JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

A Multilevel Digital Image Thresholding Technique Based on an Enhanced Firefly Algorithm with Neighborhood Attraction

Abdulkarim Bashir Suleiman [1], Kana Armand Florentin Donfack [2], Abdulkarim Muhammad [2], and Muhammad Jumare Haruna [1,*]

[1] Department of Computer Science, Federal University of Education, Zaria, Kaduna State 810282, Nigeria; e-mail : abdulkarimbashir@rocketmail.com; muhammadjumare@gmail.com
[2] Department of Computer Science, Ahmadu Bello University, Zaria, Kaduna State 700225, Nigeria; e-mail : donfackkana@gmail.com; amuhd@abu.edu.ng
***** Corresponding Author : Muhammad Jumare Haruna

Abstract: Digital image segmentation is essential in image processing, influencing the accuracy of higher-level tasks. Thresholding is widely used, yet identifying optimal threshold values remains challenging. The Firefly Algorithm with Neighbourhood Attraction (FaNA), a metaheuristic approach, is efficient for color image thresholding but underperforms on grayscale images due to suboptimal thresholds. To overcome this, an enhanced version (eFaNA) was developed by integrating a chaotic tent map for population initialization and a Lévy flight-based random walk for improved exploration. eFaNA was compared with FaNA, fuzzy firefly algorithm (FFA), and the standard Firefly Algorithm (FA) in multilevel thresholding of grayscale images. Results demonstrate that eFaNA achieves superior segmentation quality with minimal detail loss, outperforming the others. The average PSNR obtained by eFaNA, FFA, FaNA, and FA was 25.5320 dB, 25.4075 dB, 24.1522 dB, and 24.4506 dB, respectively; average SSIM was 0.8641, 0.8604, 0.8432, and 0.6703; and execution time was 50.5322, 38.7726, 38.7528, and 107.6340 seconds, respectively. This reflects a PSNR improvement of 5.71% over FaNA, 0.49% over FFA, and 4.42% over FA, and an SSIM gain of 2.48% over FaNA, 0.43% over FFA, and 28.92% over FA. While eFaNA lags behind FFA and FaNA in execution time by ~11.8 seconds, it significantly outperforms FA. The performance gain is attributed to the chaotic tent map's diverse initialization and the Lévy flight's enhanced search capability. These improvements enable eFaNA to deliver consistently better threshold values and segmentation results. However, its relatively higher computational cost may limit applicability in real-time image processing.

Keywords: Digital image; Firefly algorithm with neighborhood attraction; Firefly optimization algorithm; Image segmentation; Multilevel thresholding.

Received: April, 16th 2025
Revised: May, 24th 2025
Accepted: May, 25th 2025
Published: May, 27th 2025

1. Introduction

Digital image processing is the application of various computing algorithms to process digital images [1]. Digital image segmentation is a core activity in digital image processing. It is the process of partitioning or separating digital images into their non-overlapping classes [2], [3], where pixels within the same class belong to the same object parts or background. It is a process that converts an image into distinct and uniform regions [4], [5], and it's done by grouping neighboring pixels that have coherent intensities [5], [6]. The aim is to change the representation of a digital image to make it more meaningful and simpler for easier analysis. It has several applications, such as in the diagnosis and monitoring of diseases, for example, in medical imaging, for finding moving objects in video sequence [7], in computer vision and robotics [8], in autonomous or self-driving cars [9], among others. A lot of digital image segmentation methods have been proposed in the literature, including region and boundary-growing-based methods [10], edge-based methods [11], clustering-based methods [12], Artificial Neural Network based method (ANN) [13] and thresholding-based methods [14].

Several metaheuristics algorithms have been used to find optimal thresholds for image segmentation. These include modified Grasshopper Optimization Algorithm (GOA) [14], [15], Oppositional Symbiotic Organism Search (OSOS) [16], [17], learning enthusiasm-based teaching-learning based optimization (LebTBLO) [18], [19] and firefly metaheuristic algorithm [4] among others. However, some gaps in the existing literature associated with metaheuristic algorithms for image thresholding include not obtaining excellent threshold values, which is denoted as the local optima problem and leads to poor threshold images. As a result, threshold images are distorted and are incomparable with ground truths, often measured in terms of Peak Signal Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), etc. Another issue with these algorithms is their low performance on colored and medical images such as Magnetic Resonance Images (MRI), X-rays, etc. In addition, these algorithms have high computational time, often referred to as the Execution time. This is the cumulative time these algorithms take to obtain resultant threshold values and segment digital images into constituents.

The Firefly algorithm is one of the most commonly used metaheuristic algorithms for image thresholding due to its performance and fewer parameters [14], [20]. However, it has high computational time when used to threshold digital images. To reduce the time this algorithm takes for thresholding digital images and their variants, the Firefly algorithm with Neighborhood Attraction (FaNA) and fuzzy firefly [21] algorithms were employed. Although FaNA threshold digital-colored images have low execution time, they do not produce optimal threshold values resulting in qualitative and accurate digital greyscale images. This work contributes to knowledge by developing a chaotic tent map-based enhanced firefly algorithm with neighborhood attraction that addresses the local optima threshold problem in FaNA for grayscale images, consequently making it yield accurately segmented digital greyscale images.

The rest of this paper is structured as follows. Section 2 review of related works. Section 3 describes the firefly algorithm with neighborhood attraction. Section 4 describes the proposed method. The results obtained by the proposed method and discussion are presented in section 5. Section 6 concludes the paper and outlines possible directions for future research.

2. Related Work

In the existing literature that seeks to find a solution to local optima, threshold utilizes metaheuristic algorithms. These algorithms can be broadly categorized into three: those based on standard metaheuristic algorithm without modification, those modified by hybridization with other metaheuristic algorithms, and those modified using other non-metaheuristic algorithms or schemes. For instance, the standard Cuckoo search (CS), FA, Particle Swarm Optimization (PSO), and Differential Evolution (DE) were used to search for optima thresholds that could segment grayscale images by [21]. Optima thresholds that could segment X-ray digital images were sorted using standard FA and BAT algorithms [22]. The authors in [23] seek to find an optimal threshold that could segment digital-coloured images using FFA, FaNA, Salp Swarm algorithm (SSA), basic firefly algorithm, Opposition Dimension Firefly Algorithm (ODFA), and Black Widow Optimization (BWO) algorithm. Optima thresholds for digital grayscale images were found by [24] using a Hybridized firefly with the Dragonfly algorithms and compared with GA, PSO, FA, BFO, and Electromagnetic Optimization (EMO) algorithms. In addition, [25] utilizes hybrid FA with PSO, Genetic Algorithm (GA), PSO, AMO (Animal Migration Optimization), and the standard FA to find optima thresholds for digital grayscale images. The standard FA and Social Spider Optimization (SSO) were also used to find optima thresholds for digital grayscale images [26]. A modified firefly algorithm based on lévy flight was also used in [4] and [23] to find optima thresholds for digital grayscale images. Study [27] seeks to find an optimal threshold for digital grayscale images using LFA and chaotic Bat algorithm, PSO, lévy Bat (LBAT), and Bacteria Foraging Optimization algorithm (BFO). In [28], a lévy flight-based FA and FFA were used to find optima thresholds for digital-colored images. Lévy flight-based FA, lévy flight-based BA, lévy flight based BFO, and lévy flight PSO were employed to find optima thresholds for segmenting noised stained gray scaled images by [29]. The authors in [30] and [31] utilize Brownian random walk-based FA, BFO, and CS to search for optima thresholds for colored RGB digital images.

3. Firefly Algorithm with Neighborhood Attraction

Firefly Algorithm with Neighborhood Attraction was proposed by [32]. The contribution of this work was to select fireflies that will be compared with the current fireflies to ascertain whether an update is necessary. FaNA is based on the k nearest neighbor strategy. The attraction of fireflies is such that fireflies are attracted by brighter fireflies selected from a predefined neighborhood rather than the entire population. The following modifications were made to the standard firefly algorithm in the FaNA algorithm. Randomization parameters, α, and attractiveness coefficient were updated according to Equations (1) and (2), respectively. The movement equation used was as shown in Equation (3) [32].

$$\alpha(t+1) = \left(\frac{1}{9000}\right)^{\frac{1}{t}} \alpha(t) \tag{1}$$

$$\beta = \beta_{min} + (\beta_0 - \beta_{min})e^{-\gamma r^2} \tag{2}$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + \beta\left(X_{jd}^{(t)} - X_{id}^{(t)}\right) + \alpha(t)S_d\varepsilon_i \tag{3}$$

Where S_d is the length scale of each variable. ε_i is a random number drawn from the uniform distribution. t is the iteration count, β_{min} is the minimum value of β, β_0 is the attractiveness constant, γ is the light absorption coefficient, r is the distance between fireflies, α is the randomization parameter, and e is the exponent. β was limited within the range $[\beta_0, \beta_{min}]$. α, γ, β_{min}, β_0 were initially set to 0.5, 1.0, 0.2 and 1.0 respectively. The Firefly with Neighbourhood Attraction algorithm and its flow charts are shown in Algorithm 1.

Algorithm 1. Firefly Algorithm with Neighborhood Attraction (FaNA) [32]

INPUT: Number of fireflies (N), Attractiveness Coefficient or constant, Absorption Coefficient or constant, Maximum number of iterations or cycle
OUTPUT: Best firefly (best threshold vector)

```
Randomly initialize the population of fireflies Fi where (i = 1, 2, …n) according to Algorithm 2.
Compute the fitness value of each firefly
    While FE ≤ MAXFEs  do
        Update the parameter α according to Equation (1)
            For i = 1 to N, do
                k= 3
                For j= i-k to i+k do
                    If j NE i, then
                        Set = (j + N) % N
                        Calculate the attractiveness according to Equation (2).
                        If f(Fj) > f(Fi)
                            Move Fi towards Fj according to Equation (3).
                            Compute the fitness value of the new Fi
                        End
                    End
                Next j
            End
        Next i
    End
    FE++
End.
Return the best firefly
```

Algorithm 2. Random Initialization scheme

INPUT: UB, LB, population size, dimension
OUTPUT: Initial populations of fireflies, F

```
i=0, j=0
while (i <= population size)
```

Algorithm 2. Random Initialization scheme *(continue)*

```
        while (j <= dimension)
            F(i,j)=(UB-LB) *rand (0,1) +LB
            j=j+1
        End while
    i =i+1
End while
```

Where UB is the upper bound, LB is the lower bound, and rand (0,1) is the random function that returns values between 0 and 1.

4. Proposed Method

This paper enhanced the firefly algorithm with neighborhood attraction to threshold digital grayscale images by introducing a chaotic initialization scheme and lévy flight-based random walk mechanism. The enhanced algorithm works as follows. The first step computes the initial solutions using a chaotic tent map, the second step evolves a new population based on lévy flight random walk, the third step finds the fitness of each firefly, and the fourth step compares the fitness of each firefly with one another. The fifth step moves towards brighter firefly within the specified neighborhood, and the sixth step computes the new fitness of the evolved fireflies. These procedures are repeated until a stopping condition or the maximum number of iterations is reached. Algorithm 3 and Figure 1 show the algorithm and proposed method flow chart. In the context of the enhanced firefly algorithm with neighborhood attraction, referred to as the proposed method. The movement equation of the firefly with neighborhood attraction in Equation (3) was modified by setting the attractiveness term to zero and adding lévy flight random walk, fireflies' individuals were initialized as expressed in Equation (4).

$$F_{levy} = \text{chaotic}F + L(d) \tag{4}$$

Where $\text{chaotic}F$ is a population of chaotic fireflies. F_{levy} is the lévy based population of fireflies. T is the iteration count, which is the dimension. L is a vector of random numbers generated using the lévy distribution obtained using Equations (5)-(11).

Algorithm 4. Proposed enhanced Firefly Algorithm with Neighborhood Attraction (eFaNA)

```
INPUT: Number of fireflies, attractiveness coefficient or constant, absorption coefficient
or constant, maximum number of iterations or cycle
OUTPUT: Best firefly (best threshold vector)
Initialize the firefly population using chaotic tent map according to Algorithm 6
Evolve lévy flight-based firefly population using Algorithm 5
    Compute the fitness value of each firefly
        While FE ≤ MAXFE    do
            Update the parameter α according to Equation (1)
                For i = 1 to N do
                        k= 3
                        For j= i-k to i+k do
                                If j NE i then
                                    Set j = (j + N) % N
                                        Calculate the attractiveness according to Equation (2)
                                            If f(Fj) > f(Fi)
                                                Move Fi towards Fj according to Equation (3).
                                                Compute the fitness value of the new Fi
                                            End
                                        End
                                    Next j
                                End
                            Next i
                    End
```

Algorithm 4. Proposed enhanced Firefly Algorithm with Neighborhood Attraction (eFaNA)*(continue)*

```
21:          FE++
22: End.
23: Return the best firefly
```

4.1. Random Walk

A random walk is a random process that consists of taking a series of random steps consecutively [33]. It is one of the strategies for generating candidate solutions in a solution space. Various random walks, such as the Lévy flight and Brownian Walk, may result depending on the distribution from which the step size or length is drawn. The Lévy flight strategy is better than other strategies as it explores the search space more efficiently and has a longer step length [28]. In Lévy flight, the step size is drawn from lévy distribution expressed as a power law as in Equation (5).

$$L(s) \sim |s|^{-1-\beta} \tag{5}$$

Where β is an index or exponent and s is the step length. β is within the range $0 < \beta \leq 2$. To generate the size of the random steps, we adopted Mantegna's definition used in Nadimi-Shahraki et al. [34], and was calculated as in Equation (6).

$$s = 0.01 \frac{u}{|y|^{1/\alpha}} \tag{6}$$

Where α is lévy index and is taken as 1.5, u and y are drawn from normal distributions, denoted as Equations (7) -(9) [33].

$$\delta_u = \left(\frac{\Gamma\left(\frac{1+\alpha}{2}\right) \cdot \alpha \cdot 2^{\frac{\alpha-1}{2}}}{\Gamma(1+\alpha) \cdot sin\left(\frac{\pi\alpha}{2}\right)} \right)^{\frac{1}{\alpha}} \tag{7}$$

$$u \sim N\left(0, {\delta_u}^2\right) \tag{8}$$

$$y \sim N(0,1) \tag{9}$$

Where δ_u and δ_v defined in Equations (10) and (11), respectively [33].

$$\delta_u = \left[\frac{\Gamma(1+\beta)\sin(\pi\beta/2))}{\lambda\Gamma\left(1+\frac{\beta}{2}\right)\beta 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}} \tag{10}$$

$$\delta_v = 1 \tag{11}$$

Where $\lambda = 1.5$, Γ is the Gamma function. Lévy flight-based fireflies evolved according to Algorithm 5.

Algorithm 5. Lévy flight Based Scheme

```
INPUT: population size, dimension, chaoticF
OUTPUT: lévy flight generated populations of fireflies, F_levy
1:  i =0, j=0
2:  while (i <= population size)
3:        while (j <= dimension)
4:              Evolve lévy flight-based population using Equation (4).
5:         j=j+1
6:      End while
7:    i =i+1
8:  End while
```

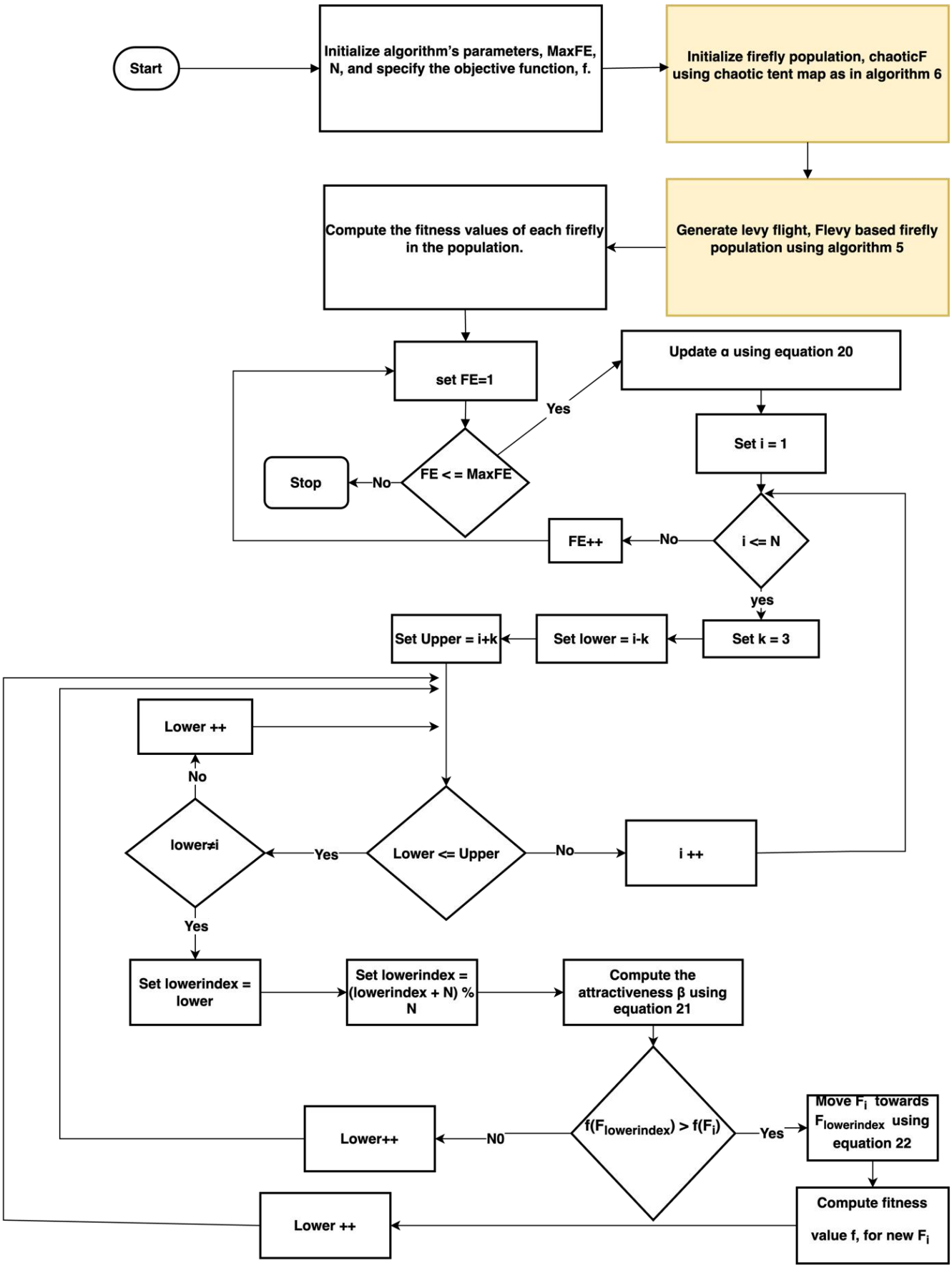


Figure 2. Flow chart of the enhanced firefly Algorithm with Neighborhood Attraction (eFaNA)

4.2. Chaotic Tent Map

Random initialization of individuals in the population may lead to an unequal distribution of the individuals within the solution space, which may cause algorithms to be struck in local optima and converge prematurely. To solve these problems, chaotic tent maps are often used. They are of utmost importance in improving the performance of evolutionary algorithms as they help them avoid local optima and speed up convergence [35]. There are various chaotic maps, but we adopted the tent map used by [36]. A tent map is a one-dimensional map that exhibits good chaotic behavior. It has better ergodicity and randomicity than random distribution and can improve the global search ability of fireflies. The chaotic Tent map is mathematically expressed in Equation (12) [36].

$$x_{k+1} = \begin{cases} 2x_k, & 0 \le x_k \le 0.5 \\ 2(1 - x_k), & 0.5 < x_k \le 1 \end{cases} \quad (12)$$

Where x_k ranges from 0 to 1. The tent map generates a chaotic sequence in (0,1). Fireflies are initialized according to the initialization scheme presented in Algorithm 6.

Algorithm 6. Chaotic tent map-based initialization scheme

INPUT: UB, LB, population size, dimension
OUTPUT: Initial populations of fireflies, chaotic$F$

```
i=0, j=0
while (i <= population size)
    while (j <= dimension)
        Randomly initialize variables Fi,j
            If (Fi,j < = 0.5)
            // Map Fi,j  to chaoticFi,j  using equation 30 as follows:
                Chaotic F_{i,j} = 2 * F_{i,j} * (UB − LB) + LB
            Else
                Chaotic F_{i,j} = 2 * (1 − F_{i,j}) * (UB − LB) + LB
        j=j+1
    End while
i =i+1
End while
```

where UB, is the upper bound and LB is the lower bound.

4.3. Enhanced Firefly Algorithm With Neighbourhood Attraction For the Multilevel Thresholding of Digital Gray Scale Images

This section presents how the enhanced algorithm(eFaNA) was applied to threshold grayscale images. The implementation is detailed as in Algorithm 7.

Algorithm 7. Applying eFaNA for thresholding of grayscale images

INPUT: Grayscale image, I_g
OUTPUT: Grayscale image, I_g, Threshold image $I_{th,}$ PSNR, SSIM values, Execution time and standard deviation

1: Read the grayscale image, Ig
2: Compute the histogram of the image, Ig
3: Compute the occurrence probabilities of each gray scale level in Ig.
4: Execute Algorithm 4 using Otsu as an objective function, f.
5: Use the obtained optimal threshold vector in step 4 above to segment Ig, resulting in Ith.
6: Compute the PSNR, RMSE, SSIM values, Execution time, mean, and standard deviation defined as in Equation (12)-(17), respectively.

4.4 Evaluation Metrics and Tools

To measure and compare the performances of the four different algorithms compared in this work, the following evaluation metrics were used. These metrics are explained below.

4.4.1. Peak Signal to Noise Ratio (PSNR)

This metric measures the quality of threshold images. Equation (13) [37] is the mathematical formula for calculating PSNR.

$$PSNR_{(I,S)} = 20log_{10}\left(\frac{Max}{RMSE_{(I,S)}}\right) \tag{13}$$

Where $RMSE$ is the root-mean-square-error expressed as in Equation (14) [37], and Max is the maximum pixel value in the given image.

$$RSME_{(I,S)} = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j) - S(i,j)\right)^2}{M * N}} \tag{14}$$

Where I is the original image, and S is the segmented image with size $M * N$. $PSNR$ is measured in decibel (db).

4.4.2. Structural Similarity Index Metric (SSIM)

This evaluates the likeness between the original and segmented image. This is computed as shown in Equation (15) [37].

$$SSIM_{(i\ s)} = \frac{(2\mu_i\mu_s + C_a)(2\delta_i\delta_s + C_b)}{(\mu_i^2 + \mu_s^2 - C_a)(\delta_i^2 + \delta_s^2 + C_b)} \tag{15}$$

Where i and s are original and segmented images, respectively. $\mu_i\mu_s$ are average values. δ_i^2 and δ_s^2 are the variances. $\delta_i\delta_s$ is the covariance. C_a = $(k_1L)^2$ and C_b= $(k_2L)^2$ stabilize the division with weak denominator, with L = 256, k_1 = 0.01, and k_2= 0.03.

4.4.3. Execution Time

This is the algorithm's cumulative time to obtain the best threshold values and segment a given grayscale image using the obtained threshold values (given in seconds). The execution time is the sum of the time the algorithm takes to obtain optimal thresholds and segment the grayscale image. It is calculated as expressed in Equation (16).

$$\text{Execution Time} = \sum \text{time taken to obtain thresholds} + \text{time taken to segment image} \tag{16}$$

This Equation denotes that the overall execution time equals the time it takes for an algorithm to search for an optima threshold for a given image and the time it takes for the algorithms to use the optima threshold to segment the image in consideration.

4.4.4. Mean

The average PSNR, fitness value, execution time, and SSIM are computed in Equation (17).

$$mean = \sum x_i/n \tag{17}$$

4.4.5. Standard Deviation

This finds out the stability of the evaluated algorithms. The value of the standard deviation should be as small as possible. It is expressed mathematically as in Equation (18).

$$STD = \sqrt{\sum_{i=1}^{maxIter}\frac{(\sigma - \mu)}{maxIter}} \tag{18}$$

4.4.6. Hardware and Software Tools

The hardware and software used throughout this study's experiments was an HP 15 Notebook computer system with the following configurations: Intel Core i3, CPU N3510, 1.99 GHz processor, and 8 GB RAM. The algorithms in this study were implemented using MATLAB version R2023a on a 64-bit Windows 10 Pro operating system.

4.4.7 Experimental Datasets

Five different digital grayscale Image datasets were used for the experimental analysis. These datasets include Lena, Mandrill, Traffic, Cameraman, and Livingroom. Each grayscale image is 512 by 512 dimensions and was selected because they have different characteristics, and the proposed algorithm's performance was evaluated on them.

4.4.8 Experimental Settings

The initial population of fireflies used was set to 25. The number of runs was 30, and the maximum number of iterations (MaxFE) was set to 100. The initial values of α, γ, β_{min}, β_0 were initially set to 0.5, 1.0, 0.2 and 1.0 respectively. β was limited within the range $[\beta_0, \beta_{min}]$. Optimal thresholds were obtained for each grayscale image test data set at 2, 3, 4, and 5 threshold levels.

5. Results and Discussion

This section presents the optimal thresholds obtained by the algorithms. It also discusses the qualities of the threshold image obtained by the algorithms and compares them regarding PNSR, SSIM, and execution time. Samples of the threshold digital images obtained by the four algorithms compared were also presented.

Table 1. Optimal threshold distributions were obtained by FA, FFA, FaNA and eFaNA for each image dataset.

Images	Algorithms	Threshold levels			
		2	3	4	5
Lena	FA [21]	84, 120	42, 85, 110	111, 136, 152 217	125, 155, 187, 232, 245
	FaNA [32]	105, 155	24, 92, 114	27, 106, 115, 255	24, 25,62, 84, 120
	eFaNA [Ours]	99, 164	134, 146, 240	32,78, 89, 138	18, 39, 80, 88, 98
	FFA [23]	101, 164	65, 99, 173	61, 86,101,167	24, 91,138,203,255
Cameraman	FA [21]	63,119	61,76, 207	5,13, 16, 68	6,19,39,63,131
	FaNA [32]	73, 142	8,36, 86	8,46,80, 223	8,8,55,59,119
	eFaNA [Ours]	127,194	60,111, 197	22,52,88,201	48,62,81,115,190
	FFA [23]	123,193	107,145,199	37,91,121, 200	62,89,103,141,194
Mandrill	FA [21]	83,125	14, 37, 69	20,43,71, 122	27,62,99,119,171
	FaNA [32]	75,138	38,83,139	38, 66, 97, 155	48,76, 109, 147, 170
	eFaNA [Ours]	70,135	46,88,145	33,66,107,159	35,74,99,120, 165
	FFA [23]	91,140	145,149, 250	104,140,146,219	133,161,173,220,255
Traffic	FA [21]	107,136	68,92,157	39,66,90,104	34,36,67,68,93
	FaNA [32]	86,138	35,75,94	46,67,79,183	38,45,64,76,102
	eFaNA [Ours]	92,155	44,71,90	32,61,84,135	41,39,71,81,82
	FFA [23]	110,163	120,146,184	124,147,178,224	100,129,164,189,223
Livingroom	FA [21]	106,140	108,137,234	36,65,106,188	4,13,44,72,158
	FaNA [32]	91,136	115,139,227	40,55,87,135	11, 9,58,71,115
	eFaNA [Ours]	87,141	79,87,141	21,55,92,151	33,27,67,83,109
	FFA [23]	68,149	31,84,199	40,57,75,120	56,97,127,154,207

Table 1 shows the optimal threshold distributions obtained by FA, FFA, FaNA, and eFaNA for each image dataset used in this work. Table 8 is a sample of the threshold digital grayscale images obtained for traffic images using the optimal thresholds in Table 1. These threshold distributions were obtained using the Otsu between class variance objective function.

Table 2 depicts the quality of the threshold digital images in terms of the average PSNR obtained by the algorithms compared at different threshold levels. Table 2 shows that as the threshold levels increase, the average PSNR values increase, proving that eFaNA can effectively threshold digital grayscale images. Table 3 compares the overall average PSNR obtained by FA, FFA, FaNA, and eFaNA at all the thresholds considered. From this table, it can be seen that eFaNA obtained an overall average PSNR of 25.530db, FFA obtained an overall

average PSNR of 25.40750db, FaNA obtained an overall average PSNR of 24.1522db and the standard firefly algorithm obtained an overall average PSNR of 24.4506db. This result means that eFaNA has an overall average PSNR improvement of about 5.7130% over FaNA, an overall average PSNR improvement of about 0.4899% over FFA, and an overall average PSNR improvement of 4.422% over FA. In addition, this table shows that eFaNA is generally more stable at obtaining PSNR values because it has a better overall average standard deviation of 1.5361db against an overall average standard deviation of 2.8317db by FaNA, 2.4742db by FFA and 4.6778db by FA. Figure 3 is a bar chart that compares the overall average PSNR obtained by the four algorithms.

Table 2. The quality of the threshold digital images in terms of the average PSNR obtained by the algorithms compared at different threshold levels.

Images	Threshold levels	Objective function: Otsu Between Class Variance							
		FA [21]		FFA [23]		FaNA [32]		eFaNA [Ours]	
		mean	std	mean	std	mean	std	mean	std
Lena	2	21.8214	3.0759	23.7961	0.4504	21.8754	0.5365	23.9082	0.2317
	3	23.1287	6.2467	23.5254	2.6749	22.7451	1.2203	24.6303	0.4017
	4	24.7947	3.4172	25.1945	2.6465	25.0653	1.4582	25.6268	1.3210
	5	25.6454	3.0759	25.8110	2.4362	25.7623	3.1982	25.8180	2.0390
Cameraman	2	24.7549	2.1207	25.1181	0.5842	24.6830	0.4146	25.1987	0.2015
	3	27.7017	7.0991	27.7224	4.7226	27.4329	1.9745	27.4720	1.2971
	4	28.3022	6.2449	28.5932	2.5648	28.5597	2.4117	29.8449	1.0361
	5	28.4279	7.1129	29.7767	5.0346	28.5855	3.8023	29.9071	2.4051
Mandrill	2	22.4423	2.3723	23.6565	0.3874	22.7159	0.9198	21.8057	0.9428
	3	25.2999	10.5197	25.8325	1.8346	24.4329	8.1106	26.7520	1.8159
	4	25.6669	4.5932	26.7137	2.8743	24.5102	4.0949	27.8582	2.9207
	5	25.9365	4.2342	26.7663	4.5048	25.3167	3.0243	27.1042	2.7415
Traffic	2	17.0845	3.5946	21.1457	1.1490	18.8816	1.3676	20.9299	0.6313
	3	22.9086	4.8921	23.6673	2.6896	21.3936	3.2791	23.5473	0.9761
	4	23.7890	3.8510	23.9746	2.6502	22.7832	4.1579	24.9936	2.9775
	5	24.2707	4.5429	25.4895	3.6255	22.9630	5.1796	25.5443	2.4702
Livingroom	2	22.1896	2.4543	23.4492	1.0340	21.8816	1.5814	23.4572	0.4021
	3	24.7275	7.3624	24.8571	2.7996	21.9254	3.3151	22.1101	1.6249
	4	24.4537	2.6040	26.0870	2.3562	25.6589	3.2092	26.9642	2.1359
	5	25.6659	4.1414	26.9739	2.4650	25.8723	3.3782	27.1679	2.1494

Table 3. The overall average PSNR and standard deviation obtained by the algorithms at each threshold level considered for each of the five images.

Mean and deviation at	Average PSNR ± STD (dB)							
	FA [21]		FFA [23]		FaNA [32]		eFaNA [Ours]	
	PSNR	std	PSNR	std	PSNR	std	PSNR	std
Levels 2 Threshold	21.6585	2.7236	23.4331	0.7210	22.0075	0.9640	23.0599	0.4819
Levels 3 Threshold	24.7533	7.2240	25.12094	2.9442	23.5860	3.5799	24.9023	1.2231
Levels 4 Threshold	25.4013	4.1421	26.1126	2.6184	25.3155	3.0664	27.0575	2.0782
Levels 5 Threshold	25.9893	4.6215	26.9635	3.6132	25.7000	3.7165	27.1083	2.3610
Average	24.4506	4.6778	25.4075	2.4742	24.1522	2.8317	25.5320	1.5361

Table 4 shows the SSIM values of the threshold digital images obtained by each algorithm at different threshold levels. This table shows that the enhanced FaNA often obtains a slightly better average SSIM as the threshold increases. Table 5 presents the overall average SSIM obtained by the algorithms. This table shows that eFaNA has an overall average SSIM

of 0.8641, FFA has an overall average SSIM of 0.8604, FaNA has an overall average SSIM of 0.8432, and FA has an overall average SSIM of 0.6703. This means that eFaNA has an overall average SSIM improvement of 2.4768% over FaNA, an overall average SSIM improvement of 0.43237% over FFA, and an overall average SSIM improvement of 28.9198% over the standard firefly algorithm. Moreover, this table shows that eFaNA is more stable as it has a better standard deviation than other algorithms in consideration, with an overall average standard deviation of 0.0479 against 0.1300 for FaNA, 0.0831 for FFA, and 0.0872 for FA. Figure 4 shows the overall average SSIM and the overall average deviation from mean SSIM by the algorithms. This result is comparable with those obtained in FA [21], FFA [23], and FaNA [32].

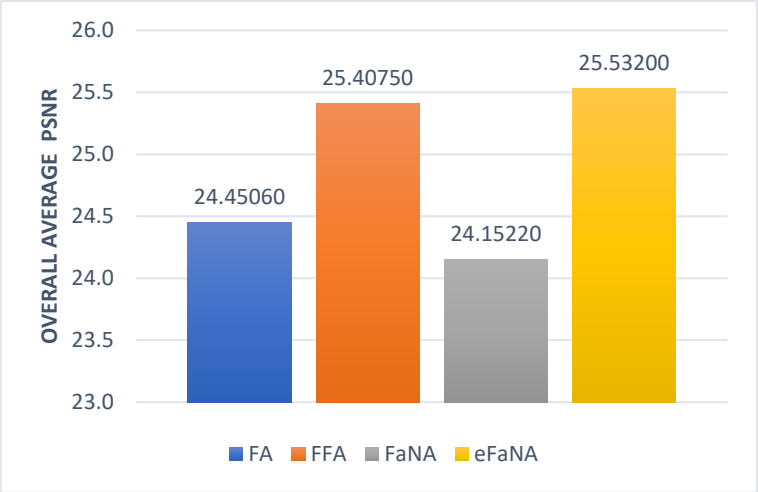


Figure 3. A comparison of the overall average PSNR obtained by FA, FFA, FaNA and eFaNA

Table 4. Comparison of Mean and standard deviation of Structural Similarity Index Measure (SSIM) for each algorithm at different threshold levels.

Images	Threshold levels	Objective function: Otsu Between Class Variance							
		FA [21]		FFA [23]		FaNA [32]		eFaNA [Ours]	
		mean	std	mean	std	mean	std	mean	std
Lena	2	0.8416	0.0445	0.8465	0.0071	0.8450	0.0076	0.8514	0.0045
	3	0.8433	0.0940	0.8548	0.0398	0.8442	1.2203	0.8632	0.0243
	4	0.8673	0.0841	0.8698	0.0428	0.8476	0.0653	0.8738	0.0229
	5	0.8726	0.0818	0.8740	0.1327	0.8594	0.1079	0.8846	0.0543
Cameraman	2	0.8713	0.0313	0.8737	0.0090	0.8656	0.0057	0.8768	0.0042
	3	0.8808	0.0668	0.8891	0.0534	0.8749	0.0474	0.8524	0.0156
	4	0.8104	0.0698	0.8769	0.0508	0.8525	0.0511	0.8774	0.0358
	5	0.8813	0.1276	0.8834	0.1901	0.8338	0.0762	0.8954	0.1000
Mandrill	2	0.8161	0.0578	0.7738	0.0130	0.8684	0.0022	0.8533	0.0026
	3	0.8569	0.1027	0.8702	0.0421	0.8628	0.0903	0.8674	0.0144
	4	0.8653	0.1248	0.8758	0.0560	0.8651	0.1394	0.8763	0.0371
	5	0.8780	0.1186	0.8832	0.1975	0.8807	0.0765	0.8892	0.1278
Traffic	2	0.7058	0.0697	0.7913	0.0229	0.7393	0.0280	0.8011	0.0113
	3	0.8071	0.0835	0.8719	0.1067	0.7845	0.0848	0.8502	0.1002
	4	0.8060	0.1031	0.8317	0.1214	0.8030	0.1173	0.8562	0.0611
	5	0.8317	0.1296	0.8523	0.1528	0.8189	0.1564	0.8607	0.1200
Livingroom	2	0.8010	0.0557	0.8494	0.0332	0.7960	0.0387	0.8559	0.0128
	3	0.8971	0.1138	0.8812	0.0682	0.8724	0.0739	0.8325	0.0469
	4	0.8713	0.0736	0.8774	0.1335	0.8760	0.0884	0.8799	0.0549
	5	0.8783	0.1104	0.8820	0.1886	0.8741	0.1227	0.8842	0.1072

Table 5. Comparison of the average SSIM and standard deviation obtained by FA, FFA, FaNA, and eFaNA for each five digital grayscale images.

Mean and deviation at	Average SSIM ± STD							
	FA [21]		FFA [23]		FaNA [32]		eFaNA [Ours]	
	SSIM	std	SSIM	std	SSIM	std	SSIM	std
Levels 2 Threshold	0.8072	0.0518	0.8269	0.0170	0.8229	0.0164	0.8477	0.0071
Levels 3 Threshold	0.8570	0.0922	0.8734	0.0620	0.8478	0.3033	0.8531	0.0403
Levels 4 Threshold	0.8441	0.0911	0.8663	0.0809	0.8488	0.0923	0.8727	0.0424
Levels 5 Threshold	0.8684	0.1136	0.8749	0.1723	0.8534	0.1079	0.8828	0.1019
Average	0.8442	0.0872	0.8604	0.0831	0.8432	0.1300	0.8641	0.0479

Table 6 shows the average execution time obtained by the four algorithms at different threshold levels on all the digital grayscale images. Table 7 shows the overall average execution time obtained by all the algorithms. From this table, it can be seen that eFaNA has an overall average execution time of 50.5322 seconds against an overall average Execution time of 38.7528 seconds for FaNA, an overall average Execution time of 38.7726 seconds by FFA, and an overall average execution time of 107.6340 seconds for FA. The overall average deviation from the mean execution time obtained by the algorithms is shown in Figure 5. This table shows that eFaNA has an average standard deviation from the mean Execution time of 6.6052 seconds, against 4.9404 seconds for FaNA, 6.0480 seconds for FFA, and 16.6569 seconds for FA. This result shows that FaNA outperforms eFaNA by segmenting digital grayscale images in lesser execution time compared to eFaNA and FA. However, eFaNA has better segmentation time efficiency than FA.

Table 6. Comparison of the mean and standard deviation of execution time (in seconds) for each algorithm at different thresholds levels using Otsu objective function.

Images	Threshold levels	Objective function: Otsu Between Class Variance							
		FA [21]		FFA [23]		FaNA [32]		eFaNA [Ours]	
		time	std	time	std	time	std	time	std
Lena	2	88.2098	19.1110	37.3723	5.8517	31.1274	0.8027	37.0262	6.8251
	3	118.2067	16.0284	39.2615	2.9556	37.4807	3.9699	35.4648	2.9037
	4	99.6616	24.6340	49.8117	9.6474	49.7550	10.8344	46.6747	12.8143
	5	107.3812	25.4585	58.4643	10.7389	68.3415	8.7257	60.0711	2.9462
Cameraman	2	78.7180	16.2980	36.2381	4.4529	34.1808	1.7240	32.4599	8.5185
	3	123.0385	3.0401	51.6079	1.4535	64.7530	1.3365	56.7885	3.2075
	4	102.1132	19.3959	54.5218	12.0404	64.7530	9.9271	55.1692	12.0563
	5	111.8808	21.0630	55.9598	9.4755	70.4936	12.4704	62.3241	12.7078
Mandrill	2	100.2471	21.1944	39.07953	9.1064	37.3175	7.7764	69.2402	4.8273
	3	137.9080	7.6201	46.99347	1.7488	41.5267	1.8933	60.5505	12.9591
	4	116.8505	25.0848	53.34716	12.0245	70.5606	8.8091	62.6374	13.0046
	5	146.1605	28.1303	47.11662	11.4617	54.6418	10.3641	98.4154	6.7694
Traffic	2	113.5037	4.4289	18.1378	10.3459	17.9014	1.2861	25.4931	2.6515
	3	85.8978	7.5799	32.3606	2.9205	16.2971	3.7287	31.2785	1.9907
	4	108.0225	21.1942	36.0326	3.7328	18.0689	3.9621	38.7988	3.0970
	5	109.8973	9.9325	47.1701	6.6616	27.0862	1.6898	45.5761	6.5386
Livingroom	2	103.3260	21.0198	18.5705	0.4924	17.3660	0.9608	30.4280	6.4810
	3	94.4083	20.2983	14.6174	3.5500	18.8603	3.3811	45.1375	7.0005
	4	101.4389	15.6460	18.3912	1.0076	14.9513	0.7861	56.5279	1.7283
	5	105.8101	5.9803	20.3981	1.2909	19.5937	4.3802	60.5817	3.0762

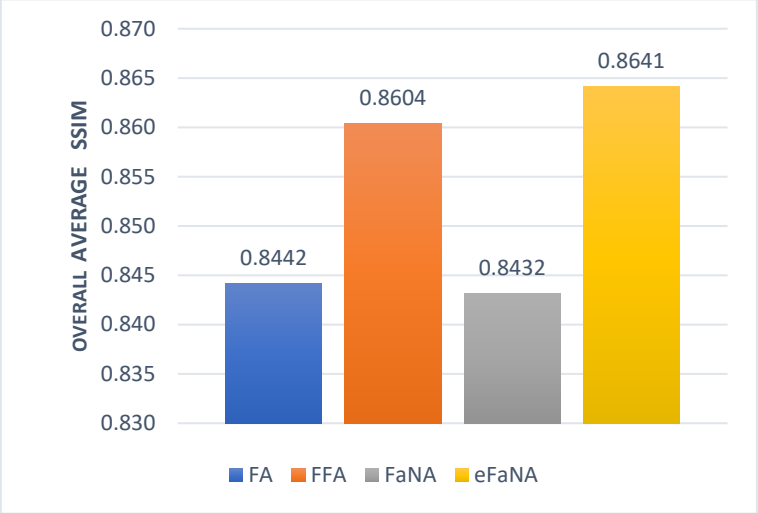


Figure 4. A comparison of the overall average standard deviation from the mean SSIM obtained by FA, FFA, FaNA, and eFaNA

Table 7. Comparison of the overall mean execution time (seconds) and its deviation obtained by FA, FFA, FaNA, and eFaNA.

Mean and deviation at	Average execution time ± STD							
	FA [21]		FFA [23]		FaNA [32]		eFaNA [Ours]	
	time	std	time	std	time	std	time	std
Levels 2 Threshold	96.8009	16.4104	29.8796	6.0499	27.5786	2.5100	38.9295	5.8607
Levels 3 Threshold	111.8919	10.9134	36.9682	2.5257	35.7836	2.8619	45.8440	5.6123
Levels 4 Threshold	105.6173	21.1910	42.4209	7.6905	43.6178	6.8638	51.9616	8.5401
Levels 5 Threshold	116.2260	18.1129	45.8218	7.9257	48.0314	7.5260	65.3937	6.4076
Average	107.6340	16.6569	38.7726	6.0480	38.7528	4.9404	50.5322	6.6052

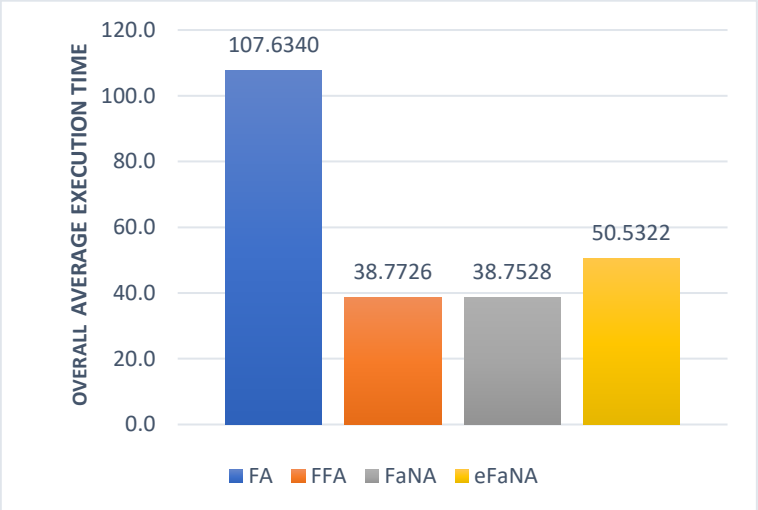
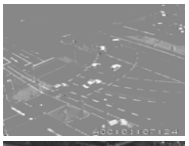


Figure 5. Comparison of the overall mean execution time obtained by FA, FFA, FaNA, and eFaNA for each considered threshold level.

A sample of the resultant traffic threshold images obtained are as shown in Table 8 for all the compared algorithms at threshold levels 2 to 5. Based on this table, it can be seen that FA threshold images with major defects compared to the ground truth, while FFA, FaNA and eFaNA threshold images with minimal defects with the traffic images yielded by eFaNA being more visible with the least minimal defects

Table 8. The compared algorithms determined the traffic image threshold.

Method	Threshold levels			
	2	3	4	5
FaNA [32]				
FA [21]				
FFA [23]				
eFaNA [ours]				

6. Conclusions and Recommendation

This work enhanced the Firefly algorithm with neighborhood attraction for digital gray scale image thresholding. The results obtained reveal that eFaNA is comparable to FFA at thresholding digital gray scale image. The enhanced FaNA is more consistent at yielding optimal threshold values that segment digital gray scale images without much loss of important image details compared to FA and FaNA and FFA, but may not be a good option for real time applications. These results also show that, enhancing firefly algorithm with neighborhood attraction by introducing chaotic tent map and lévy flight based random walk into the algorithm can improve the algorithm's ability at yielding better threshold values, for segmenting digital gray scale images. This result was obtained because, the chaotic tent map included in the algorithm generates more better initial population of fireflies and the lévy flight based random walk scheme improved its exploration capability. Therefore, good initial population of fire-flies are used to begin the search for optimal solutions that were used to segment the given digital gray scale images. The enhanced firefly algorithm with neighborhood attraction proposed in this work yield threshold values that segments digital gray scale images and obtained threshold images with better PSNR and SSIM. It is however, not a good choice for applications involving real time thresholding of digital gray scale images. Its performance compared to other metaheuristics algorithms is also not known. Also, the performance of the enhanced algorithm using other objective functions is not known. In addition, the performance of the algorithm on colored digital images was not investigated in this work.

Despite these promising results, several limitations remain. The algorithm's performance has not been benchmarked against other metaheuristic methods such as PSO, ABC, ACO, or ANN-based approaches for gray scale image thresholding. Additionally, its effectiveness using alternative objective functions, such as Tsalli, Kapur, or Minimum Cross Entropy (MCE), has yet to be explored. Lastly, the algorithm was not evaluated on colored digital images. Addressing these limitations in future research would provide a more comprehensive understanding of the enhanced algorithm's capabilities and applicability.

Author Contributions: Conceptualization: Abdulkarim Bashir Suleiman and KAF Donfack; methodology, Abdulkarim Bashir Suleiman and Muhammad Jumare Haruna; software: Muhammad Jumare Haruna; validation: Abdulkarim Bashir Suleiman, Abdulkarim Muhammad., and Donfack A Kana; formal analysis: Abdulkarim Bashir Suleiman; investigation: Abdulkarim Bashir Suleiman; resources: Muhammad Jumare Haruna and Abdulkarim Muhammad; data curation: Abdulkarim Bashir Suleiman; Writing—original draft preparation: Abdulkarim

Bashir Suleiman; Writing—review and editing: KAF Donfack and Muhammad Jumare Haruna; Visualization: Abdulkarim Bashir Suleiman and Muhammad Jumare Haruna; Supervision: KAF Donfack and Abdulkarim Muhammad; Project administration: Abdulkarim Muhammad and KAF Donfack; Funding acquisition: Abdulkarim Bashir Suleiman and Muhammad Jumare Haruna. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] W. Zheng, "Current Technologies and Applications of Digital Image Processing," *J. Biomed. Sustain. Healthc. Appl.*, vol. 3, no. 1, pp. 13–23, Jan. 2023, doi: 10.53759/0088/JBSHA202303002.

[2] N. M. Zaitoun and M. J. Aqel, "Survey on Image Segmentation Techniques," *Procedia Comput. Sci.*, vol. 65, pp. 797–806, 2015, doi: 10.1016/j.procs.2015.09.027.

[3] J. C. Tilton, S. Aksoy, and Y. Tarabalka, "Image segmentation algorithms for land categorization," in *Remote Sensing Handbook, Volume II*, CRC Press, 2024, pp. 196–232. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.1201/9781003541158-11/image-segmentation-algorithms-land-categorization-james-tilton-selim-aksoy-yuliya-tarabalka

[4] A. Sharma, R. Chaturvedi, U. K. Dwivedi, S. Kumar, and S. Reddy, "Firefly algorithm based effective gray scale image segmentation using multilevel thresholding and entropy function," *Int. J. Pure Appl. Math.*, vol. 118, no. 5, pp. 437–443, 2018, [Online]. Available: https://www.acadpubl.eu/jsi/2018-118-5/articles/5/25.pdf

[5] Y. Yu *et al.*, "Techniques and Challenges of Image Segmentation: A Review," *Electronics*, vol. 12, no. 5, p. 1199, Mar. 2023, doi: 10.3390/electronics12051199.

[6] A. Hossain *et al.*, "Identifying the retinal layers from optical coherence tomography images using a 3D segmentation method," *IET Image Process.*, vol. 19, no. 1, p. e13306, Jan. 2025, doi: 10.1049/ipr2.13306.

[7] B. Abhisheka, S. K. Biswas, B. Purkayastha, D. Das, and A. Escargueil, "Recent trend in medical imaging modalities and their applications in disease diagnosis: a review," *Multimed. Tools Appl.*, vol. 83, no. 14, pp. 43035–43070, Oct. 2023, doi: 10.1007/s11042-023-17326-1.

[8] T. Habuza *et al.*, "AI applications in robotics, diagnostic image analysis and precision medicine: Current limitations, future trends, guidelines on CAD systems for medicine," *Informatics Med. Unlocked*, vol. 24, p. 100596, 2021, doi: 10.1016/j.imu.2021.100596.

[9] A. Karnati and D. Mehta, "Artificial intelligence in self driving cars: Applications, implications and challenges," *Ushus J. Bus. Manag.*, vol. 21, no. 4, pp. 1–28, 2022, doi: 10.12725/ujbm/61.1.

[10] N. Kheradmandi and V. Mehranfar, "A critical review and comparative study on image segmentation-based techniques for pavement crack detection," *Constr. Build. Mater.*, vol. 321, p. 126162, Feb. 2022, doi: 10.1016/j.conbuildmat.2021.126162.

[11] J. Jing, S. Liu, G. Wang, W. Zhang, and C. Sun, "Recent advances on image edge detection: A comprehensive review," *Neurocomputing*, vol. 503, pp. 259–271, Sep. 2022, doi: 10.1016/j.neucom.2022.06.083.

[12] K. K. D. Ramesh, G. K. Kumar, K. Swapna, D. Datta, and S. S. Rajest, "A Review of Medical Image Segmentation Algorithms," *EAI Endorsed Trans. Pervasive Heal. Technol.*, vol. 7, no. 27, p. e6, Apr. 2021, doi: 10.4108/eai.12-4-2021.169184.

[13] M. A. Abdou, "Literature review: Efficient deep neural networks techniques for medical image analysis," *Neural Comput. Appl.*, vol. 34, no. 8, pp. 5791–5812, Jul. 2022, [Online]. Available: https://link.springer.com/article/10.1007/s00521-022-06960-9

[14] L. Abualigah, K. H. Almotairi, and M. A. Elaziz, "Multilevel thresholding image segmentation using meta-heuristic optimization algorithms: comparative analysis, open challenges and new trends," *Appl. Intell.*, vol. 53, no. 10, pp. 11654–11704, May 2023, doi: 10.1007/s10489-022-04064-4.

[15] H. Liang, H. Jia, Z. Xing, J. Ma, and X. Peng, "Modified Grasshopper Algorithm-Based Multilevel Thresholding for Color Image Segmentation," *IEEE Access*, vol. 7, pp. 11258–11295, 2019, doi: 10.1109/ACCESS.2019.2891673.

[16] F. Chakraborty, D. Nandi, and P. K. Roy, "Oppositional symbiotic organisms search optimization for multilevel thresholding of color image," *Appl. Soft Comput.*, vol. 82, p. 105577, Sep. 2019, doi: 10.1016/j.asoc.2019.105577.

[17] F. Chakraborty, P. K. Roy, and D. Nandi, "A novel chaotic symbiotic organisms search optimization in multilevel image segmentation," *Soft Comput.*, vol. 25, no. 10, pp. 6973–6998, May 2021, doi: 10.1007/s00500-021-05611-w.

[18] S. Singh *et al.*, "An efficient multi-level thresholding method for breast thermograms analysis based on an improved BWO algorithm," *BMC Med. Imaging*, vol. 24, no. 1, p. 191, Jul. 2024, doi: 10.1186/s12880-024-01361-x.

[19] S. Singh, N. Mittal, and H. Singh, "A multilevel thresholding algorithm using LebTLBO for image segmentation," *Neural Comput. Appl.*, vol. 32, no. 21, pp. 16681–16706, Nov. 2020, doi: 10.1007/s00521-020-04989-2.

[20] Q. Liu, Z. Jiang, and H. Shi, "Maximum Entropy Image Segmentation Method Based On Improved Firefly Algorithm," *J. Phys. Conf. Ser.*, vol. 1213, no. 3, p. 032023, Jun. 2019, doi: 10.1088/1742-6596/1213/3/032023.

[21] I. Brajevic and M. Tuba, "Cuckoo Search and Firefly Algorithm Applied to Multilevel Image Thresholding," in *Cuckoo Search and Firefly Algorithm: Theory and Applications*, Springer International Publishing, 2014, pp. 115–139. doi: 10.1007/978-3-319-02141-6_6.

[22] E. Turajlic, "Application of firefly and bat algorithms to multilevel thresholding of X-ray images," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2018, pp. 1104–1109. doi: 10.23919/MIPRO.2018.8400201.

[23] S. Liu and Y. Wang, "A Lévy Flight Based Firefly Algorithm for Multilevel Thresholding Image Segmentation," *J. Phys. Conf. Ser.*, vol. 1865, no. 4, p. 042098, Apr. 2021, doi: 10.1088/1742-6596/1865/4/042098.

[24] S. Singh, N. Mittal, and H. Singh, "A multilevel thresholding algorithm using HDAFA for image segmentation," *Soft Comput.*, vol. 25, no. 16, pp. 10677–10708, Aug. 2021, doi: 10.1007/s00500-021-05956-2.
[25] T. Rahkar Farshi and A. K. Ardabili, "A hybrid firefly and particle swarm optimization algorithm applied to multilevel image thresholding," *Multimed. Syst.*, vol. 27, no. 1, pp. 125–142, Feb. 2021, doi: 10.1007/s00530-020-00716-y.
[26] G. D. Singh, M. Prateek, S. Kumar, M. Verma, D. Singh, and H.-N. Lee, "Hybrid Genetic Firefly Algorithm-Based Routing Protocol for VANETs," *IEEE Access*, vol. 10, pp. 9142–9151, 2022, doi: 10.1109/ACCESS.2022.3142811.
[27] S. C. Satapathy, N. Sri Madhava Raja, V. Rajinikanth, A. S. Ashour, and N. Dey, "Multi-level image thresholding using Otsu and chaotic bat algorithm," *Neural Comput. Appl.*, vol. 29, no. 12, pp. 1285–1307, Jun. 2018, doi: 10.1007/s00521-016-2645-5.
[28] S. Pare, A. K. Bhandari, A. Kumar, and G. K. Singh, "A new technique for multilevel color image thresholding based on modified fuzzy entropy and Lévy flight firefly algorithm," *Comput. Electr. Eng.*, vol. 70, pp. 476–495, Aug. 2018, doi: 10.1016/j.compeleceng.2017.08.008.
[29] K. Sundaravadivu, C. Ramadevi, and R. Vishnupriya, "Design of Optimal Controller for Magnetic Levitation System Using Brownian Bat Algorithm," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2015*, Springer India, 2016, pp. 1321–1329. doi: 10.1007/978-81-322-2656-7_120.
[30] S. Suresh, S. Lal, C. S. Reddy, and M. S. Kiran, "A Novel Adaptive Cuckoo Search Algorithm for Contrast Enhancement of Satellite Images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 8, pp. 3665–3676, Aug. 2017, doi: 10.1109/JSTARS.2017.2699200.
[31] P. Anitha, S. Bindhiya, A. Abinaya, S. C. Satapathy, N. Dey, and V. Rajinikanth, "RGB image multi-thresholding based on Kapur's entropy — A study with heuristic algorithms," in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Feb. 2017, pp. 1–6. doi: 10.1109/ICECCT.2017.8117823.
[32] H. Wang *et al.*, "Firefly algorithm with neighborhood attraction," *Inf. Sci. (Ny).*, vol. 382–383, pp. 374–387, Mar. 2017, doi: 10.1016/j.ins.2016.12.024.
[33] R. Bhattacharya and E. C. Waymire, *Random Walk, Brownian Motion, and Martingales*, vol. 292. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-78939-8.
[34] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, L. Abualigah, M. Abd Elaziz, and D. Oliva, "EWOA-OPF: Effective Whale Optimization Algorithm to Solve Optimal Power Flow Problem," *Electronics*, vol. 10, no. 23, p. 2975, Nov. 2021, doi: 10.3390/electronics10232975.
[35] Y. Zhang, J. Lu, C. Zhao, Z. Li, and J. Yan, "Chaos Optimization Algorithms: A Survey," *Int. J. Bifurc. Chaos*, vol. 34, no. 16, p. 2450205, Dec. 2024, doi: 10.1142/S0218127424502055.
[36] L. He and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152–174, May 2017, doi: 10.1016/j.neucom.2017.02.040.
[37] D. Oliva, N. Ortega-Sánchez, S. Hinojosa, and M. Pérez-Cisneros, *Modern Metaheuristics in Image Processing*. Boca Raton: CRC Press, 2022. doi: 10.1201/9781003183501.