Indonesian Journal of Electrical Engineering and Computer Science
Vol. 31, No. 2, August 2023, pp. 933~944
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v31.i2.pp933-944

An Otsu thresholding for images based on a nature-inspired optimization algorithm

Khamael Raqim Raheem, Hafedh Ali Shabat
Technical Institute of Babylon, AL-Furat AL-Awsat Technical University (ATU), Kufa, Iraq

Article Info

Article history:

Received Oct 18, 2022
Revised Mar 12, 2023
Accepted Mar 24, 2023

Keywords:

Bat algorithm
Hill climbing algorithm
Image thresholding
Optimization
Otsu thresholding

ABSTRACT

Thresholding is a type of image segmentation, where the pixels change to make the image easier to analyze. In bi-level thresholding, the image in grayscale format is transformed into a binary format. The traditional methods for image thresholding may be inefficient in finding the best threshold and take longer computation time. Recently, metaheuristic swarm-based algorithms were applied for optimization in different applications to find optimal solutions with minimum computational time. The proposed work aims to optimize the fitness function obtained by the Otsu algorithm using a metaheuristic swarm-based algorithm called the bat algorithm. As a result, the optimal threshold value for bi-level images in cloud detection was obtained. Also, one of the trajectory-based algorithms called hill climbing was applied to optimize the fitness function taken from the Otsu algorithm. The HYTA dataset was used to evaluate the work, which was later confirmed through testing. The findings of experiments indicated that the developed algorithm is promising and the performance of the metaheuristic population-based algorithm is better than the trajectory-based algorithm in terms of efficiency and computational time for image thresholding.

Corresponding Author:

Khamael Raqim Raheem
Technical Institute of Babylon, AL-Furat AL-Awsat Technical University (ATU)
Kufa, Iraq
Email: khmrakrah@atu.edu.iq

1. INTRODUCTION

Thresholding is a simple but efficient technique in terms of segmenting images. Bi-level thresholding and multi-level thresholding are the two types of thresholding that are used in practice. Over the past ten years, bi-level thresholding has attracted a lot of attention from researchers [1], [2]. The study of multilevel thresholding has also been ongoing [3]-[5]. The image in grayscale format is transformed into an image in binary format using the technique of bi-level thresholding [6]. The best thresholding image can be obtained by choosing an appropriate threshold value while separating the foreground from the background. The basic thresholding algorithms are global thresholding, local thresholding, and hybrid thresholding. The methods used by global thresholding algorithms are based on classification techniques, histograms, clustering, entropy, and Gaussian distribution [7]. Thresholding is widely used in medical image segmentation [8], [9], engineering [10], stone inscription [11], and document image [12]. The Otsu algorithm is a global thresholding algorithm, which falls within the field of classification techniques.

Otsu's approach was improved by Sha *et al*. [13] by creating a two-dimensional histogram based on an image that had been processed with median and average filters. This method also adds a region post-processing phase to cope with noise and edge-filled pixel issues. Also, Indra *et al*. [14] used the Otsu thresholding algorithm to separate objects and backgrounds to distinguish fertile and infertile domestic fowl eggs for the grayscale image. Additionally, a strategy in [15] was offered as a method for detecting green plants in a maize crop. This technique is based on segmentation and dimensionality decrease using principal component analysis, Otsu thresholding, threshold combination, and final thresholding. Meanwhile, brain image segmentation using Otsu thresholding is proposed by Badriyah *et al*. [16] to determine the characteristics of a specific stroke kind. They evaluated the results using the peak signal-to-noise ratio and mean-square error. Additionally, [17] used Otsu thresholding to isolate the roads and residential areas from the vegetated areas in remote sensing images. They used accuracy and precision for testing the results. A granule size selection method based on the homogeneity histogram is also suggested by Lei and Fan [18] for bi-level thresholding for images, which is useful in handling small objects and local changes in the images. In addition, by investigating the link between pixel grayscale value and cumulative pixel number change, the study in Yang *et al*. [19] improved the strategy for Otsu thresholding to adjust the threshold bias and as the adjusted threshold. The ratio of pixel gray level value to a particular cumulative pixel number was chosen. For quantitative evaluation, two regularly used measures were chosen: misclassification error and dice similarity coefficient. Also, for gesture image segmentation, the work in [20] dealt with Otsu thresholding where a noise-adaptable angle threshold was devised. To prevent interference from extreme noise by removing neighborhood extremes, a two-dimensional histogram of gray value-neighborhood trimmed gray mean is initially constructed. After then, adaptive filtering is used to increase the algorithm's overall applicability by calculating the likelihood that each pixel is noise based on the current circumstances. To increase efficiency, the threshold search range is compressed and the threshold space is finally transformed into an angled space from 0° to 90°.

Over the past forty years, the popularity of the nature-inspired and bio-inspired metaheuristic optimization era has grown significantly [21]. The effectiveness of the metaheuristic population-based algorithms is evaluated by looking at how well exploration and exploitation are balanced. The likelihood of the algorithm becoming trapped in local optima, early convergence, and stagnation is higher when there is a poor balance between exploration and exploitation [22]. It's interesting to note that the number of suggested nature-inspired optimization algorithms has increased exponentially. These algorithms deal with many data types, including text data [23] and image data [24]. Recently researchers have proposed a variety of techniques to speed up image thresholding, including swarm intelligence optimization algorithms such as the particle swarm algorithm [25], ant colony algorithm [26], and fruit fly optimization algorithm [27]. which have achieved less computational cost. Mostly, these algorithms have been successful to minimize time complexity and improve image quality to some extent, but there is further scope for improvement using other nature-inspired optimization algorithms such as the bat algorithm. Bat algorithm is a global optimization intelligent algorithm based on biological heuristics. To determine the ideal value, it mimics how the bat population searches for food using echolocation. Practice demonstrates that Bat algorithm is a successful search algorithm with quick convergence and great global optimization search capability. The algorithm has the characteristics of high robustness and identification accuracy while being simple in concept and less constrained by parameters.

The aim of the proposed work is to optimize the fitness function obtained from the Otsu algorithm using the Bat algorithm for cloud detection and then compare its performance with a trajectory-based local search algorithm called the hill climbing algorithm. In ground-based sky imager applications, cloud detection is a prerequisite before other information (such as cloud cover) may be derived. In the literature, limited works applied for bi-level image thresholding using metaheuristic algorithms. The contribution of this paper is to demonstrate the feasibility of the Bat algorithm and Otsu algorithm for bi-level thresholding. Also, it offers a new option to conventional methods due to its simplicity and efficiency with minimum computational time.

The remainder sections are constructed according as: section 2 explains the concepts of the hill climbing algorithm, bat algorithm, and Otsu algorithm. Section 3 introduces the proposed thresholding algorithm. The experimental results and discussion are illustrated in section 4. Finally, section 5 explains the conclusion and future work.

2. THEORETICAL BASIS

2.1. Hill climbing algorithm

A relatively straightforward local search technique called "Hill climbing" aims to enhance a single candidate solution from a randomly chosen starting point. Figure 1 illustrates the pseudo-code for the hill climbing algorithm where the adjacent search space is assessed in relation to the current location. The search shifts to a more suitable candidate solution if one is discovered. The algorithm ends if there isn't a better solution in the vicinity. Since the procedure is comparable to climbing hills on the surface of the fitness function, the technique has been referred to as "Hill climbing".

```
i = initial solution
While f(s) ≤ f (i) s Є Neighbours (i) do
    Generates an s Є Neighbours (i);
    If fitness (s) > fitness (i) then
        Replace s with the i;
    End If
```

Figure 1. Hill climbing algorithm [28]

2.2. Bat algorithm

The Bat algorithm is a nature-inspired metaheuristic based on a population that was put forth by Yang [29]. The algorithm utilizes the echolocation of bats, It is well knowledge that sound pulses change into a frequency that obstacles reflect. Bats navigate utilizing the delay of time between emission and reflection. After hitting and reversal, bats use their pulse as beneficial data to assess how far away their target is. Yang implemented the Bat algorithm by the three following broad guidelines: i) Echolocation is used by all bats to locate distance, and in some mysterious way, they can also discern between background barriers and prey or food. ii) Bats fly at random with a velocity of v_i, a position of x_i, a fixed frequency of f_{min}, a variable wavelength λ, and loudness of A_0 to find prey. They can automatically vary the wavelength of their produced pulses as well as the rate of pulse emission rϵ[0,1] relying on how close their target is. iii) Though loudness can alter in a variety of ways, it normally ranges from a big (positive) A_0 to a little constant value A_{min}.

Figure 2 illustrates the flowchart of Bat algorithm steps, at the first the bat population is randomly initialized. Namely, generating new solutions is performed by moving virtual bats by:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{1}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x*)f_i \tag{2}$$

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

where βϵ[0,1] is a random vector taken from a uniform distribution. Here X* represents the existing best global location (solution). This can be determined by comparing all of the bats' solutions. Each bat is initially assigned a frequency chosen uniformly from [f_{min}, f_{max}] at random. In the local search that alters the existing optimal solution, a random walk with direct exploitation is used as in (4).

$$x_{new} = x_{old} + \partial A^t \tag{4}$$

Where ∂ϵ[-1,1] is a random number, whilst A^t is the average loudness of all the best at this time step. By the rate r_i of pulse emission, the local search is started. The loudness can be set to any convenient value because, after a bat finds its prey, the loudness often drops while the rate of pulse emission rises. As a result, both characteristics mimic those of natural bats. Mathematically, these characteristics are captured as:

$$A_i^{t+1} = \alpha A_i^t, r_i^{t+1} = r_i^0[1 - \exp(-\gamma t) \tag{5}$$

where α and $\boldsymbol{\gamma}$ are constants. In reality, the parameter α has a role similar to the cooling factor of a cooling schedule in the simulated annealing.

2.3. Otsu algorithm

Otsu algorithm was proposed by Otsu [30] in 1979 for image thresholding where a criterion function generates some sort of measure of dissociation between regions of the image. Otsu thresholding selects the threshold value that minimizes intra-class variation of the foreground and background pixels. According to Bangree *et al*. [31] lists the steps of the Otsu thresholding algorithm as shown in:

- Step 1: Calculate a histogram for a two-dimensional image.
- Step 2: For a single threshold, determine the foreground and background variances (a measure of spread).
  a. Do the background and foreground pixels' weight calculations.
  b. Determine the mean value for both the background and foreground pixels.
- Step 3: Compute "within class variance".

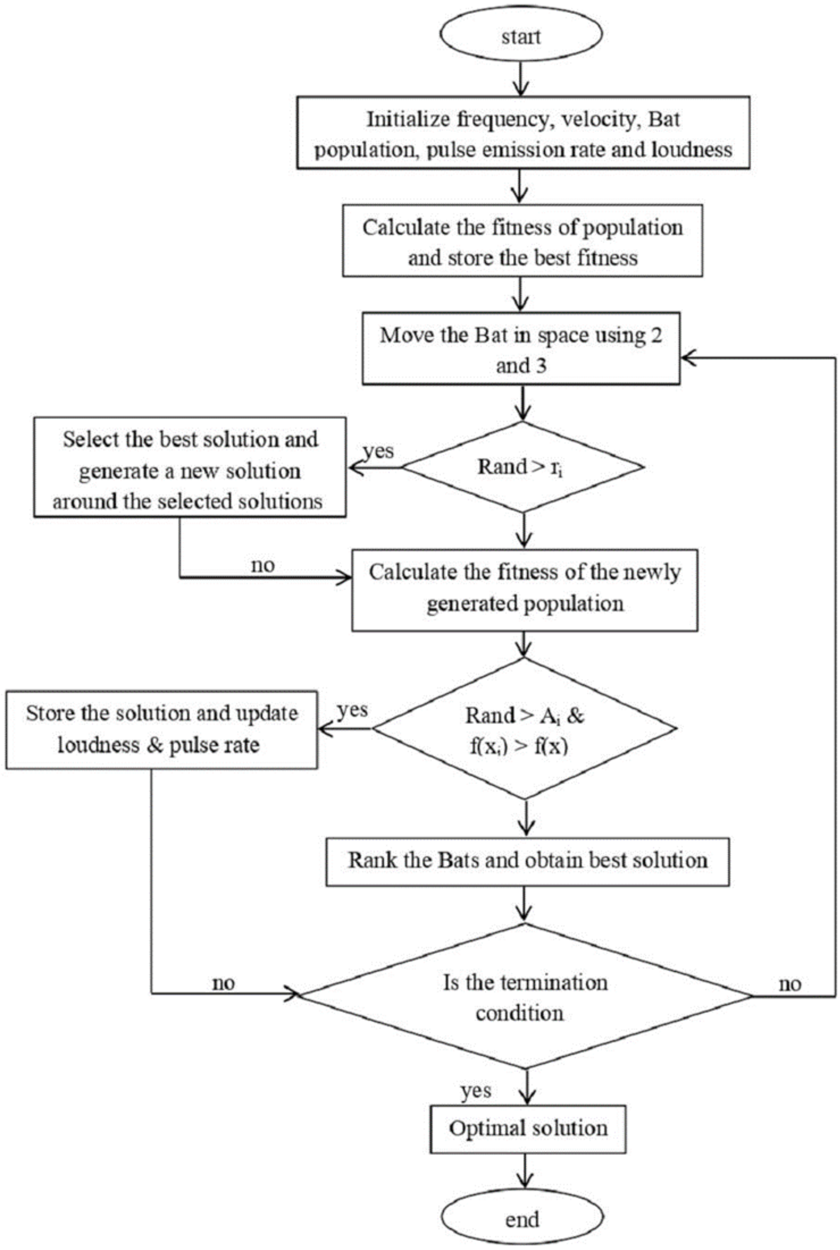


Figure 2. The flowchart of Bat algorithm [32]

The range of grayscale values for the input image is i=0,1,…….., L-1, and the pixel numbers with the grayscale k are n_k then the overall number of pixels N in an image is computed as:

$$N = \sum_{k=0}^{L-1} n_k = n_0 + n_1 + \cdots + n_{L-1} \tag{6}$$

the probability of gray level k's occurrence is:

$$p_k = \frac{n_k}{N} = \frac{n_k}{\sum_{k=0}^{L-1} n_k} \tag{7}$$

the gray level threshold t can separate the gray level of an image into class 1: (0,1,…….t), and class 2: (t+1,t+2,….., L-1). The estimated class probabilities are as:

$$q_1(t) = \sum_{i=0}^{t} P(i) \tag{8}$$

$$q_2(t) = \sum_{i=t+1}^{L-1} P(i) \quad (9)$$

and the means for the class are provided by:

$$u_1(t) = \sum_{i=0}^{t} \frac{iP(i)}{q_1(t)} \quad (10)$$

$$u_2(t) = \sum_{i=t+1}^{L-1} \frac{iP(i)}{q_2(t)} \quad (11)$$

the following are the individual class variances:

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - u_1(t)]^2 \frac{P(i)}{q_1(t)} \quad (12)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{L-1} [i - u_2(t)]^2 \frac{P(i)}{q_2(t)} \quad (13)$$

the goal is to pick the value that decreases the weighted within-class variance in (14).

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (14)$$

3. METHOD

Figure 3 shows the block diagram of the proposed algorithm. Initially, the entered color image will be converted to grayscale image. Then two optimization algorithms were applied, one of which is swarm-based, which is the Bat algorithm, and the second is trajectory-based, which is the hill climbing algorithm. Both algorithms will optimize the fitness function taken from the Otsu algorithm. The images resulted from both algorithms will be evaluated in terms of F-score and computation time.

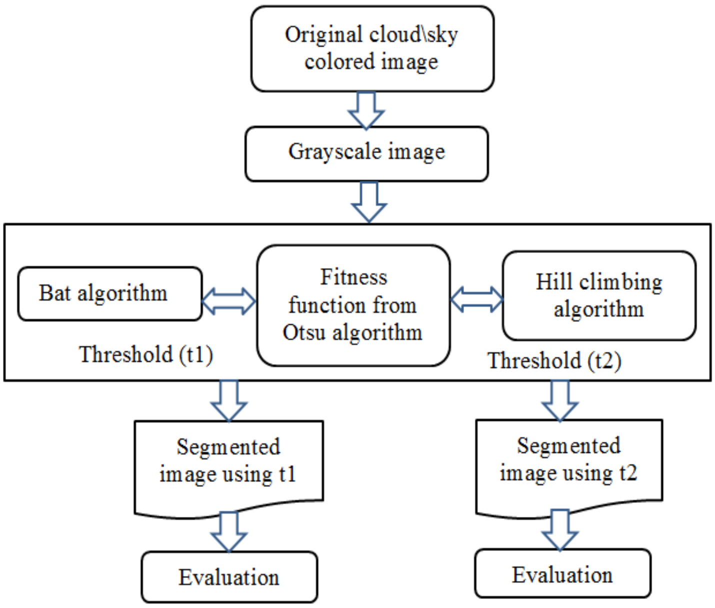


Figure 3. Block diagram of the proposed algorithm

Figure 4 shows the steps of the proposed algorithm using Bat algorithm to optimize fitness function obtaining from Otsu algorithm.

- Step 1: The Bat algorithm produces in random an initial population of N solutions (bats) with one dimension indicated by X where X=[x_1, x_2, ….., x_N], x_i is restricted into [0,…, L-1], and L represents the maximum gray level of the input image, which are in the range [0,..., L-1]. All solutions' fitness values are examined, and cycle=1 is set. The Bat algorithm identifies the best effective solution as the best solution before beginning the iterative search procedure.

- Step 2: Virtual bats are moved by (1), (2), and (3) to generate a new solution. Namely, by adjusting the velocity matrix V and the frequency vector f using (1) and (2) respectively, new solutions are produced. Following that, (3) was used in an update procedure for each solution in the search population X.
- Step 3: For the local search that updates the existing best solution vector by (4), a random walk with direct exploitation is utilized. Namely, the local search is initiated with the proximity relying on pulse rate r.
- Step 4: The fitness function value (Otsu criterion) for the current best solution vector acquired in step 3 will be computed in this step using (14). Accept the new solution vector and change the old fitness value if the fitness value of the solution vector is more than the old fitness value and the loudness A produced by (5) is not loud. If not, preserve the best solution from before.
- Step 5: Keep the best solution vector with the best optimal solution. The cycle is increased by one.
- Step 6: Finish the algorithm if the cycle equals the maximum iterations allowed; otherwise, go to step 2.

Step1: Producing the initial population of solutions randomly: $X=[x_1, x_2, \ldots.., x_N]$ where x_i is restricted into [0,…, L-1], and L maximum gray level of input image.
Step2: Generation of new solutions using (1), (2), and (3).
Step3: Applying local searching using (4).
Step4: Generation of a new solution and computes fitness function based on Otsu criteria in (14).
Step5: Keep track of the best solution.
Step6: Examine the termination criterion. If the cycle not equals the maximum iterations go to Step 2, otherwise finish the algorithm.

Figure 4. Proposed algorithm using Bat algorithm and Otsu fitness function

Figure 5 shows the steps of the proposed algorithm using hill climbing to optimize the fitness function obtaining from Otsu algorithm.

- Step1: The hill climbing algorithm produces an initial solution randomly indicated by x where, x is restricted into [0,…, L-1], and L represents the maximum gray level of the input image, which are in the range [0,..., L-1].
- Step2: Generating a new solution based on the neighbors of the current solution.
- Step3: The fitness function obtained from Otsu algorithm for the current best solution acquired in step will be computed in this step using (14).
- Step4: Accept the new solution and change the old fitness value if the fitness value of the new solution is better than the old fitness then go to 2, otherwise terminate the algorithm.

Step1: Producing the initial solution randomly:
x where x is restricted into [0,…, L-1], and L maximum gray level of input image.
Step2: Generation new solution s from neighbors of x.
Step3: Compute Otsu fitness function using (14).
Step4: if fitness (s) is better than fitness(x) then s is the best solution and go to 2, otherwise the algorithm finish.

Figure 5. Proposed algorithm using hill climbing algorithm and Otsu fitness function

4. RESULTS AND DISCUSSION

The proposed algorithm was performed in python. Tests were applied on a PC with core (TM) i5 and 12 GB of RAM on the Windows 10 operating system. HYTA dataset is a repository that contains images and ground-truth images that were used for the evaluation. The dataset prepared by the researchers in [33], stated that the statistical properties of the cloud\sky images can be roughly split into two groups: unimodal and bimodal. Typically, unimodal images consist of a single element (i.e., cloud or sky), while bimodal images are made up of both sky and cloud components. In contrast to a bimodal image, which may have two or more peaks and a big variance, the histogram of the unimodal image always has a single peak and a modest variance.

In the preprocessing, the input color image was converted into a grayscale image before applying the proposed optimization algorithm. Figure 6 illustrates bimodal images from the HYTA dataset with their grayscale images and histograms. 15 out of 32 images are selected from the HYTA dataset for evaluation. Several unimodal images were excluded such as clear sky images as in Figure 7.

Figure 6. Grayscale images and histograms for bimodal images from HYTA dataset

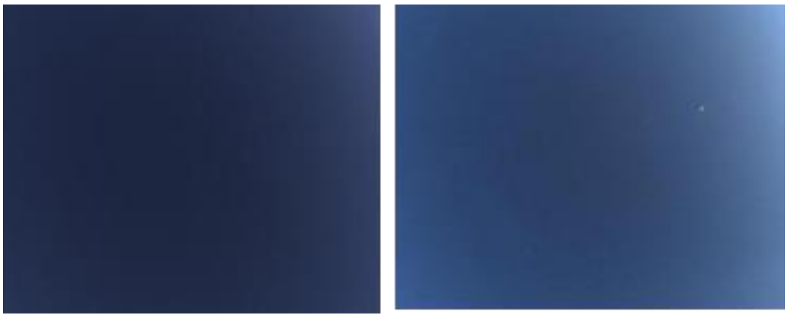


Figure 7. Unimodal images in the HYTA dataset

Two experiments were conducted in the proposed work, the first experiment applied the Bat algorithm to optimize the fitness function obtained from the Otsu algorithm. Based on practice work the best control parameters obtained are: the size of population (N) is 10, number of iterations is 15, loudness (A) is 0.5, pulse rate (r) is 0.5, minimum frequency (f_{min}) is 0, and maximum frequency (f_{max}) is 2. The second experiment uses the hill climbing algorithm to optimize the fitness function obtained from the Otsu algorithm. Based on practice work the best-selected parameters are: the number of iterations is 1000, and the step size is 0.7. Figure 8 shows all the stages of the proposed algorithm in images, where the color image was converted to grayscale. The bat algorithm was applied using the Otsu fitness function to get the first resulting image, and the hill climbing algorithm was also applied using the Otsu fitness function to get the second resulting image. The resulting images were evaluated by calculating the precision, recall, and f-score based on the ground truth image.

A comparison was made between the results of the two experiments, depending on the fitness function value, as well as the computation time. Table 1 illustrates that the computational time of the hill climbing is very slow compared to the Bat algorithm. The time needed by hill climbing for the first image is 3,407 seconds, while the time consumed by the second algorithm is 0.693 seconds. Also, regarding the fitness function, the Bat algorithm achieved better or close results. The fitness function for the second image is 556,084 using the hill climbing and 549,719 using the Bat algorithm, where the lower value of the function indicates an optimal threshold.

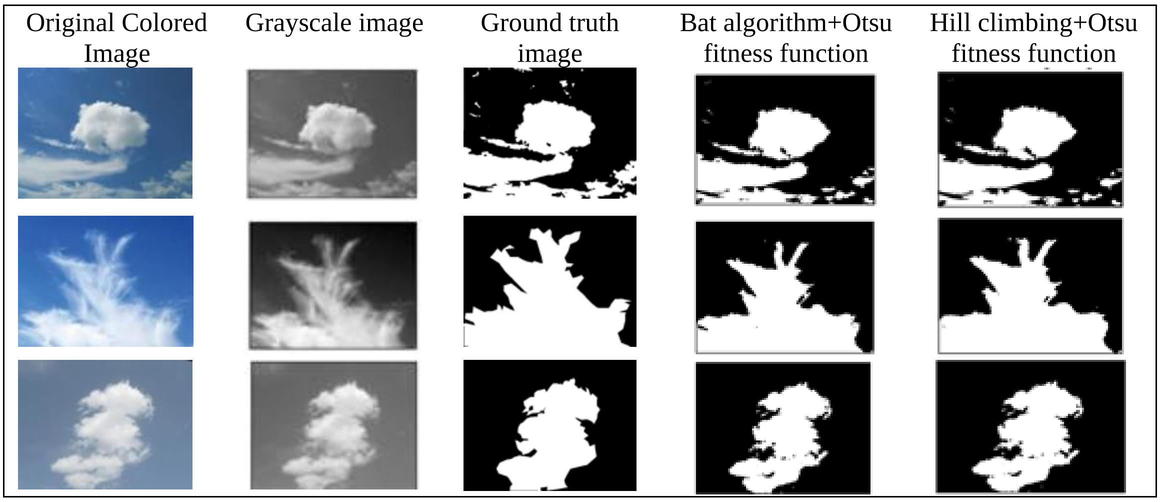


Figure 8. Resulted images for the proposed optimization algorithm

Table 1. A comparison between hill climbing and bat algorithm results

Input image	Hill climbing using Otsu algorithm			Bat algorithm using Otsu algorithm		
	Threshold	Fitness value	Time	Threshold	Fitness value	Time
	91.77	575.296	3.407	90.749	575.554	0.693
	136.504	556.084	2.65	137.284	549.719	0.735
	136.504	476.8	7.168	139.833	475.7	1.182

Precision (P), recall (R), and F-score in (15), (16), and (17) were calculated to assess the proposed work performance. As a first step, we use the output image and the ground truth image to calculate the confusion matrix. A confusion matrix is utilized to assess how well a classification model works on a set of test data whose true values are known. Table 2 illustrates the confusion matrix for binary classification. Each pixel in the resulting image can belong to the foreground represented by 1 (white) or belong to the background represented by 0 (black). True positives (TP) are pixels that were accurately predicted as positive values, implying that the real class value is white and the predicted class value is also white. True negatives (TN) are pixels that successfully predicted negative values, implying that the real class value is black and the predicted class value is similarly black. False positives (FP) occur when a pixel's real class is black while its predicted class is white. False negatives (FN) are pixels whose actual class is white but whose predicted class is black.

$$P \frac{TP}{TP+FP} \tag{15}$$

$$R \frac{TP}{TP+FN} \tag{16}$$

$$F - Score = \frac{2*(Recall*Precision)}{(Recall+Precision)} \tag{17}$$

Table 2. Confusion matrix for binary classification

	Predicted class in output image		
Actual class in ground truth image		White (Foreground)	Black (Background)
	White	TP	FN
	Black	FP	TN

Table 3 illustrates a comparison between the evaluation measures using the Bat algorithm and hill climbing. We note that the P value for the both algorithms is close, but regarding the R, there is a clear difference between them. Since the F-score represents the harmonic mean of a system's P and R values, the Bat algorithm achieved the best performance.

Table 3. The performance measures for the proposed algorithm

Dataset	Algorithm	P	R	F-Score
HYTA	Hill climbing with Otsu algorithm	0.874533	0.676333	0.732733
	Bat algorithm with Otsu algorithm	0.882944	0.757644	0.800833

The hill climbing algorithm is a trajectory-based and local search algorithm, it is more probable to trap in local optima, premature convergence, and stagnation. This algorithm begins with one solution and tried to find the best based on the neighbors therefore more computational time is consumed. Figure 9 shows the change in the value of the fitness function during the cycles of the hill climbing algorithm until the best solution is reached, which represents the best threshold value to represent the resulting image. The best solution is indicated using the red dotted line.

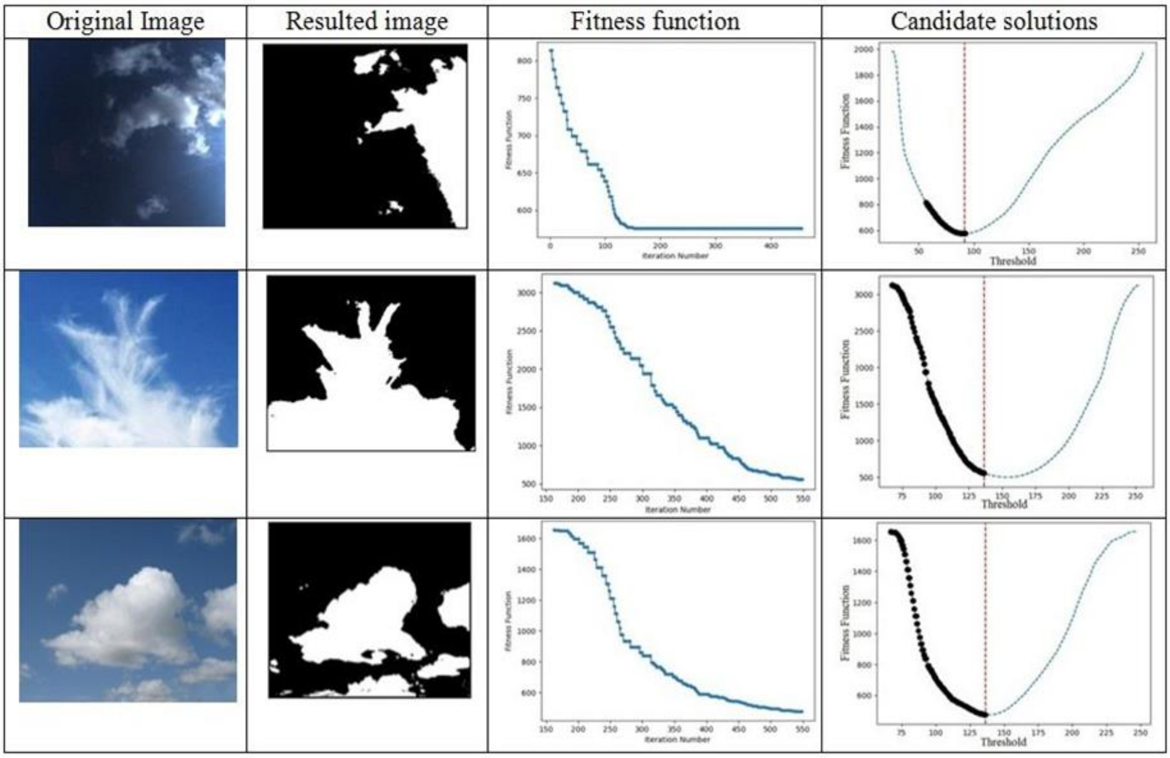


Figure 9. Fitness function and candidate solution using hill climbing algorithm

By utilizing fewer control parameters than other population-based algorithms, Bat algorithm is straightforward, adaptable, and superior. In contrast to previous computational intelligence algorithms, our technique does not require adjusting extraneous parameters like mutation, cross-over rate, etc. The simulation experiment clearly demonstrates that the Bat algorithm's addition increases the effectiveness of image thresholding while essentially maintaining its original accuracy. Also, the computational time is very low compared with the non-population algorithms such as the hill climbing algorithm. Figure 10 shows the best solution obtained using the Bat algorithm, the best solution represents the best threshold value to represent the resulting image, also this Figure shows the set of solutions found by the Bat algorithm, the optimal solution was indicated by the red dotted line.

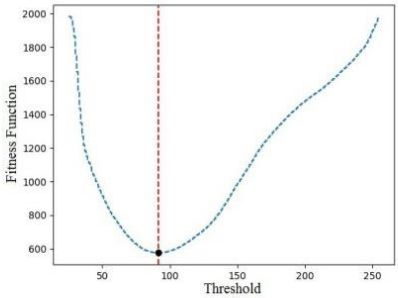
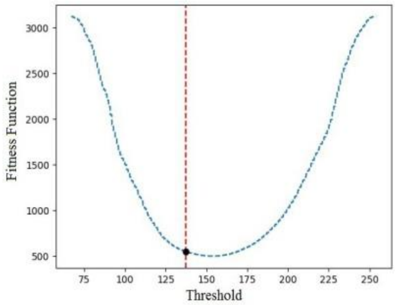
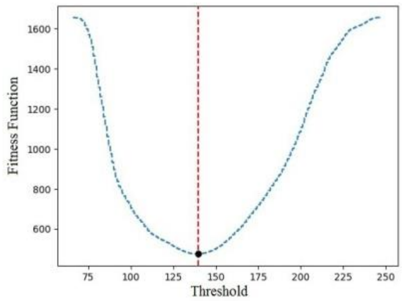
Original Image	Resulted image	Threshold	Best solution
		90.749	
		137.284	
		139.833	

Figure 10. Best threshold value using Bat algorithm with Otsu fitness function

5. CONCLUSION

The proposed work has proven that the Bat algorithm can optimize the fitness function obtained from the Otsu algorithm to quickly select the optimal threshold for image in cloud detection. The best threshold value was obtained based on the Otsu criterion that represents the intra-class variance. trajectory-based algorithms such as hill climbing is nonefficient because more time is needed for computation. On the other hand, metaheuristic, swarm-based algorithms are more effective in relation to accuracy, and computational time. Promising results have been obtained, as it provides an opportunity in the future for improving the image thresholding algorithms by making a combination between the Bat algorithm and other nature-inspired algorithms such as particle swarm optimization and ant colony.

REFERENCES

[1] A. J. Christy and A. Umamakeswari, "A novel percentage split distribution method for image thresholding," *Optik*, vol. 218, p. 164953, Sep. 2020, doi: 10.1016/j.ijleo.2020.164953.

[2] K. Katsuragawa, A. Kamal, Q. F. Liu, M. Negulescu, and E. Lank, "Bi-level thresholding: Analyzing the effect of repeated errors in gesture input," *ACM Transactions on Interactive Intelligent Systems*, vol. 9, no. 2–3, pp. 1–30, Sep. 2019, doi: 10.1145/3181672.

[3] D. Oliva, M. A. Elaziz, and S. Hinojosa, "Multilevel thresholding for image segmentation based on metaheuristic algorithms," in *Studies in Computational Intelligence*, vol. 825, 2019, pp. 59–69, doi: 10.1007/978-3-030-12931-6_6.

[4] A. K. M. Khairuzzaman and S. Chaudhury, "Masi entropy based multilevel thresholding for image segmentation," *Multimedia Tools and Applications*, vol. 78, no. 23, pp. 33573–33591, Dec. 2019, doi: 10.1007/s11042-019-08117-8.

[5] A. M. Ashir and J. R. C. Piqueira, "Multilevel thresholding for image segmentation using mean gradient," *Journal of Electrical and Computer Engineering*, vol. 2022, pp. 1–9, Feb. 2022, doi: 10.1155/2022/1254852.

[6] D. Balcan, B. Gonçalves, H. Hu, J. J. Ramasco, V. Colizza, and A. Vespignani, "Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model," *Journal of Computational Science*, vol. 1, no. 3, pp. 132–145, Aug. 2010, doi: 10.1016/j.jocs.2010.07.002.

[7] V. Sokratis, E. Kavallieratou, R. Paredes, and K. Sotiropoulos, "A hybrid binarization technique for document images," in *Studies in Computational Intelligence*, vol. 375, 2011, pp. 165–179, doi: 10.1007/978-3-642-22913-8_8.

[8] X. Li, L. Yu, H. Chen, C. W. Fu, L. Xing, and P. A. Heng, "Transformation-consistent self-ensembling model for semisupervised medical image segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 523–534, Feb. 2021, doi: 10.1109/TNNLS.2020.2995319.

[9] N. J. Vickers, "Animal communication: When I'm calling you, will you answer too?," *Current Biology*, vol. 27, no. 14, pp. R713–R715, Jul. 2017, doi: 10.1016/j.cub.2017.05.064.

[10] N. D. Hoang and Q. L. Nguyen, "A novel method for asphalt pavement crack classification based on image processing and machine learning," *Engineering with Computers*, vol. 35, no. 2, pp. 487–498, Apr. 2019, doi: 10.1007/s00366-018-0611-9.

[11] Sukanthi, S. S. Murugan, and S. Hanis, "Binarization of stone inscription images by modified Bi-level entropy thresholding," *Fluctuation and Noise Letters*, vol. 20, no. 6, Dec. 2021, doi: 10.1142/S0219477521500541.

[12] S. Susan and K. M. R. Devi, "Text area segmentation from document images by novel adaptive thresholding and template matching using texture cues," *Pattern Analysis and Applications*, vol. 23, no. 2, pp. 869–881, May 2020, doi: 10.1007/s10044-019-00811-5.

[13] C. Sha, J. Hou, and H. Cui, "A robust 2D Otsu's thresholding method in image segmentation," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 339–351, Nov. 2016, doi: 10.1016/j.jvcir.2016.10.013.

[14] D. Indra, T. Hasanuddin, R. Satra, and N. R. Wibowo, "Eggs detection using otsu thresholding method," in *Proceedings - 2nd East Indonesia Conference on Computer and Information Technology: Internet of Things for Industry, EIConCIT 2018*, Nov. 2018, pp. 10–13, doi: 10.1109/EIConCIT.2018.8878517.

[15] M. Montalvo, M. Guijarro, and Á. Ribeiro, "A novel threshold to identify plant textures in agricultural images by Otsu and principal component analysis," *Journal of Intelligent and Fuzzy Systems*, vol. 34, no. 6, pp. 4103–4111, Jun. 2018, doi: 10.3233/JIFS-171524.

[16] T. Badriyah, N. Sakinah, I. Syarif, and D. R. Syarif, "Segmentation stroke objects based on CT scan image using thresholding method," in *Proceedings - 2019 1st International Conference on Smart Technology and Urban Development, STUD 2019*, Dec. 2019, pp. 61–65, doi: 10.1109/STUD49732.2019.9018825.

[17] C. Srinivas, M. Prasad, and M. Sirisha, "Remote sensing image segmentation using Otsu algorithm," *International Journal of Computer Applications*, vol. 178, no. 12, pp. 46–50, May 2019, doi: 10.5120/ijca2019918885.

[18] B. Lei and J. Fan, "Image thresholding segmentation method based on minimum square rough entropy," *Applied Soft Computing Journal*, vol. 84, p. 105687, Nov. 2019, doi: 10.1016/j.asoc.2019.105687.

[19] P. Yang, W. Song, X. Zhao, R. Zheng, and L. Qingge, "An improved Otsu threshold segmentation algorithm," *International Journal of Computational Science and Engineering*, vol. 22, no. 1, pp. 146–153, 2020, doi: 10.1504/IJCSE.2020.107266.

[20] L. Xiao, H. Ouyang, C. Fan, T. Umer, R. C. Poonia, and S. Wan, "Gesture image segmentation with Otsu's method based on noise adaptive angle threshold," *Multimedia Tools and Applications*, vol. 79, no. 47–48, pp. 35619–35640, Dec. 2020, doi: 10.1007/s11042-019-08544-7.

[21] J. O. Agushaka and A. E. Ezugwu, "Initialisation approaches for population-based metaheuristic algorithms: a comprehensive review," *Applied Sciences (Switzerland)*, vol. 12, no. 2, p. 896, Jan. 2022, doi: 10.3390/app12020896.

[22] M. M. Shehab, M. A. Al-Betar, and A. T. Khader, "New selection schemes for particle swarm optimization," in *The 7th International Conference on Information Technology*, May 2015, pp. 17–25, doi: 10.15849/icit.2015.0003.

[23] H. A. Shabat and N. A. Abbas, "Enhance the performance of independent component analysis for text classification by using particle swarm optimization," in *3rd International Conference on Advanced Science and Engineering, ICOASE 2020*, Dec. 2020, pp. 1–6, doi: 10.1109/ICOASE51841.2020.9436547.

[24] M. T. Cao, K. T. Chang, N. M. Nguyen, V. D. Tran, X. L. Tran, and N. D. Hoang, "Image processing-based automatic detection of asphalt pavement rutting using a novel metaheuristic optimized machine learning approach," *Soft Computing*, vol. 25, no. 20, pp. 12839–12855, Oct. 2021, doi: 10.1007/s00500-021-06086-5.

[25] S. Vijh, S. Sharma, and P. Gaurav, "Brain tumor segmentation using Otsu embedded adaptive particle swarm optimization method and convolutional neural network," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 32, 2020, pp. 171–194, doi: 10.1007/978-3-030-25797-2_8.

[26] B. Khorram and M. Yazdi, "A new optimized thresholding method using ant colony algorithm for MR brain image segmentation," *Journal of Digital Imaging*, vol. 32, no. 1, pp. 162–174, Feb. 2019, doi: 10.1007/s10278-018-0111-x.

[27] S. Liu, "Image segmentation technology of the Ostu method for image materials based on binary PSO algorithm," in *Advances in Intelligent and Soft Computing*, vol. 104, 2011, pp. 415–419, doi: 10.1007/978-3-642-23777-5_68.

[28] M. Shehab, A. T. Khader, and M. Laouchedi, "A hybrid method based on Cuckoo search algorithm for global optimization problems," *Journal of Information and Communication Technology*, vol. 17, no. 3, pp. 469–491, Jun. 2018, doi: 10.32890/jict2018.17.3.8261.

[29] X. S. Yang, "A new metaheuristic Bat-inspired algorithm," in *Studies in Computational Intelligence*, vol. 284, 2010, pp. 65–74, doi: 10.1007/978-3-642-12538-6_6.

[30] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans Syst Man Cybern*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/tsmc.1979.4310076.

[31] S. L. Bangare, A. Dubal, P. S. Bangare, and S. T. Patil, "Reviewing otsu's method for image thresholding," *International Journal of Applied Engineering Research*, vol. 10, no. 9, pp. 21777–21783, May 2015, doi: 10.37622/ijaer/10.9.2015.21777-21783.

[32] J. L. Templos-Santos, O. Aguilar-Mejia, E. Peralta-Sanchez, and R. Sosa-Cortez, "Parameter tuning of PI control for speed regulation of a PMSM using bio-inspired algorithms," *Algorithms*, vol. 12, no. 3, p. 54, Mar. 2019, doi: 10.3390/A12030054.

[33] Q. Li, W. Lu, and J. Yang, "A hybrid thresholding algorithm for cloud detection on ground-based color images," *Journal of Atmospheric and Oceanic Technology*, vol. 28, no. 10, pp. 1286–1296, Oct. 2011, doi: 10.1175/JTECH-D-11-00009.1.

BIOGRAPHIES OF AUTHORS

Khamael Raqim Raheem received the B.Sc. degree in Computer Science from the University of Babylon, Iraq, the M.Sc. degree in Information Technology-AI Department from the University of UKM, Malaysia., and the Ph.D. degree in Information Technology with specialization in image processing from the University of Babylon, Iraq. Her research areas are text mining, image processing, named entity recognition, and recommendation system. She can be contacted at email: khmrakrah@atu.edu.iq.

Hafedh Ali Shabat received the B.Sc. degree in Computer Science from the University of Babylon, Iraq, the M.Sc. degree in Information Technology-AI Department from the University of UKM, Malaysia., and the Ph.D. degree in Information Technology with specialization in text mining from the University of Babylon, Iraq. His research areas are text mining, classification, machine learning, named entity recognition, and optimization. he used to hold an administrative post with the technical Institute of Babylon, AL-Furat AL-Awsat Technical University (ATU) director of the Centre of the computer. He can be contacted at email: h.ali@atu.edu.iq.