Improved Grey Wolf Optimizer based on Levy Flight for Multi-thresholding Image Segmentation

Ntaye Emmanuel
Kwame Nkrumah University of Science and Technology, Kumasi
Department of Computer Science

Michael Asante
Kwame Nkrumah University of Science and Technology, Kumasi
Department of Computer Science

Dennis Redeemer Korda
Bolgatanga Technical University Bolgatanga
Department of ICT

Emmanuel Oteng Dapaah
E.P College of Education, Bimbilla
Department of ICT

Dickson Kodzo Mawuli Hodowu
Ghana Water Company Limited
Department of Technology & Innovation

ABSTRACT
The Gray Wolf Optimizer is a relatively new and efficient population-based optimizer that seeks to speed up computations and find optimal solution for image segmentation problems. It is a metaheuristic algorithm that mimics the social hierarchy and hunting behaviour of the gray wolfs. However, because of the insufficient diversity wolves in some cases, it is still prone to stagnation at a local optimum. This may often happen when the GWO is not able to perform a smooth transaction from exploration to exploitation potential by more iteration. This paper proposed an improved gray wolf optimizer for Multilevel image segmentation based on levy flight (LGWO). Levy flight is an efficient strategy that increase the population diversity and prevents premature convergence by improving the ability to jump out of a local optimum. The performance of the LGWO is than evaluated and compared with two conventional population-based algorithms, the Particle Swarm Optimizer (PSO) and the Bat Algorithm (BA) by using the Kapur's entropy and Otsu's between-class variance function with ten standard gray scale images in a multi-threshold problem. The quality of the segmented images is compared using the maximum objective function, peak signal- to noise ratio (PSNR), CPU computation time and the optimal threshold value. The experimental results proved the LGWO algorithm an efficient and reliable algorithm in solving continuous image segmentation problems.

Keywords
Segmentation; Gray Wolf Optimizer; Optimization; Lévy Flight

1. INTRODUCTION
The adoption and implementation of nature- inspired algorithm for solving real world and continuous optimization problems has become an area of interest for most researchers in recent times. The foraging and leadership hierarchy for some social creatures like the birds, bees, bats, whales, wolves and ants have inspires the development of publication- based algorithms like the Bat Algorithm (BA) [1] [2], Firefly Algorithm [3] [4], Ant Colony Optimizer (ACO) [5] [6], Grey Wolf Optimizer (GWO) [7] [8], Particle Swarm Optimizer (PSO) [9], etc. Image segmentation is an important step in image processing and one of such area under which such algorithms are applied.

Image segmentation involves the breaking down of a large image into smaller, homogeneous fragments with identical density, color, and shape. It is one of the most fundamental procedures in image processing. When it comes to comprehending images and their representation, image segmentation is usually the initial step. High-level (HLL) applications such as feature extraction, picture recognition, semantic interpretation, and object categorization exploit segmentation's output [10] [11] [12] [13] [14].

After the image has been broken down into smaller fragments, it is imperative to find a better means of securing these fragments of data to minimize the loss and leakage of data, thereby improving the integrity of fragments passing through and increasing the level of trust of users [15] [16].

Thresholding techniques are very common in partitioning greyscale images due to their simplicity, accuracy, and robustness [17] [18]. Segmentation of images often simplifies splitting an image into pieces for use in certain applications. It is an important job that improves relevant analysis -and informative interpretation of the relevantly obtained image in various fields [19]. It is frequently used in character recognition [20] , automatic target detection [21], video change detection [22], medical imaging [23] [24] and similar [25] application areas. Many algorithms for image segmentation have been proposed in research studies over the past few decades. Algorithms for image segmentation broadly are put into four categories: thresholding, region growth, edge-based, as well as clustering.

Threshold evaluation presents an extremely important and effective function in the operations of image segmentation. There are two approaches to threshold an image, and this is largely dependent on the threshold values obtained out of the image's histogram. These are (a) bi-level thresholding [26] and (b) multi-level thresholding [27] [28].

Many thresholding methods over the years have been developed for image partitioning, such as the traditional techniques [29] and smart methods [30]. The histogram thresholding strategy over time has proven a simple yet effective approach. This technique does segmentation of the original image by choosing a threshold value within the gray-levels of the generated histogram of the original image. For the solution to the problem of thresholding, there are numerous thresholding strategies. Examples of these methods, called herbaceous criteria, selects the optimal or best threshold values by aiming for the grey level image's maximum variance value between the classes. Thresholding is a segmentation approach which works best with gray-level images. The concept is to search for a threshold, such that if a pixel is below it, it is regarded a background; if it is above it, it is assumed as a part of an object. Single-level and multi-level thresholding algorithms are two types of threshold-based algorithms. The multi-threshold method broadens the scope of thresholding by identifying numerous thresholds that try to separate different objects. In this thesis, a new grey wolf optimization algorithm grounded on Levy flight (LGWO) is proposed for the solution of the multilevel image thresholding problem and focuses on enhancing the speed and accuracy of the classic GWO. The GWO algorithm is simple to use and produces high-quality solutions. As a result, Otsu's between-class variance and Kapur's entropy function, were applied to the proposed LGWO algorithm to identify the multi-level thresholds. The study was carried out using MATLAB.

2. THE GREY WOLF OPTIMIZER

The grey wolf optimization algorithm, developed by [31]simulates grey wolf hunting and social behavior. Grey worms are divided into four social groups: alpha(α), beta(β), delta, and omega. Because the wolf group follows the Alpha group's rules, it is a dominant species. The beta class is made up of secondary wolves who assist the alpha in making decisions. The lowest-ranking grey wolves are represented by Omega. If a wolf does not belong to any of the above-mentioned species, it is referred to as a delta. Group hunting is an intriguing social behavior of grey wolves as well as the social interaction of wolves. The main elements of the GWO are the containment, hunting, and attacking of prey. For GWO, the hunting is primarily directed by alpha, beta, and delta.

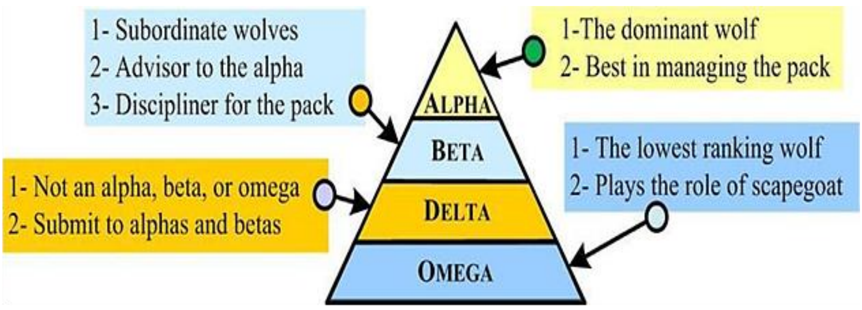


Figure 1: Social hierarchy of wolves and their characteristics in GWO [32].

2.1 Social hierarchy

Candidate solutions are arranged according to the wolf's social structure. Alpha, beta, delta, and omega, in that order, are the wolves with the greatest suitability levels.

2.2 Encircling prey

Equations 1 and 2 allow the grey wolf to update its position around the prey at random. The following is a diagram of grey wolf siege behaviour [31].

$$D = \left|C.X_p\,(t) - X(t)\right| \quad (1)$$

$$iX(t+1) = |X_p(t) - A.D| \quad (2)$$

The current iteration is represented by t, the coefficient vectors are represented by A and C, and the position vector of the pray is represented by X_p. X is a gray wolf's position. Equations 3 and 4 are used to calculate the values A and B, respectively [33].

$$A = |2a.r_1 - a| \quad (3)$$

$$C = |2a.r_2| \quad (4)$$

During iterations, the components of an are linearly reduced from 2 to 0. It's a [0, 1] random vector between r_1 and r_2. Worms can reach any place in the 2D and 3D space represented in Figure 2. and 3 using the random vectors r_1 and r_2.
The grey wolf, according to Equations (1) and (2), can reorganize its placement in the area surrounding the prey at any arbitrary location (2). Figure 2 and 3 depicts two-dimensional and three-dimensional space in the same way [34].

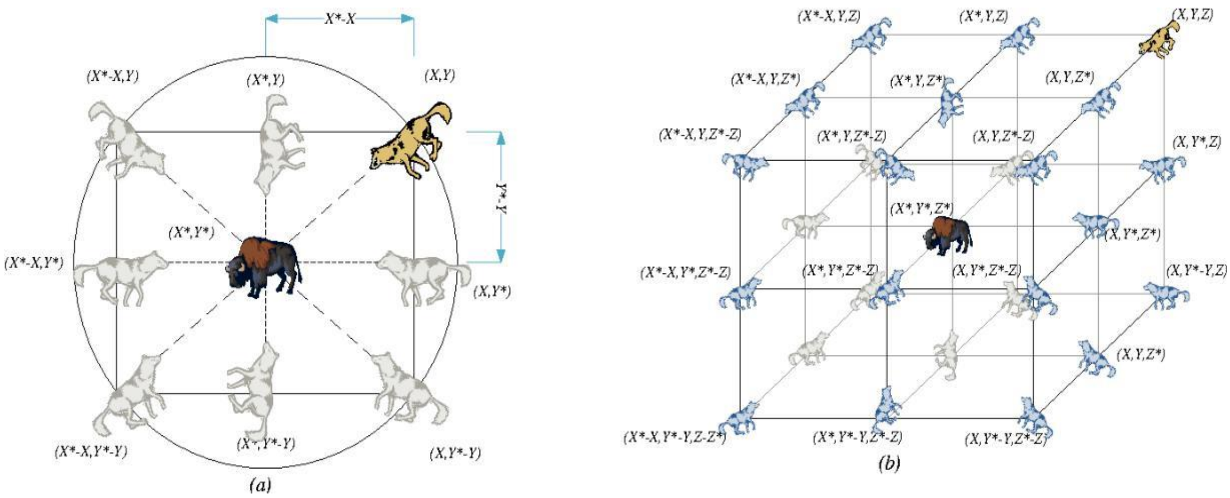


Figure 2: Position vectors and possible next positions of gray wolves in 2D and 3D space

2.3 Hunting

Grey wolves of the alpha, beta, and delta species have exceptional knowledge of their prey's current location. As a result, the top three best answers are saved, and additional wolves are free to update their locations in relation to the best search agents. In this case, equations 5-11 can be employed [35].

$$D_\alpha = |C_1.X_\alpha - X| \quad (5)$$

$$D_\beta = |C_2.X_\beta - X| \text{............ Eq.} \quad (6)$$

$$D_\delta = |C_3 X_\delta - X| \text{................ Eq.} \quad (7)$$

$$X_1 = |X_\alpha - A_1 D_\alpha| \text{...............Eq.} \quad (8)$$

$$X_2 = |X_\beta - A_2 D_\beta| \text{...............Eq.} \quad (9)$$

$$X_3 = |X_\delta - A_3 D_\delta| \text{...............Eq.} \quad (10)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3i} \text{.......... Eq.} \quad (11)$$

2.4 Attacking Prey

The value "a" is red at this point. The search agent's future position will be anywhere between the present position and the prey's position when a random value "A" in the range [-1 1] is used. The search agent's next position will be anywhere between the current position and the prey's position when A has random values in the range [-1 1], reducing A's range of change. The next position of the search agent will be anywhere between the present position and the position of the prey if A has random values in the range [-1 1].

2.5 Search for prey

Grey wolves are usually on the lookout for alpha, beta, and delta points. They are separated from one another in search of prey before reuniting in an assault. Parameter A with random values larger than or less than 1 is used to mathematically represent the distribution. This underlines the value of exploration and promotes the global search capability of the GWO algorithm.

Start Gray Wolf of Population $X_i = (1,2,3, \dots n)$
Assign a, A, C Parameter
Calculate eligibility of value of each agent
Find $X_\propto, X_\beta, X_\delta$
$X_\propto =$ *Agent with best position in the population*
$X_\beta =$ *Agent with 2nd best-position in the population.*
$X_\delta =$ *Agent with 3rd best-position in the population.*
While (t < maximum number of iterations)
  For each agent
    Update the location of existing search agent with Eq. (11).
  End for
  Update a, A, and C Parameter
  Calculate the eligibility value of each agent
  Update $X_\propto, X_\beta, X_\delta$
  t = t+1
end while
Return $X_\propto$

Figure 3: GWO algorithm

3. MULTI-LEVEL THRESHOLDING

Image segmentation is performed using the thresholding technique, which is based on the histogram of the given image. A method for segmenting a gray-level image into multiple separate sections is multilevel thresholding image segmentation. This technique separates an image into specified brightness zones that correspond to one background and several objects by determining multiple thresholds for the supplied image. For objects with colored or complicated backgrounds, where bi-level thresholding fails to yield adequate results, this method is ideal [36].

Each region of the image is given a separate threshold in local thresholding. In global thresholding single global threshold is derived from the whole image. To process an image, grey levels (L), the threshold (t) value *between* 0 and $L-1$, can be defined as in Equation 1 and 2 for two-level thresholding for an image. Where PF is the Pixel for the Foreground image,PB denotes Pixels for the Background image? I is the input image and t is the threshold valve [37].

$$PF\{(ix, y) \in I | 0 \le (ix, y) \le t - 1\} \quad (12)$$

$$PB = \{(ix, y) \in I | t \le (ix, yi) \le L - 1\} \quad (13)$$

By increasing the number of segments for thresholding, two-level thresholding can be converted to multi-level thresholding (Smith et al., 1979). The conversion is given in Equation 3.

$$P_0 = \{(x, y) \in I | 0 \le (x, y) \le t_0 - 1\} \quad (14)$$
$$P_1 = \{(x, y) \in I | t_0 \le I(x, y) \le t_0 - 1\} \quad (15)$$
$$\dots \dots \dots \dots \dots$$
$$\dots \dots \dots \dots \dots$$
$$P_n = \{(x, y) \in I | t_n - 1 \le \mathrm{I}\,(x, y) \le L - 1\} \quad (16)$$

4. OTSU THRESHOLDING METHOD

One of the most prominent methods given for image thresholding is the herbaceous approach, which is based on maximum of variance between classes. Otsu used variance between classes to establish the threshold value for two-level threshold valuation. The best t value for the two-level threshold value can be found when the total of the sigma functions assessed for all classes is maximized [38]. Mathematical modeling of the objective function is as follows in Equation 17 – 22.

$$t^* = argmax[f(t)] \quad (17)$$
$$f(t) = \sigma_0 + \sigma_1 \quad (18)$$
$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2, \sigma_1 = w_1(\mu_1 - \mu_T)^2 \quad (19)$$
$$\mu_0 = \frac{1}{w_0}\sum_{i=0}^{t-1} ip_i \ , \ \mu_0 = \frac{1}{w_0}\sum_{i=t}^{L-1} ip_i \quad (20)$$
$$\omega_0 = \frac{1}{w_0}\sum_{i=0}^{t-1} p_i \ , \ \omega_1 = \frac{1}{w_0}\sum_{i=t}^{L-1} p_i \quad (21)$$
$$P_i = \frac{x_i}{X} \quad (22)$$

Here x_i denotes total number pf pixels of intensity level, X stands for total number of pixels in the gray-scale image p_i as seen in Equation 9 shows the probability level at the grey level.w_0 *and* w_1 are the estimated probability of occurrence of segments 0 and 1 in Equation 8. μ_0 *and* u_1 represents the average density of classes 0 and 1 respectively as in Equation 7 and μ_T represents the average value of the image as in Equation 6, respectively. Finally, as shown in Equation 5,σ_0 is the variance of class 0 and σ_1 is the variance of class 1. Two-level image thresholding based on interclass variance is extended to multi-level thresholding as Equation 23 – 27 [38]

$$t^* = argmax[f(t)] \quad (23)$$

$$f(t) = \sigma_0 + \sigma_1 + \sigma_2 \ldots\ldots+\sigma_n \quad (24)$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2, \sigma_1 = w_1(\mu_1 - \mu_T)^2 \ldots.. \ \sigma_n = w_n(\mu_n - \mu_T)^2 \quad (25)$$

$$\mu_0 = \frac{1}{w_0}\sum_{i=t_0}^{t_1-1} ip_i \quad \mu_1 = \frac{1}{w_n}\sum_{i=t_1}^{t_1-1} ip_i \ldots. \ \mu_n = \frac{1}{w_n}\sum_{i=t_n}^{L-1} ip_i \quad (26)$$

$$\omega_0 = \frac{1}{w_0}\sum_{i=0}^{t_0-1} p_i \ , \ \omega_1 = \frac{1}{w_0}\sum_{i=t_0}^{t_1-1} p_i \ \ldots\ldots\omega_n = \frac{1}{w_0}\sum_{i=t_n}^{L-1} P_i \quad (27)$$

4.1 Kapur's Entropy Method

By maximizing the entropy of the segmented classes, Kapur's technique determines the best thresholds [39]. It makes advantage of Shannon's entropy idea. The following are the threshold criteria for this approach.

Let's say there are L grey levels in a given image, and these grey levels are in the range {0,1,2,3, ...(L-1)}

It can then be defined by

$p_i = \frac{h(i)}{N}, \big(0 \le i \le (L-1)\big)$ where by $h(i)$ indicates the number of pixels in the image which is equal to p_i the average threshold value

$$\sum_{i=0}^{L-1} h(i)$$

Then there's the goal of maximizing the fitness function.

$$f(t) = H_0 + H_1 \quad (28)$$

Where;

$$H_0 = -\sum_{i=0}^{t-1} \frac{P_i}{w_0}\ln\frac{P_i}{w_0}, \quad w_0 = \sum_{i=0}^{t-1} P_i$$

$$H_1 = -\sum_{i=0}^{L-1} \frac{P_i}{w_1}\ln\frac{P_i}{w_1}, \quad w_1 = \sum_{i=t}^{L-1} P_i$$

This method of Kapur's entropy criteria has also been extended to multilevel thresholding, as follows: For the generation of m ideal thresholds for a given image [t_1, t_2,....t_m], the optimal multilayer thresholding issue can be set as an m-dimensional optimization problem, with the goal of maximizing the objective function:

$$f([t_1, t_2 \ldots t_m]) = H_0 + H_1 + H_2 + \cdots H_m \quad (29)$$

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{P_i}{w_0}\ln\frac{P_i}{w_0}, \quad w_0 = \sum_{i=0}^{t_1-1} P_i \quad (30)$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1}\ln\frac{P_i}{w_1}, \quad w_1 = \sum_{i=t_1}^{t_2-1} P_i \quad (31)$$

$$H_2 = -\sum_{i=t_2}^{t_3-1} \frac{P_i}{w_1}\ln\frac{P_i}{w_2}, \quad w_2 = \sum_{i=t_2}^{t_3-1} P_i \quad (32)$$

....

....

$$H_m = -\sum_{i=t_m}^{L-1} \frac{P_i}{w_m}\ln\frac{P_i}{w_m}, \quad w_m = \sum_{i=t_m}^{L-1} P_i \quad (33)$$

$H_0, H_1, H_2, \ldots\ldots..H_m$ are the Kapur's entropies $\omega_0, \omega_1, \omega_2, \ldots\ldots..\omega_m$ are probabilities of the partitioned classes: $c_0, c_1, c_2, \ldots\ldots..c_m$ respectively [39].

5. THE PROPOSED ALGORITHM

5.1 Lévy flight

Lévy flight is a unique random walk model that adheres to the multiple powers law. Large steps done every now and then aid the algorithm's ability to conduct a worldwide search. Lévy flight is useful for achieving a better balance between algorithm exploration and exploitation, as well as avoiding local optimization. Many animals and insects in nature exhibit Lévy distribution in their foraging behaviour. The following formula can be used to express the Lévy distribution: [40]

$$Levy(\lambda) \sim u = t^{-\lambda} \qquad 1 < \lambda \le 3 \quad (34)$$

In mathematical calculations, the Mantegna algorithm is commonly used to replicate the Lévy distribution. The step length s can be represented as follows using the Mantegna algorithm: [40]

$$S = \frac{\mu}{|V|^{\frac{1}{B}}} \quad (35)$$

$$\mu = N(O, \sigma_\mu^2), v = N(O, \sigma_v^2)$$

With

$$\sigma_\mu = \left[\frac{\Gamma(1+\beta) \times sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right]^{\frac{1}{\beta}} \quad (36)$$

6. THE LGWO ALGORITHM

When compared to other well-known optimizers, the GWO method can provide efficient results. However, in other circumstances, the agents of GWO may face the possibility of stagnation in the local optimum due to insufficient wolf variety. This issue frequently arises when a traditional GWO is unable to make a smooth transition from exploration to exploitation potential through additional iteration. As a result, if the hunters are reassembled at a distance via Lévy flying, the algorithm will be optimized in a larger space, allowing it to escape the local optimization. The distributions of levy flights are Markovian stochastic processes with individual jumps distributed by the probability density function $\lambda(x)$ decaying at large x as $\lambda(x) \simeq |x|^{-1-\alpha}$ with $0 < \alpha < 2$ and by virtue of their variance divergence,$\langle x^2(t)\rangle \to \infty$, extremely long jumps may occur, and typical trajectories are self – similar, on all scales showing cluster of long jumps interspersed by long excursions. The LWGO relies on the advantage of the distributed excursion length, which optimize the search as compared to the tradition methods. As a result of this discovery, Lévy's flight path can assist GWO achieve a better equilibrium of exploration and exploitation.

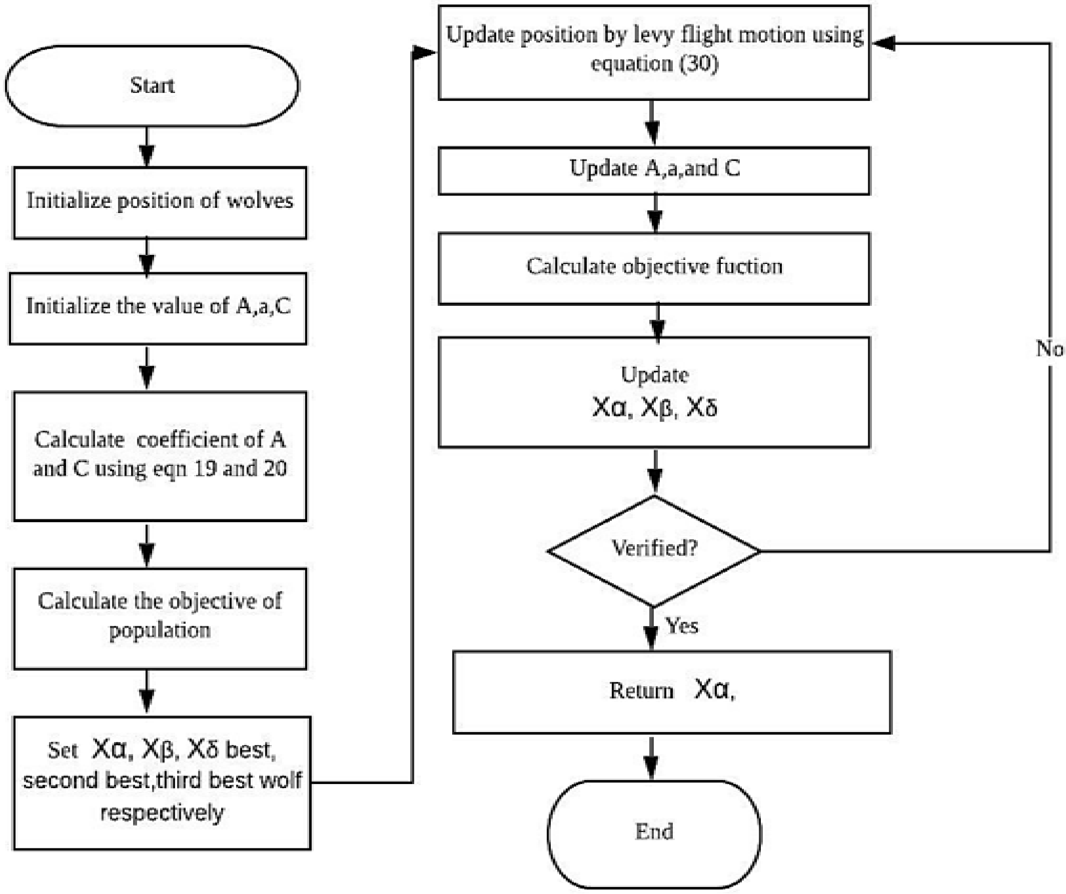


Figure 4: Flow chart of LGWO algorithm

Initialize the position of the gray wolf X_i= (1,2,3…n)
Initiate the value of a as 2
Calculate the coefficient of A and C using equation 3 and 4 respectively.
Calculate the objective value of each wolf by using Eq.23 for Otsu or Eq.29 for Kapur.
X∝ , Xβ, Xδ are the positions of α,β,& δ wolf
While (t < maximum number of iterations)
For each agent
Update the location of existing search
  agent with Eq. (30).
    End for
Decrease linearly the value of a from 2 to 0 during the iteration.
Update A and C using Eq. (3) or Eq. (4)
Calculate the objective function value of each wolf using Eq. (10) or Eq. (16)
Update X∝, Xβ, Xδ
t = t +1
end while
return X∝

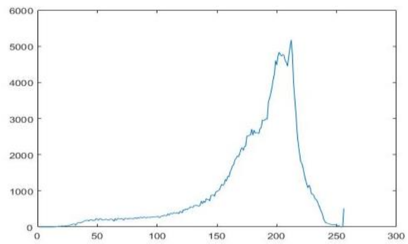
Figure 5. Pseudocode for proposed LGWO based on multilevel thresholding.

7. EXPERIMENTAL ENIVRONMENT

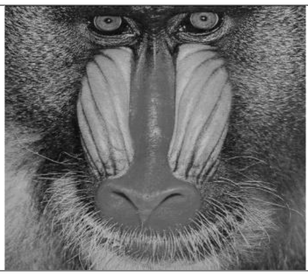
Ten standard benchmark gray-scale images of varying complexities were loaded and implemented in MATLAB and their histograms generated. The study selected thresholds of 2,4,6, and 8 since metaheuristic algorithms have stochastic properties and each segmented image was run 50 time for each threshold value.

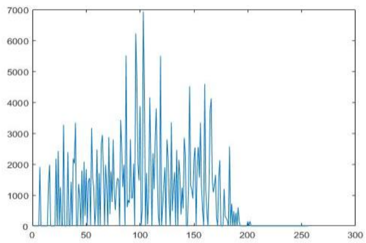
The average execution time of each algorithm running 50 times independently which reflect its computational complexity were calculated.

The Peak – to noise ratio (PSNR) of the segmented image and the original image is measured according to the intensity value in the image. The proposed LGWO was implemented with Otsu and Kapur methods using equation (10) for Otsu, or equation (16) for Kapur alongside the PSO and GA algorithms in MATLAB. The outputs of the Objectives function values of the various algorithm as shown in Table 3.

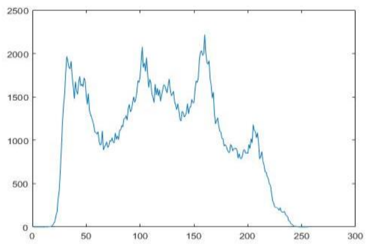


Aerial Image Histogram

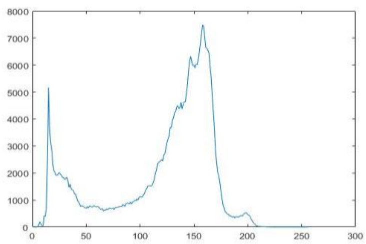




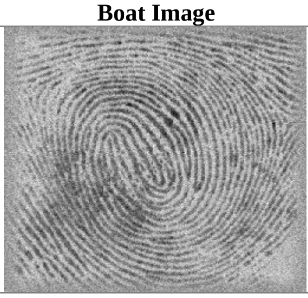
Baboon Image Histogram

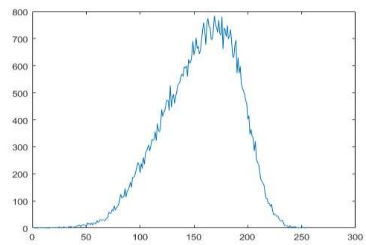


Barbara Image Histogram

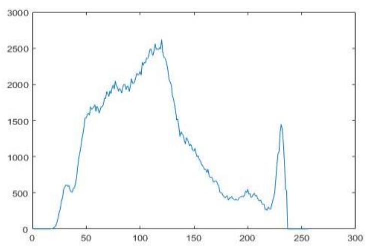


Boat Image Histogram



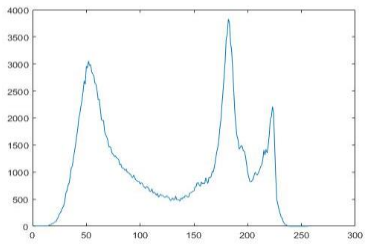


Finger Image Histogram



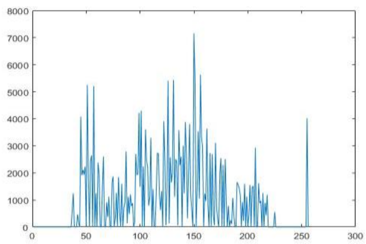
Goldhill Image Histogram

Lake Image



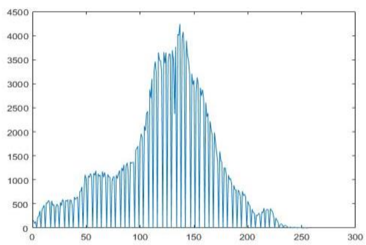
Histogram

Lenna Image



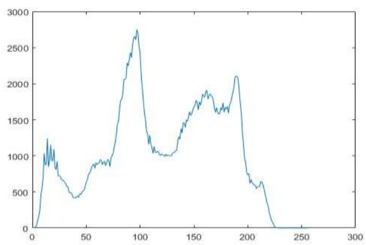
Histogram

Livingroom Image



Histogram

Pepper Image



Histogram

Figure 6: Images and generated Histogram

Table 1. The objective function values obtained by LGWO, GA, and PSO based methods

Test Image	**k**	**Otsu's Objective Function Values**			**Kapur's Objective Function Value**		
		LGWO	**PSO**	**GA**	**LGWO**	**PSO**	GA
Lenna	2	13.3770	13.3770	13.3770	11.4419	11.4419	11.4399
	4	**15.6973**	15.6672	16.4056	**18.0013**	18.008	16.5266
	6	**16.4710**	16.4056	15.5093	**20.6073**	20.6047	20.7579
	8	17.4905	**17.7305**	17.4905	**24.6818**	24.6669	24.1452
Barbara	2	12.8035	12.8035	12.8035	**12.6683**	**12.6683**	12.6683
	4	16.4606	16.4606	16.4606	**15.7470**	15.7470	15.7470
	6	**20.4118**	20.1100	20.1914	**18.5567**	18.5496	18.5567
	8	21.8569	22.1701	22.1250	21.2456	21.2418	21.2456
Livingroom	2	13.3453	13.3453	13.3453	**12.4059**	12.4057	12.4059
	4	**16.5694**	**16.5694**	16.5578	**15.5526**	15.5520	15.5526

	6	18.0689	**18.0902**	18.0353	**18.4710**	18.4673	18.4710
	8	20.5724	**21.5479**	21.2851	**21.1503**	21.1315	21.1494
Boats	2	9.31390	9.3139	9.3139	12.5747	12.5747	12.5747
	4	19.3022	19.3022	19.3022	**15.8209**	15.8206	**15.8209**
	6	20.7145	**20.9337**	20.7880	**18.6557**	18.6401	18.6557
	8	**22.9161**	22.8142	23.0469	**21.4016**	21.3920	21.4015
Goldhill	2	14.0412	14.0412	14.0412	12.5463	12.5463	12.5463
	4	14.9529	**17.3099**	14.7896	15.6077	15.6077	15.6077
	6	18.8108	**19.0786**	18.8069	**18.4142**	18.4141	18.4142
	8	**20.9066**	20.0750	19.9522	21.0991	**21.0990**	21.0991
Aerial	2	13.5605	13.5605	13.5605	12.5382	12.5382	12.5382
	4	**14.6375**	14.6210	14.6210	15.7518	15.7518	15.7518
	6	**15.9576**	15.2998	15.3414	18.6158	18.6158	18.6158
	8	**15.1537**	14.6671	14.8393	21.2104	**21.1923**	21.2104
Lake	2	12.6406	12.6406	12.6406	**12.5203**	12.5203	12.5203
	4	14.4537	14.4537	14.4537	**15.5662**	15.5662	15.5662
	6	14.4537	14.4537	14.4537	**18.3656**	18.3575	18.3656
	8	18.3708	**18.3708**	18.5456	**21.0249**	21.0159	21.0249
Finger	2	11.3617	**11.3617**	11.3617	11.8376	11.8376	11.8369
	4	**13.0476**	12.7972	12.7972	**17.7064**	17.7062	17.6864
	6	**14.9854**	14.0976	14.0976	**23.0571**	23.028	22.9108
	8	**16.5954**	15.2673	15.3430	**27.9943**	27.9748	27.57
Pepper	2	**11.6862**	**11.6862**	11.6862	**12.4352**	**12.4352**	12.4348
	4	18.9197	18.9197	18.9197	**18.1952**	18.1898	18.178
	6	**22.6946**	22.6124	22.6124	**23.4078**	23.4053	23.2467
	8	20.2524	**21.4951**	21.1431	27.9462	27.9358	27.5322
Baboon	2	11.7125	11.7125	11.7125	**11.2858**	11.2858	11.2837
	4	**12.7361**	12.7447	12.7447	**16.2875**	16.2836	16.2506
	6	19.6802	19.6802	19.6802	**20.6756**	20.6642	20.4461
	8	**19.2146**	19.0863	19.1213	**24.3934**	24.3363	23.8417

The LGWO algorithm obtained more successful results than PSO and GA methods under Kapur method. But under Otsu method for example when K = 8, the Barbara, living room, Lena, lake and pepper test data exceeded the value obtained by other algorithms and performed the best performance and was also successful in PSO methods in terms of stability for this test data. Pepper has achieved successful results in all cases except K = 8 for test data.

Good results were obtained for boat data, except for K = 6, similar to Living Room test data. When the results of Goldhill dataset were examined, the best values were obtained for K = 2, 4 and 6. In the case of K = 8, the LGWO algorithm obtained a result very close to the PSO and GA algorithms. For the Aerial and Finger test data, LGWO has shown that it is more stable than PSO and GA methods. Finally, the best results were obtained for Baboon data in case of K = 4 and K=8. In case of K = 4 and K = 6, LGWO algorithm was the most successful algorithm after PSO and GA algorithms in terms of solution quality.

In summary, the LGWO algorithm gave the best performance in all cases of K in Aerial and Finger test data, while the LGWO method reached the best result in the other test data except K = 8.

8. ANALYSES OF RESULTS

Table 2. Average CPU time of LGWO, GA, and PSO based methods at 50 runs each

Test images	***m***	**Kapur method**			**Otsu method**		
		LGWO	**PSO**	**GA**	**LGWO**	**PSO**	**GA**
Lena	2	0.4062	0.4099	**0.4475**	0.2694	0.3222	**0.2534**
	4	0.5236	0.6206	**0.6119**	0.2981	0.3412	**0.2884**

	6	0.6669	0.6995	**0.6384**	0.3417	0.3710	**0.3064**
	8	**0.7132**	0.7792	0.7558	**0.3416**	0.3940	0.3436
Barbara	2	0.4831	0.5132	**0.4559**	0.2871	0.3417	**0.2797**
	4	0.5898	0.6190	**0.5731**	0.3386	0.3647	**0.3417**
	6	0.6679	0.7283	**0.6535**	0.3471	0.3884	**0.3390**
	8	0.6624	0.6473	0.7581	**0.3663**	0.4261	0.3789
Living Room	2	0.4772	0.5209	**0.4702**	0.2718	0.3181	**0.2549**
	4	0.5914	0.6281	**0.5741**	0.3153	0.3333	**0.2898**
	6	0.6693	0.7307	0.6723	0.3382	0.3598	**0.3142**
	8	**0.7523**	0.7948	0.7762	0.34428	0.3821	**0.3348**
Boats	2	0.4857	0.5136	**0.4662**	0.2914	0.3257	**0.2917**
	4	0.5775	0.6276	**0.5766**	0.3308	0.3806	**0.3274**
	6	0.6841	0.7259	**0.6704**	0.3685	0.4011	**0.3490**
	8	0.7623	0.8151	**0.7564**	0.3912	0.4524	**0.3847**
Goldhill	2	0.4944	0.5349	**0.4464**	0.2883	0.2988	**0.2591**
	4	0.5662	0.6115	**0.5574**	0.3100	0.3362	**0.2907**
	6	0.6612	0.7264	**0.6597**	0.3232	0.3643	**0.3196**
	8	**0.6861**	0.8261	0.7651	**0.3548**	0.4078	0.3572
Aerial	2	0.4997	0.5138	**0.4433**	0.3357	0.3333	**0.2859**
	4	0.6021	0.6352	**0.5753**	0.3126	0.3540	**0.2949**
	6	**0.6775**	0.7332	0.6803	0.3539	0.3908	**0.3421**
	8	0.7824	0.8152	**0.7513**	0.37836	0.4722	**0.3601**
Finger	2	0.4943	0.5361	**0.4634**	**0.2821**	0.3209	0.3002
	4	0.5926	0.6310	**0.5756**	**0.3186**	0.3774	0.3250
	6	0.6916	0.7194	**0.6637**	0.3626	0.3985	**0.3411**
	8	**0.7586**	0.8204	0.7792	0.3884	0.4347	**0.3753**
Lake	2	0.4926	0.5493	**0.4855**	0.2655	0.3132	**0.2526**
	4	0.5848	0.6387	**0.5822**	0.2996	0.3388	**0.2847**
	6	0.7202	0.7455	**0.6579**	0.3284	0.3531	**0.3082**
	8	**0.7660**	0.8498	0.7974	0.3420	0.3817	**0.3368**
Pepper	2	0.4942	0.5250	**0.4598**	0.2666	0.3101	**0.2533**
	4	**0.5915**	0.6234	0.6520	0.3215	0.3311	**0.2823**

	6	0.6721	0.7224	**0.6511**	0.3295	0.3583	**0.3133**
	8	0.7577	0.7259	0.7807	0.3500	0.3887	**0.3395**
Baboon	2	0.4526	0.5069	**0.4406**	0.3050	0.3225	**0.2737**
	4	0.5434	0.6145	**0.5411**	0.3182	0.3523	**0.3046**
	6	0.6562	0.6690	**0.6249**	**0.3473**	0.4068	0.3584
	8	**0.7191**	0.7696	0.7368	0.39038	0.4122	**0.3759**

Since, the real-time applications need less running time in addition to high performance, CPU time of each algorithm has been examined. Corresponding results of average CPU time of 10 images is given in Table 2. As indicated in the tables, computation time increases significantly as the threshold level increases.

For example, in case of Barbara image with six thresholds, the average CPU time for Kapur based method are 0.6679, 0.7283, and 0.6535 ms for LGWO, PSO, and GA respectively.

Whereas, the average CPU time for Otsu based methods are 0.3471, 0.3884, and 0.3390 ms for LGWO, PSO, and GA respectively. It is also evident that computation of the proposed LGWO algorithm based on the Kapur's and Otsu's function is much faster (CPU time is less) than PSO but slower than GA.

Table 3. Comparison of PSNR values of the segmented images obtained by LGWO, GA, and PSO-based methods

Test images	*m*	PSNR values of Kapur methods			PSNR values of Otsu methods		
		LGWO	**PSO**	**GA**	**LGWO**	**PSO**	**GA**
Lena	2	11.4931	12.3455	12.3345	15.4015	15.0772	15.0406
	4	17.3660	17.8381	17.0892	18.3370	18.3052	17.9209
	6	**20.6047**	20.4423	19.5498	19.5987	18.7702	18.4021
	8	**23.5657**	22.1064	21.2161	**25.5814**	22.2378	21.2096
Barbara	2	14.4880	13.7415	10.4750	15.5678	13.6092	13.0807
	4	**19.1815**	18.3861	18.4133	**18.8922**	17.0105	17.1054
	6	20.7765	**21.2756**	20.1720	**21.2161**	18.0989	18.5493
	8	**23.0756**	22.7424	21.6211	**22.6354**	21.2356	21.1952
Living Room	2	14.5485	13.4626	12.2064	15.9646	15.4081	15.0371
	4	19.5368	20.1553	18.4506	**20.7602**	18.7631	18.8507
	6	**22.7822**	19.6461	21.211	**23.7635**	19.4643	19.2001
	8	**24.0625**	23.5699	23.4150	**25.3922**	23.5282	22.2937
Boats	2	14.5524	12.2599	11.9414	17.7083	17.0331	17.0487
	4	17.2370	**18.0003**	17.1668	**22.1064**	21.2548	20.5233
	6	**22.3093**	20.9631	19.7959	**24.0898**	22.0953	21.3690
	8	**23.3036**	22.9204	21.2116	**24.4695**	23.7114	22.8048
Goldhill	2	14.2565	12.3704	12.3490	13.9801	13.0927	13.8904
	4	18.7229	18.0408	17.2184	18.4097	17.0884	17.5087
	6	20.1748	**20.5335**	19.5637	**22.3424**	21.1283	20.8360
	8	**23.1110**	22.8703	22.2043	**23.8353**	22.0268	21.2843
Aerial	2	**15.0029**	14.6638	12.3435	16.0079	15.4801	15.5031
	4	**20.4054**	19.2787	17.9089	**20.4784**	18.4763	18.5067
	6	**22.6333**	21.2047	19.5549	**23.9793**	21.5033	21.2019
	8	**24.0242**	22.8007	22.6117	**25.6985**	23.2832	22.2537
Lake	2	**14.5119**	13.4715	12.7454	14.5233	13.9134	13.8790
	4	**17.4023**	16.725	14.877	**17.3621**	16.9362	17.2485
	6	**18.0693**	18.0051	17.9856	**20.9357**	19.8259	18.9061
	8	**23.8841**	21.9086	21.7256	**22.9204**	22.2063	21.3089
Finger	2	15.5491	11.4554	12.7345	**13.2510**	11.3618	10.4724
	4	**19.8675**	19.7868	18.3681	**18.4493**	17.9992	17.6218
	6	**22.1356**	23.5993	21.9256	**22.4943**	20.8533	20.6039
	8	**24.7923**	23.8999	22.8306	**26.1636**	25.6050	24.3475
Pepper	2	16.3651	14.6275	14.2877	16.3742	14.6863	13.5415
	4	**18.4206**	17.8924	17.8089	**20.0035**	18.9197	18.6381

	6	**21.3100**	20.8774	19.6549	**23.3439**	22.6124	21.4069
	8	**23.8841**	21.9086	21.7256	**22.9204**	22.2063	21.3089
Baboon	2	12.3554	12.2137	12.1846	16.4837	15.0886	15.3041
	4	**17.9143**	17.5741	16.9354	**20.5860**	19.2333	18.7086
	6	**20.5088**	20.2248	19.6625	**22.5091**	20.5268	20.2030
	8	**23.2398**	22.1356	22.9204	**25.3670**	23.9793	23.6402

The quality of the segmented images is evaluated by using PSNR. The difference between the segmented image and the reference image is measured according to the intensity value in the image.

The larger the PSNR value, the better the segmentation effect. The PSNR values of the segmented images obtained by all the methods are given in the Table **3**. PSNR gives a higher value when the segmented image is more similar to the original image. From the Table **3** it is found that PSNR values of the segmented images by LGWO based methods are higher than the GA and PSO based methods.

For example, the PSNR values in case of Lena image with eight thresholds for Kapur based methods are 23.5657, 22.1064, and 21.2161for LGWO, PSO, and GA respectively. It clearly shows that LGWO based method gives higher quality segmentation compared to GA and PSO based methods. It is also seen from Table **3** that, the value of PSNR index increases as the number of thresholds increase. This indicates that segmentation quality improves as the number of thresholds.

9. SUMMARY

The suggested LGWO-based multilevel thresholding technique's findings and analysis in terms of solution quality, stability, and computing time are presented in this part. The next subsections go through each of these points in detail. The CPU time of each approach has been investigated because real-time applications require low running time in addition to great performance. Table 4 shows the average CPU time of 10 photos and the corresponding results. The computation time increases dramatically as the threshold level increases, as shown in the tables. For example, the average CPU time for the Kapur-based technique on a Barbara picture with six thresholds is 0.6679, 0.7283, and 0.6535 ms for LGWO, PSO, and GA, respectively.

For LGWO, PSO, and GA, the average CPU times for Otsu-based algorithms are 0.3471, 0.3884, and 0.3390 ms, respectively. It is also clear that the suggested LGWO method, which is based on Kapur's and Otsu's functions, is substantially faster (in terms of CPU time) than PSO but slower than GA.

10. CONCLUSION AND FUTURE WORKS

This work proposes a modified version of GWO by incorporating levy flight for leading wolves in GWO to optimize the search ability for prey by wolf pack. Set of 10 standard benchmark images have been taken to check the robustness of the proposed LGWO algorithm. The performance of proposed algorithm is compared with PSO and BA algorithms that shows that LGWO is very competitive with the other algorithms.

From the analysis of the results done in this article it is recommended that LGWO outperforms PSO and GA in terms of Objective function value, PSNR as well as computational time. Also, LGWO provides a significantly better results compared to BA and PSO in the paper. This proposed algorithm is giving a new direction toward the improvement of leader’s search ability such that real word applications problems can be solved. Similarly other improvement for leading wolves can also be proposed to solve unconstrained optimization problems. Also, in future LGWO can be developed for solving different types of optimization problems like constrained optimization problems, integer programming problems etc.

11. REFERENCES

[1] X. Y. Yang, "A new metaheuristic bat-inspired Algorithm," Studies in Computational Intelligence,, vol. 284, pp. pp. 65–74, 2010, 2010.

[2] A. Alihodzic and M. Tuba, "Improved hybridized bat algorithm for global numerical optimization," in Proceedings of the 16th IEEE International Conference on Computer Modelling and Simulation, UKSim-AMSS '14, March,2014.

[3] X.-S. Yang, "Firefly algorithms for multimodal optimization,” in Stochastic Algorithms," Foundations and Applications, vol. 5792 of Lecture Notes in Computer Science, p. 169–178, 2009.

[4] I. Fister,, I. J. Fister , X. S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," Swarm and Evolutionary Compu tation, vol. 13, no. 1, p. 34–46, 2013.

[5] R. R. Jovanovic and M. Tuba, "Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem," Computer Science and Information Systems, vol. 10, no. 1, pp. 133 - 149, 2013.

[6] M. M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," Biosystems, vol. 43, no. 2, pp. 73 - 81, 1997.

[7] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," Advance Engineering Software, vol. 69, pp. 46 - 61, 2014.

[8] S. Mirjalili, "How effective is the Grey Wolf Optimizer in training multi-later perceptrons," Applied intelligence, vol. 43, no. 1, pp. 150 - 161, 2015.

[9] Eberhart, R; Kennedy, J;, A new optimizer using particle swarm theory, Proc.Sixth Int.Symp.Micro Mach.Hum.Sci, 1995, p. 39 43.

[10] Sathya, P D; Kayalvizhi, R;, Modified bacteria foraging for image segmentation, 2011.

[11] J. L. M. ´. ´. J. A. A. A. a. C. C. J. Lazaro, "Neuro semantic thresholding using OCR software for high precision OCR applications," Image and Vision Computing, vol. 28, no. 4, p. 71–578, 2010.

[12] C.-L. C. Y.-L. L. a. J. J. Y.-T. Hsiao, "Robust multiple objects tracking using image segmentation and trajectory estimation scheme in video frames," Image and Vision computing , vol. 24, no. 10, pp. 1123 - 1136, 2006..

[13] M. Y. M. H. R. a. N. H. H. R. Adollah, "Multilevel thresholding as a simple segmentation technique in acute leukemia images," Journal of Medical Imaging and Health

informatics, vol. 2, no. 3, pp. 285 - 288, 2012..

[14] G. C. Anagnostopoulos, "SVM-based target recognition from synthetic aperture radar images using target region outline descriptors," Nonlinear Analysis: Theory, Methods and Applications, vol. 71, no. 12, p. e2934–e2939, 2009..

[15] D. K. M. Hodowu, D. R. Korda and E. Ansong, "An Enhancement of Data Security in Cloud Computing with an Implementation of a Two-Level Cryptographic Technique, using AES and ECC Algorithm," International Journal of Engineering Research & Technology, vol. 09, no. 09, 2020.

[16] D. R. Korda, E. Ansong and D. K. M. Hodowu, "Securing Data in the Cloud using the SDC Algorithm," International Journal of Computer Applications, vol. 183, no. 25, pp. 24-29, 2021.

[17] P. K. Sahoo, S. Soltani and A. . K. Wong, " A survey of thresholding techniques," Copmuter Vision, Graphics, and Image Processing, vol. 41, no. 2, pp. 233-260, 1988.

[18] N. R. Pal and S. K. Pal, "Parttern Recognition," Expert Systems with Applications, vol. 26, no. 9, p. 1274–1294., 1993.

[19] S. Ouadfel and A. Taleb-ahmed, "Social spiders optimization and flower pollination algorithm for multilevel image thresholding : A performance study," Expert Systems With Applications, vol. 55, p. 566–584, 2016.

[20] J. L. M. J. A. A. A. a. C. C. J. Lázaro, "Neuro semantic thresholding using OCR software for high precision OCR application," Image and Vision Computing, vol. 28, no. 4, pp. 571 - 578, Apr. 2010.

[21] G. C. Anagnostopoulos, "SVM-based target recognition from synthetic aperture radar images using target region outline descriptors. Nonlinear Analysis, Theory, Methods and Applications," Nonlinear Analysis: Theory, Methods & Applications, vol. 71, no. 12, p. e2934–e2939, 2009a.

[22] C. L. C. Y. L. L. a. J. A. J. Y. T. Hsiao, "Robust multiple objects tracking using image segmentation and trajectory estimation scheme in video frames," Image and Vision Computing, vol. 24, no. 10, p. 1123–1136, Oct. 2006.

[23] M. Y. M. H. R. a. N. H. H. R. Adollah, "Multilevel Thresholding as a Simple Segmentation Technique in Acute Leukemia Images," Journal of Medical Imaging and Health Informatics, vol. 2, no. 3, p. 285–288, Sep 2012.

[24] A. &. N. A. K. Rojas Domínguez, "Detection of masses in mammograms via statistically based enhancement, multilevel-thresholding segmentation, and region selection," Computerized Medical Imaging and Graphics, vol. 32, no. 4, p. 304–315., 2008.

[25] A. A. a. M. Tuba, "Improved Bat Algorithm Applied to Multilevel Image Thresholding," Scientific World Journa, vol. 2014, pp. 1 -16, 2014.

[26] S. K. P. S. T. K. &. P. M. Kumar, "Bi-level thresholding using PSO, Artificial Bee Colony and MRLDE embedded with Otsu method," Memetic Computing, vol. 5, no. 4, p. 323–334, 2013.

[27] Y. z. Z. W. x. J. P. p. M. C. h. &. Y. J. j. Guo, "Multi-object extraction from topview group-housed pig images based on adaptive partitioning and multilevel thresholding segmentation," Biosystems Engineering, vol. 135, p. 54–60, 2015.

[28] S. K. A. B. V. &. S. G. K. Pare, "A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve," Applied Soft Computing Journa, vol. 1, no. 47, p. 76–102, 2016.

[29] N. R. P. a. S. K. Pa, "A review on image segmentation techniques," Pattern Recognition,, vol. 26, no. 9, p. 1277–1294, Sep.1993.

[30] E. C. a. H. S. V. Osuna-Enciso, "A comparison of nature inspired algorithms for multi-threshold image segmentation," Expert Systems with Applications, vol. 40, no. 4, pp. 1213 - 1219, Mar,2013.

[31] S. M. M. a. A. L. S. Mirjalili, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, p. 46–61, Mar. 2014.

[32] A. A. &. P. P. Heidari, " An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," Applied Soft Computing Journal, vol. 60, no. july, p. 115–134, 2017.

[33] C. Z. X. R. G. A. H. a. K. M. Yang X.-S., "Swarm Intelligence and Bio-Inspired Computation: Theory and Applications - .," Google Books, 2013.

[34] M. W. W. J. S. Z. L. F. G. S. S. &. X. W. Guo, " An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems," IEEE, vol. pp, 2020.

[35] H. &. C. S. Yang, "Incorporating a multi-criteria decision procedure into the combined dynamic programming/production simulation algorithm for generation expansion planning," IEEE Transactions on Power Systems, vol. 4, no. 1, pp. 165-175, 1989.

[36] S. S. R. S. F. &. K. H. R. Reddi, "Multilevel thresholding for image segmentation through a fast statistical recursive algorithm," ScienceDirect, vol. 4, p. 661–665, 1984.

[37] T. L. T. K. A. J. &. R. C. R. Da Silveira, "Automated drowsiness detection through wavelet packet analysis of a single EEG channel," Expert Systems with Applications, vol. 55, p. 559–565, 2016.

[38] N. Otsu, "A threshold selection method from gray level histograms," IEEE Transaction on Systems, vol. , pp. 62 - 66, 1979.

[39] J. N. S. P. K. &. W. A. K. C. Kapur, "A new method for gray-level picture thresholding using the entropy of the histogram," Computer Vision, Graphics, & Image Processing, vol. 29, no. 3, p. 273–285, 1985.

[40] C. Z. X. R. G. A. H. a. K. M. Yang X.-S., Swarm Intelligence and Bio-Inspired Computation: Theory and Applications - ., Google Books, 2013.