*

This sector gives the experiments and outcomes obtained with the proposed algorithms in the multilevel thresholding problem. The experimental conditions, the performance metrics used to estimate the performance of the different algorithms and the outcomes obtained with each algorithm in comparison with others are presented. The general standard images "Cameraman", "Goldhill", "Lena", "Pirate" and "Starfish" that are used for evaluating different algorithms and the corresponding histograms of the images are depicted in Figures 8a–e–12a–e respectively [30]. The experimental parameters are then compared among different algorithms and displayed qualitatively in Tables 1–5. The best values found at the end of the entire test are marked in bold for each thresholding level. MATLAB R2018a on Intel®CoreTM i5 processor-based Computer with 2.4 GHz CPU and 8 GB RAM has been used for the experiment. To compare the outcome of different algorithms, we evaluated them both visually and numerically. Figure 8 depicts the segmented version for different thresholding levels of 4, 5 and 6 for "cameraman" image. Figures 9–12 depicts segmented version for "Goldhill", "Lena", "Pirate" and "Starfish" images, respectively.

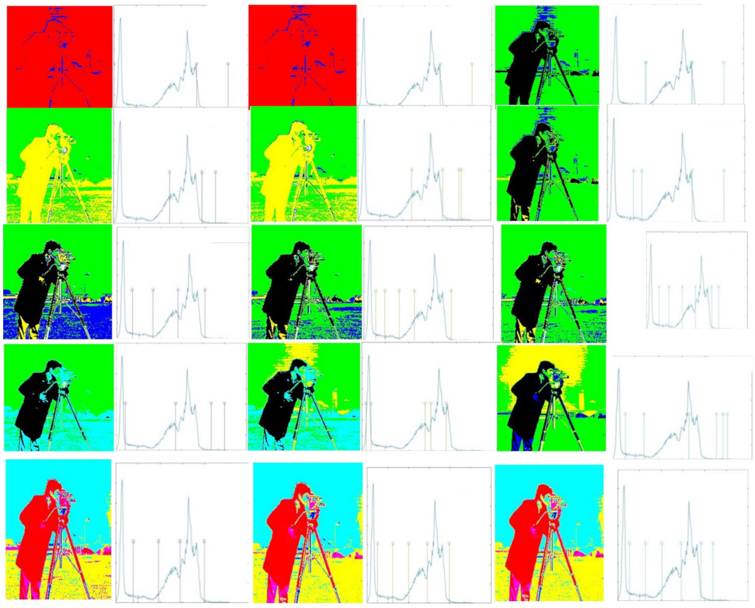


Figure 8. Cameraman: Segmentation using DE, PSO, BA, FA, GSA and EBOA algorithms along with 4, 5, 6 level thresholds and respective histograms

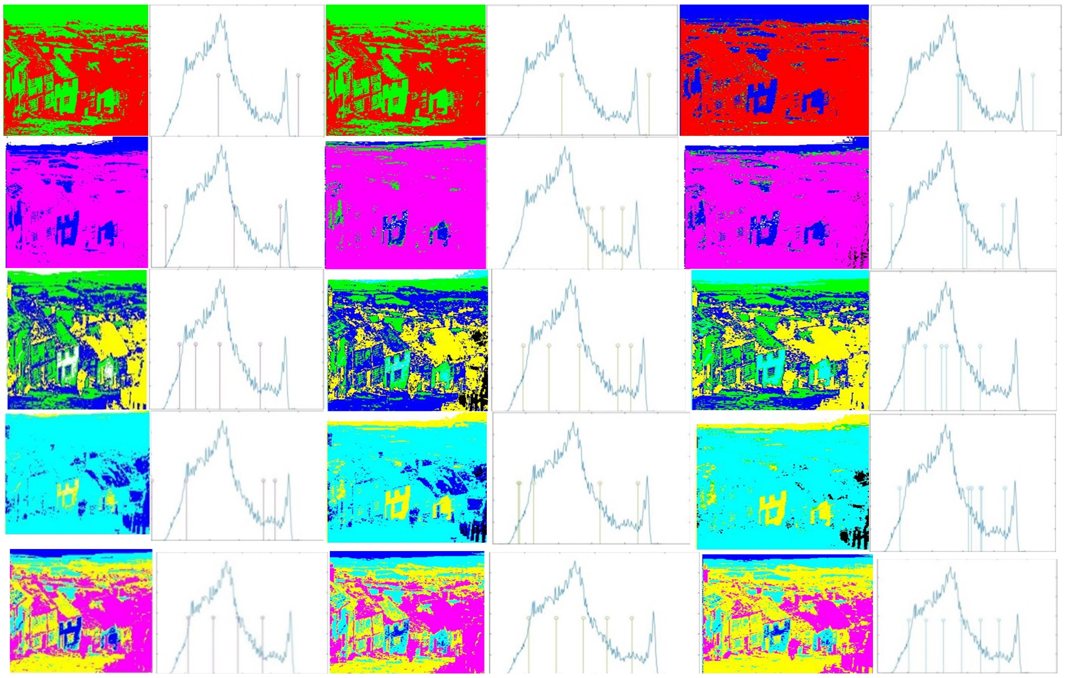


Figure 9. Goldhill: Segmentation using DE, PSO, BA, FA, GSA and EBOA algorithms along with 4, 5, 6 level thresholds and respective histograms

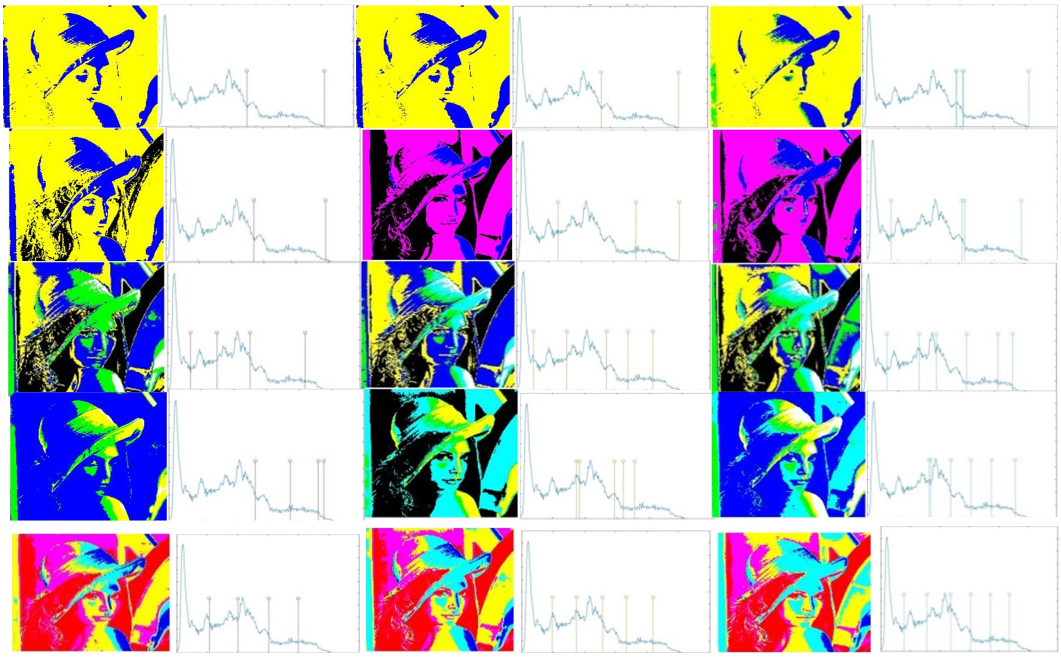


Figure 10. Lena: Segmentation using DE, PSO, BA, FA, GSA and EBOA algorithms along with 4, 5, 6 level thresholds and respective histograms

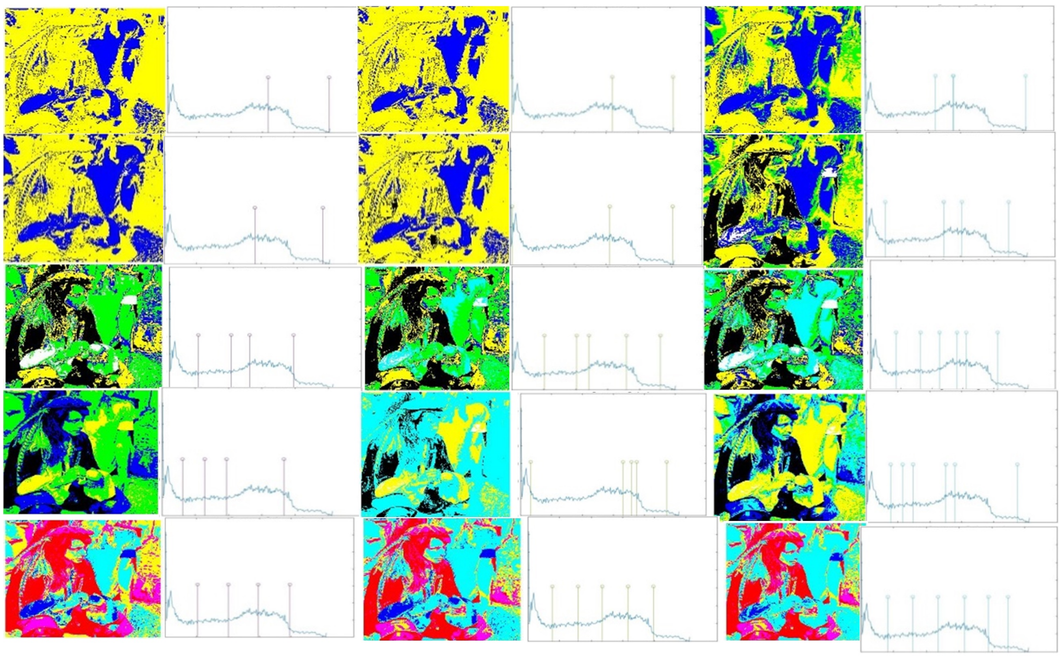


Figure 11. Pirate: Segmentation using DE, PSO, BA, FA, GSA and EBOA algorithms along with 4, 5, 6 level thresholds and respective histograms

Tables 1–5 deliver the numerical performance metrics of different algorithms for considered 5 images. We used the DE, PSO, BA, FA, CS, GSA and EBOA algorithms to optimize the related objective functions. These results were acquired by applying the algorithms to the 5 test images; we ran each algorithm over the test images. From Tables 1–5, reports the thresholds and the corresponding performance metric values, which were obtained and averaged from the runs for each algorithm on each test image with Otsu's objective function. PSNR is one of the metrics used to measure the quality of image segmentation. The dissimilarity between threshold image and input image can be shown by PSNR as a measure of visual difference of two images.

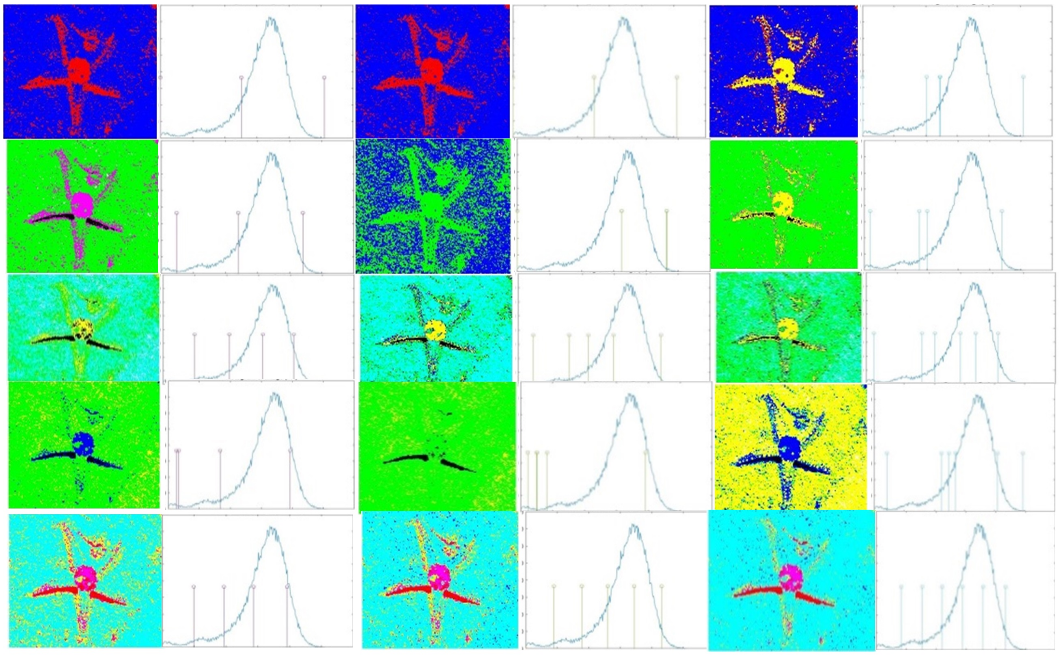


Figure 12. Starfish: Segmentation using DE, PSO, BA, FA, GSA and EBOA algorithms along with 4, 5, 6 level thresholds and respective histograms

Table 1 refers the PSNR value obtained by various algorithms. Here the EBOA and GSA algorithms have achieved higher PSNR value in comparison with others. Among all cases as well as in all images, PSNR value is rising with increasing threshold values. EBOA algorithm provides the max. value of PSNR value with Th = 5 when compared to other algorithms. Hence, we can get better segmented images with the higher level of thresholds.

Table 1. The PSNR measure of different algorithms

Image	Th	DE	PSO	BA	FA	CS	GSA	EBOA
Camera	3	26.60213	28.25586	29.56014	29.02149	30.90165	31.20465	29.73963
	4	26.61813	29.18863	31.33285	29.99707	31.45186	30.97312	31.68389
	5	26.61813	28.50686	30.61129	31.1032	31.05284	**33.55251**	32.13543
	6	28.80897	28.93363	31.9263	30.5323	29.5968	32.29559	31.96813
Goldhill	3	28.87907	28.43481	29.59048	29.1239	29.90869	28.90563	29.98469
	4	28.65093	28.51662	30.68978	30.67167	29.16283	31.05153	30.75931
	5	28.65093	28.84518	30.95788	29.58159	31.28225	29.88166	31.39651
	6	28.37174	28.74964	32.36449	29.41004	31.84383	31.98471	**32.92608**
Lena	3	27.92262	29.39812	29.78983	29.36779	30.62203	30.63608	29.59694
	4	27.77071	28.44995	30.52292	28.38537	28.4067	29.90879	30.36138
	5	27.77071	29.26986	30.81649	29.71176	30.33747	28.91859	30.88855
	6	27.88285	29.0964	31.11051	29.82705	30.72116	30.71936	31.29434
Pirate	3	27.36583	29.8279	29.1518	29.27281	29.48887	28.75609	29.30291
	4	28.04602	28.39604	29.94291	30.00692	29.76595	30.07748	30.31626
	5	28.04602	28.36981	30.04319	29.13188	30.73779	30.60739	30.95875
	6	29.1251	30.08621	31.56044	30.89285	30.95642	30.8693	31.12991
Starfish	3	27.73579	29.57534	29.08599	29.85448	28.28988	29.13537	30.41443
	4	28.75041	28.9168	30.40462	28.93692	30.34345	29.31764	30.56258
	5	28.75041	28.65474	30.70223	28.52489	29.09815	29.38433	31.0359
	6	29.11875	28.91502	32.48521	29.99572	30.75464	30.09689	32.32898

The disadvantage with PSNR is that any of the human visual system (HVS) attributes are not considered, in PSNR measurements. HVS parameters are considered in WPSNR, and it is more accurate than PSNR. The WPSNR uses the principle of redundancy of the human eye towards high-frequency components in images. The human perception of vision is less sensitive to edges than smooth areas. The value of Noise Visibility Function (NVF) is near to zero, in smooth regions and in the regions with edges and texture NVF is near to unity. Table 2 shows the WPSNR value obtained by different algorithms. From Table 2 it is observed that EBOA is better in WPSNR value compared to other algorithms.

Table 2. The WPSNR measure of different algorithms

Image	Th	DE	PSO	BA	FA	CS	GSA	EBOA
Camera	3	−30.9579	−24.72295	−21.9069	−23.9133	−21.2245	−21.0375	−21.7861
	4	−30.7821	−27.41713	−18.1073	−21.982	−19.8163	−21.2265	−18.1612
	5	−30.7821	−23.71671	−20.3112	−21.3218	−19.3738	−17.8056	−17.2241
	6	−23.523	−22.86379	−17.0843	−20.4472	−22.4571	−17.2534	−17.206
Goldhill	3	−31.2143	−27.22673	−23.1698	−22.7592	−23.1806	−25.7733	−19.9387
	4	−27.1814	−25.14317	−20.2554	−23.9193	−24.0788	−27.5566	−18.712
	5	−27.1814	−25.36869	−19.0725	−24.1362	−18.7845	−22.0732	−17.5438
	6	−25.3459	−24.6279	−16.7657	−24.585	−17.28	−17.9393	−15.7415
Lena	3	−33.005	−29.23403	−22.6573	−29.0577	−27.7202	−27.4565	−22.5387
	4	−28.7565	−27.07864	−21.015	−28.0572	−28.1433	−23.7257	−20.4705
	5	−28.7565	−25.45638	−19.5227	−23.929	−24.9051	−26.6931	−19.136
	6	−28.8747	−24.65748	−18.5302	−24.6508	−21.5971	−21.192	−17.8646
Pirate	3	−33.0063	−23.83324	−24.3604	−24.9838	−25.0992	−28.4001	−22.9615
	4	−30.1658	−29.12416	−21.1809	−22.5039	−26.4258	−25.5781	−20.2118
	5	−30.1658	−29.43271	−21.0433	−27.6521	−23.7769	−21.1982	−18.527
	6	−26.8616	−22.98106	−18.1323	−22.0565	−19.4577	−20.484	−18.2908
Starfish	3	−26.595	−21.21303	−22.6041	−22.9067	−25.3169	−23.4928	−20.5253
	4	−24.2177	−23.74275	−19.1982	−23.3254	−20.9666	−24.1755	−19.2704
	5	−24.2177	−23.32056	−20.7121	−24.3576	−24.5802	−23.1378	−18.2698
	6	−23.9592	−24.16839	−16.5682	−21.511	−20.0447	−22.4153	−16.6567

Misclassification error is a measure of uniformity in threshold image and is used to compare optimization techniques performance. Generally, misclassification errors lie between 1 & 0 and higher value of misclassification error shows better performance of the algorithm. Misclassification error measure is demonstrated in Table 3 for different algorithms under comparison and here the BA algorithm has lesser misclassification error.

Table 3. The Misclassification error measure of different algorithms

Image	Th	DE	PSO	BA	FA	CS	GSA	EBOA
Camera	3	0.54871	0.890899	0.933911	0.887988	0.880142	0.916302	0.909514
	4	0.458821	0.863031	0.890428	0.903899	0.794824	0.920022	0.879794
	5	0.348526	0.873434	0.73148	0.719453	0.780259	0.852995	0.773898
	6	0.692627	0.77251	0.777219	0.845019	0.457737	0.836168	0.740903
Goldhill	3	0.672829	0.870927	0.917136	0.921484	0.801273	0.867112	0.899479
	4	0.700017	0.839849	0.718559	0.591499	0.809197	0.662501	0.784466
	5	0.650022	0.805232	0.722134	0.300322	0.518196	0.789986	0.7089
	6	0.578608	0.718628	0.45824	0.777049	0.708081	0.671038	0.477807
Lena	3	0.566252	0.854306	0.890497	0.84143	0.703318	0.835479	0.902753
	4	0.686249	0.823072	0.809626	0.796009	0.785921	0.839878	0.843545
	5	0.632812	0.881801	0.776067	0.755647	0.742936	0.787118	0.763371
	6	0.544771	0.812042	0.718216	0.692999	0.713047	0.243027	0.671996
Pirate	3	0.510699	0.837521	0.914789	0.883704	0.881758	0.755109	0.898081
	4	0.624295	0.772276	0.83492	0.723176	0.802252	0.750077	0.839484
	5	0.555368	0.69096	0.793645	0.740421	0.173702	0.56963	0.715995
	6	0.519417	0.696708	0.642991	0.408299	0.70639	0.455449	0.667661
Starfish	3	0.768847	0.910608	0.912066	0.873859	0.842786	0.895186	0.881207
	4	0.774157	0.855462	0.872968	0.711157	0.884595	0.801281	0.832246
	5	0.742696	0.859307	0.726847	0.417832	0.722639	0.728948	0.767664
	6	0.489139	0.469298	0.676861	0.706953	0.439618	0.570052	0.642989

The visual similarity between the original image and the reconstructed image/thresholded image is evaluated by Structural Similarity Index (SSIM). SSIM ranges from -1 to +1 and SSIM value equal to +1 shows original image and reconstructed image/thresholded image is similar. The algorithm is said to be good if SSIM value is near around '+1'. Table 4 shows the SSIM of various algorithms under comparison and it shows that the EBOA algorithm SSIM is higher than other algorithms.

Table 4. The SSIM measure of different algorithms

Image	Th	DE	PSO	BA	FA	CS	GSA	EBOA
Camera	3	0.5685	0.8356	0.9747	0.9212	1.03498	1.0534	1.04336
	4	0.5934	0.8138	1.0082	0.9726	1.0367	1.0338	1.07374
	5	0.5934	0.8694	1.0185	0.9767	1.04618	1.0795	1.08923
	6	0.8403	0.8844	1.0285	0.9518	1.01144	1.049	1.09405
Goldhill	3	0.4012	0.5165	0.731	0.8315	0.90528	0.6639	0.94995
	4	0.6692	0.6364	0.9154	0.8147	0.78178	0.8901	0.99405
	5	0.6692	0.5997	0.9178	0.7261	0.98424	0.8778	1.02573
	6	0.6437	0.6482	1.0084	0.6998	1.02749	1.028	1.07184
Lena	3	0.4506	0.7192	0.8582	0.7784	0.91209	0.8965	0.90375
	4	0.574	0.6746	0.9084	0.693	0.71468	0.8751	0.95752
	5	0.574	0.774	0.936	0.8437	0.89476	0.759	0.9838
	6	0.5756	0.8063	0.9372	0.8172	0.9715	0.9839	1.02174
Pirate	3	0.3126	0.8658	0.8017	0.8318	0.88433	0.7148	0.93136
	4	0.5117	0.6088	0.9294	0.9488	0.83941	0.9042	1.01275
	5	0.5117	0.5713	0.9287	0.6998	0.95747	1.0042	1.06374
	6	0.6998	0.8678	1.0095	0.9943	1.03664	1.0318	1.06727
Starfish	3	0.2416	0.9182	0.7403	0.7747	0.64199	0.7617	1.01197
	4	0.5875	0.6141	0.9877	0.718	1.02454	0.5691	1.04741
	5	0.5875	0.8321	0.8985	0.6612	0.54001	0.8448	1.09861
	6	0.5639	0.4916	1.0663	0.9057	1.00231	0.8514	1.13065

Feature similarity index matrix (FSIM) is a method used to map the features and measure the similarity between images (i.e., the stored and the original image). It is used in quality image assessment. FSIM only considers the luminance component of the image. The results for the FSIM metric are provided in Table 5. In this table, the EBOA is better in most cases because it has a higher value.

Table 5. The FSIM measure of different algorithms

Image	Th	DE	PSO	BA	FA	CS	GSA	EBOA
Camera	3	0.682708	0.8146	0.805348	0.894196	1.015135	1.023422	1.015525
	4	0.709145	0.814295	0.861992	0.940496	1.027062	1.017921	1.065615
	5	0.709145	0.855372	0.861633	0.964682	1.031454	1.084466	1.08839
	6	0.840142	0.880837	0.89034	0.959987	1.009114	1.061976	1.095832
Goldhill	3	0.521689	0.623775	0.684309	0.876796	0.942332	0.770064	0.993697
	4	0.704053	0.711821	0.821184	0.88633	0.848817	0.92766	1.036451
	5	0.704053	0.704432	0.832433	0.828413	1.036332	0.93676	1.070311
	6	0.695975	0.730948	0.889028	0.803813	1.071304	1.045449	1.112034
Lena	3	0.621385	0.758798	0.751098	0.822457	0.927726	0.91802	0.945347
	4	0.673635	0.721451	0.796671	0.808264	0.831227	0.928681	0.996526
	5	0.673635	0.80778	0.820173	0.895486	0.933248	0.866035	1.022888
	6	0.674911	0.8154	0.84065	0.901803	0.993027	0.988662	1.057706
Pirate	3	0.475195	0.895962	0.738338	0.882487	0.955286	0.827562	0.988601
	4	0.632109	0.701426	0.832648	0.992877	0.90678	0.941063	1.05695
	5	0.632109	0.674805	0.842448	0.817585	0.99317	1.044921	1.094944
	6	0.759971	0.910414	0.899212	1.010521	1.078057	1.06189	1.106775
Starfish	3	0.339871	0.932946	0.771684	0.926768	0.755716	0.920216	1.058608
	4	0.721644	0.794676	0.902441	0.872557	0.982478	0.740747	1.116699
	5	0.721644	0.772092	0.857087	0.782738	0.727557	0.93393	1.136493
	6	0.723164	0.690497	0.935108	1.014631	1.094359	0.981415	1.156856

5. Conclusions

Metaheuristic algorithms are the best popular and efficient stochastic methods for resolving optimization issues. The Election Based Optimization Algorithm (EBOA), which is based on the voting process was proposed in this article. By comparing the presentation of six state-of-the-art metaheuristic algorithms in terms of performance metrics like PSNR, WPSNR, Misclassification error, SSIM, and FSIM, the quality of results produced by the EBOA is measured. By comparing the simulation results, it is possible to determine that EBOA outperforms the six metaheuristic algorithms in terms of optimization results and efficiency. The analysis's conclusions indicate how highly capable and efficient the EBOA is at addressing multilevel image thresholding optimization problems.

References

[1] Y. Chen, et al., "Research of improving semantic image segmentation based on a feature fusion model," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 11, pp. 5033–5045, 2022.

[2] A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Singh, "Cuckoo search algorithm and wind driven optimization-based study of satellite image segmentation for multilevel thresholding using Kapur's entropy," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3538–3560, 2014.

[3] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[4] J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vis. Graph. Image Process.*, vol. 29, no. 3, pp. 273–285, 1985.

[5] A. Marciniak, M. Kowal, P. Filipczuk, and J. Korbicz, "Swarm intelligence algorithms for multi-level image thresholding," in *Intelligent Systems in Technical and Medical Diagnostics*, Berlin, Heidelberg, Germany: Springer, 2014, pp. 301–311, https://doi.org/10.1007/978-3-642-39881-0.

[6] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Eng. Appl. Artif. Intell.*, vol. 27, pp. 57–69, 2014.

[7] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm uses dynamic feature fusion and IV-SVM classifier," *Cluster Comput.*, vol. 22, pp. 7665–7675, 2019.

[8] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[9] H. Mittal, R. Pal, A. Kulhari, and M. Saraswat, "Chaotic kbest gravitational search algorithm," in *Proc. 9th Int. Conf. Contemp. Comput. (IC3)*, Noida, India, Aug. 11–13, 2016, pp. 1–6, https://doi.org/10.1109/IC3.2016.7880252.

[10] M. Seyedali and L. Andrew, "The Whale Optimization Algorithm," *Adv. Eng. Software*, vol. 95, pp. 51–67, 2016.

[11] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, 2015, https://doi.org/10.1016/j.knosys.2015.07.006.

[12] P. K. Sahoo and G. Arora, "A thresholding method based on two-dimensional Renyi's entropy," *Pattern Recognit.*, vol. 37, no. 6, pp. 1149–1161, 2004, https://doi.org/10.1016/j.patcog.2003.10.008.

[13] C. Karri, G. R. Babu, P. M. K. Prasad, and M. S. R. Naidu, "Novel 2-D histogram-based soft thresholding for brain tumor detection and image compression," *Int. J. Appl. Metaheuristic Comput. (IJAMC)*, vol. 13, no. 1, pp. 1–21, 2022.

[14] F. Hamdaoui, A. Sakly, and A. Mtibaa, "An efficient multi-level thresholding method for image segmentation based on the hybridization of modified PSO and Otsu's method," *Comput. Intell. Appl. Model. Control*, vol. 575, pp. 343–367, 2015.

[15] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 392, p. 114616, 2022, https://doi.org/10.1016/j.cma.2022.

[16] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "QANA: Quantum-based avian navigation optimizer algorithm," *Eng. Appl. Artif. Intell.*, vol. 104, p. 104314, 2021.

[17] K. Chiranjeevi, U. Jena, and M. N. Rao, "Image compression based on adaptive image thresholding by maximizing Shannon or fuzzy entropy using teaching learning-based optimization," *Int. J. Adv. Intell. Paradigms*, vol. 18, no. 2, pp. 193–231, 2021.

[18] M. S. R. Naidu, P. R. Kumar, and K. Chiranjeevi, “Shannon and fuzzy entropy based evolutionary image thresholding for image segmentation,” *Alexandria Eng. J.*, vol. 57, no. 3, pp. 1643–1655, 2018.

[19] F. S. Gharehchopogh and T. Ibrikci, “An improved African vulture’s optimization algorithm using different fitness functions for multi-level thresholding image segmentation,” *Multimedia Tools Appl.*, vol. 83, no. 6, pp. 16929–16975, 2024.

[20] E. H. Houssein, N. Abdalkarim, K. Hussain, and E. Mohamed, “Accurate multilevel thresholding image segmentation via oppositional Snake Optimization algorithm: Real cases with liver disease,” *Comput. Biol. Med.*, vol. 169, p. 107922, Feb. 2024, https://doi.org/10.1016/j.compbiomed.2024.10792.

[21] L. Qiao, K. Liu, Y. Xue, W. Tang, and T. Salehnia, “A multi-level thresholding image segmentation method using hybrid Arithmetic Optimization and Harris Hawks Optimizer algorithms,” *Expert Syst. Appl.*, vol. 241, p. 122316, 2024, https://doi.org/10.1038/s41598-024-58456-2.

[22] J. Zhang, G. Zhang, M. Kong, and T. Zhang, “SCGJO: A hybrid golden jackal optimization with a sine cosine algorithm for tackling multilevel thresholding image segmentation,” *Multimedia Tools Appl.*, vol. 83, no. 3, pp. 7681–7719, 2024, https://doi.org/10.1007/s11042-023-15812.

[23] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu, “A new iterative triclass thresholding technique in image segmentation,” *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1038–1046, 2014.

[24] J. H. Xue and D. M. Titterington, “T-Test, F-tests and Otsu’s methods for image thresholding,” *IEEE Trans. Image Process.*, vol. 20, pp. 2392–2396, 2011, https://doi.org/10.1109/TIP.2011.2114358.

[25] P. Ranefall and C. Wählby, “Global gray-level thresholding based on object size,” *Cytometry Part A*, vol. 89A, pp. 385–390, Sweden, 2016, https://doi.org/10.1002/cyto.a.22806.

[26] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu, “A new iterative triclass thresholding technique in image segmentation,” *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1038–1046, 2014.

[27] Y. Zou, F. Dong, B. Lei, S. Sun, T. Jiang, and P. Chen, “Maximum similarity thresholding,” *Digit. Signal Process.*, vol. 28, pp. 120–135, 2014.

[28] H. Gao, S. Kwong, J. Yang, and J. Cao, “Particle swarm optimization based on an intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation,” *Inf. Sci.*, vol. 250, pp. 82–112, 2013.

[29] H. Gao, W. Xu, J. Sun, and Y. Tang, “Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 4, pp. 934–946, 2009.

[30] C. Karri and U. Jena, “Fast vector quantization using a Bat algorithm for image compression,” *Eng. Sci. Technol. Int. J.*, vol. 19, no. 2, pp. 769–781, 2016.