Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 1449 – 1457

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

RGB Histogram based Color Image Segmentation Using Firefly Algorithm

V. Rajinikanth[a,*], M. S. Couceiro[b]

[a]Department of EIE, St. Joseph's College of Engineering,Chennai 600 119, Tamil Nadu, India.
[b] Institute of Systems and Robotics, University of Coimbra, Ingeniarius, Lda.,Rua da Vacariça, n.37, 3050-381, Mealhada, Portugal.

Abstract

In this paper, optimal multi-level image segmentation is proposed using the Firefly Algorithm (FA). In this work, RGB histogram of the image is considered for bi-level and multi-level segmentation. Optimal thresholds for each colour component are attained by maximizing Otsu's between-class variance function. The proposed segmentation procedure is demonstrated using standard RGB dataset and validated using the existing FA in the literature combined with three randomization search strategies, such as Brownian Distribution, Lévy Flight and the Gaussian distribution related random variable. The performance assessment between FAs is carried out using parameters, such as objective value, PSNR, SSIM and CPU time.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: RGB histogram; Segmentation; Otsu; Firefly algorithm; PSNR; SSIM; CPU time.

1. Introduction

Image segmentation is an essential procedure, being extensively considered to extract meaningful information from grey scale or colour (RGB) images. During the segmentation process, a digital image is separated into multiple regions, or objects, in order to extract and interpret the relevant information. In recent years, this procedure has been widely considered in many key fields, such as remote sensing[3,4,5] medical imaging[16], and pattern recognition[9].

* Corresponding author. Tel.: +91 9380593801.
E-mail address :rajinikanthv@stjosephs.ac.in

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.064

Determining the exact threshold level to separate an image into desirable objects (foreground) from background remains an extremely significant step in imaging science.

In the literature, a considerable number of parametric and nonparametric bi-level and multi-level thresholding procedures have been proposed and implemented mainly for grey scale images[1,2,4,11]. Among them, global thresholding is considered as the most preferred image segmentation technique because of its simplicity, robustness, accuracy, and competence[14]. In general, existing parametric thresholding approaches are computationally costly, time consuming, and some times the performance degrades depending on the image quality[6,9]. Nonparametric traditional approaches, on the other hand, methods such as Otsu, Kapur, Tsai, and Kittler are simpler and successful for bi-level thresholding[11]. When the number of threshold level increases, the complexity of thethresholding problem also increases and the traditional method requires more computational time. Therefore, to overcome the computational complexity of most traditional methods, heuristic based bi-level and multi-level image thresholding procedures have been widely proposed by researchers for grey scale[1,2], RGB[10], multi-spectral and hyperspectral images[3,5]. Recent meta-heuristic algorithms, such as cuckoo search[1], bee colony[2], and firefly[14], are also employed to solve the m-level image thresholding problem. Most of the above discussed methods are applied and validated on a class of grey scaled images.

In recent years, the segmentation of RGB images, or more generally multi-spectral images, is also getting the attention of researchers. The authors from Ghamisi et al. proposed a heuristic-based segmentation technique for a class of hyperspectral colour images[3,5]. Su and Hu discussed a colour image quantization technique using self-adaptive differential evolution algorithm and the technique was validated using standard test images[10]. Sarkar and Das proposed a colour image segmentation procedure using Tsallis entropy and differential evolution. The authors validated the proposed method using a class of RGB images using 2D histogram technique[12].

In the proposed work, the RGB histogram of the colour image is considered to solve the *m*-level thresholding problem. The maximization of Otsu's between-class variance function is chosen as the objective function. The proposed segmentation procedure is a nonparametric approach, thus employing heuristic methods, such as Brownian search based Firefly Algorithm (BFA), Lévy Flight based Firefly Algorithm (LFA) and FA with Gaussian distribution related random variable (ε). The proposed method is implemented and validated on standard colour images.

2. Problem formulation

Otsu's based image thresholding was initially proposed back in 1979[8]. This method returns the optimal threshold of a given image by maximizing the between-class variance function. This procedure already proved its efficiency on grey scale [2,4,7,11,14] and colour images [3,5].

In this paper, Otsu's approach is considered for colour image segmentation with the aid of the RGB histogram. In RGB space, each colour pixel of the image is a mixture of Red, Green, and Blue (RGB) and for that same image, the data space size is $[0, L\text{-}1]^3$ (R = [0, L-1], G = [0, L-1], and B = [0, L-1]). In spite of this, one can formalize the heuristic based segmentation procedure as it follows[5].

For a given RGB image, let there be L intensity levels in the range [0,1,2,…, L-1]. Then, the probability distribution p_i^C can be defined as:

$$p_i^C = \frac{h_i^c}{N} \qquad \sum_{i=0}^{L-1} p_i^C = 1 \tag{1}$$

where i is a specific intensity level in the range $\{\, 0 \le i \le L-1 \,\}$ for the colour component C = {R,G,B}, $N$is the total number of pixels in the image, and h_i^C is the number of pixels for the corresponding intensity level I in component C.

The total mean of each component of the image is calculated as: $\mu_T^C = \sum_{i=0}^{L-1} i p_i^C = 1$ (2)

The$m$-level thresholding presents m-1 threshold levels t_j^C, where $j = 1,2,\ldots,m$-1, and the operation is performed as:

$$F^C(x,y) = \begin{cases} 0, & f^C(x,y) \le t_1^C \\ \frac{1}{2}(t_1^C + t_2^C), & t_1^C < f^C(x,y) \le t_2^C \\ \vdots & \vdots \\ \frac{1}{2}(t_{m-2}^C + t_{m-1}^C), & t_{m-2}^C < f^C(x,y) \le t_{m-1}^C \\ L-1, & f^C(x,y) > t_{m-1}^C \end{cases} \tag{3}$$

wherein x and y are the width (W) and height (H), in pixels, of the image of size H × W denoted by $f^C(x, y)$ with L intensity levels for each component.

The probabilities of occurrence w_j^C of classes $D_i^c, \ldots, D_m^c$ are given by:

$$w_j^C = \begin{cases} \sum_{i=0}^{t_j^C} p_i^C, & j=1 \\ \sum_{i=t_j^C-1+1}^{t_j^C} p_i^C, & 1<j<m \\ \sum_{i=t_j^C-1+1}^{L-1} p_i^C, & j=m \end{cases} \tag{4}$$

The mean of each class μ_j^C can then be calculated as:

$$:\mu_j^C = \begin{cases} \sum_{i=0}^{t_j^C} \frac{p_i^C}{w_j^C q}, & j=1 \\ \sum_{i=t_j^C-1+1}^{t_j^C} \frac{p_i^C}{w_j^C}, & 1<j<m \\ \sum_{i=t_j^C-1+1}^{L-1} \frac{p_i^C}{w_j^C}, & j=m \end{cases} \tag{5}$$

At last, Otsu's between-class variance of each component can be defined as:

$$\sigma_B^{c^2} = \sum_{j=1}^{m} w_j^C \left(\mu_j^C - \mu_T^C\right)^2 \tag{6}$$

where w_j^C is the probability of occurrence. The m-level thresholding is reduced to an optimization problem to search for t_j^C, that maximizes the objective function (J_{max}) of each image component C being defined as:

$$\varphi^C = \max_{1<t_i^C<,\cdots,L-1} \sigma_B^{c^2}(t_j^C) \quad \text{for C =\{R,G,B\}} \tag{7}$$

Solving this optimization problem for an RGB image may require a much larger computational effort for both bi-level and multi-level thresholds. Many methods have been proposed in the literature to solve the image thresholding problem[6,9,13]. Compared to traditional analytical techniques, heuristic-based segmentation techniques are used as alternatives due to their computational efficiency. Next section briefly describes some of these.

3. Brief overview of algorithms in the study

In this paper, the Firefly Algorithm (FA) and its recent improved forms are considered. The classical FA was initially proposed by Yang[19]. It is a nature-inspired meta-heuristic algorithm, in which flashing illumination patterns generated by invertebrates, such as glowworms and fireflies, were at the essence of its creation[15].

The traditional FA is developed by considering the following conditions[17,18,20]:

(i) Fireflies are unisex and one firefly will be attracted towards the nearest firefly regardless of its sex;

(ii) The attractiveness between two fireflies is proportional to the luminance;

(iii) The brightness of a firefly is somehow related with the analytical form of the fitness or cost function assigned to guide the search process. For instance, in a maximization problem, the luminance of a firefly is considered as to be directly proportional to the value of cost function (i.e., the luminance is the fitness function).

The movement of the attracted firefly *i* towards a brighter firefly *j* can be determined by the following position update equation:

$$X_i^{t+1} = X_i^t + \beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t) + randomization\ parameter \tag{8}$$

where X_i^{t+1} is the updated position of firefly, X_i^t is the initial position of firefly, and $\beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t)$ may be considered as the attractive force between fireflies.

The parameterizations of the algorithm, namely the necessary parameters to update the position of a firefly, have been discussed in the literature. In a recent paper from Raja et al.[15], the following three random parameters, such as Brownian search based FA (eq. 9), Lévy flight based FA (eq. 10), and the traditional FA, were considered to update the position of fireflies.

$$\alpha_1 . sign(rand - 1/2) \oplus B(s) \tag{9}$$

$$\alpha_1 . sign(rand - 1/2) \oplus L(s) \tag{10}$$

$$\alpha_1 . N_i (0,1) \tag{11}$$

where $L(s) = A.|s|^{1/\beta}$, $B(s) = A.|s|^{\alpha/2}$, $A = \beta\Gamma(\beta) sin\left(\frac{\beta\pi}{2}\right)\frac{1}{\pi}$. A is a random variable, β is the spatial exponent, α is the temporal exponent, and $\Gamma(\beta)$ is the Gamma function.

Initial firefly algorithm parameters are assigned based on the discussion presented by Raja et al.[15] which is summarized in Table 1.

Table.1 Initial parameters of heuristic algorithms

Parameter	Values
Number of Iterations	250
Population	20
Search dimension	m
Stopping criteria	J_{max}

4. Implementation

The grey level thresholding problem deals with finding the most favourable thresholds within the range [0, L-1] that maximize a fitness criterion. Similarly, considering the RGB histogram based technique, the heuristic algorithm finds the optimal thresholds within the data space of $[0, L-1]^3$ by maximizing Otsu's between-class variance function. The dimension of the segmentation problem mainly depends on the required threshold (*m*) levels. In this work, for the colour image segmentation problem, heuristic algorithms are allowed to explore $[[0, L-1]^3]^m$ data space in order to obtain the optimal threshold levels. Hence, RGB histogram based colour image segmentation is a

challenging work when compared to its grey level alternative.
The quality of the segmented image is assessed using well-known image metrics, such as the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Matrix (SSIM). Additionally, both fitness function value J_{max} and CPU time are considered.

The PSNR gives the similarity of the segmented image against the original image based on the Mean Square Error (MSE) of each pixel[11,14]:

$$PSNR(o,s) = 20 \log_{10} \left(\frac{255}{\sqrt{MSE(o,s)}} \right); \text{dB} \quad (12)$$

$$RMSE_{(o,s)} = \sqrt{MSE_{(x,y)}} = \sqrt{\frac{1}{MN} \sum_{i=1}^{H} \sum_{j=1}^{W} [o(i,j) - s(i,j)]^2} \quad (13)$$

where o and s are the original and segmented images of size H x W.
The SSIM is generally used to estimate the image superiority and inter-dependencies between the original and the processed image[2].

$$SSIM_{(o,s)} = \frac{(2\mu_O \mu_S + C_1)(2\sigma_{OS} + C_2)}{(\mu_O^2 + \mu_S^2 - C_1)(\sigma_o^2 + \sigma_s^2 + C_2)} \quad (14)$$

where μ_o and μ_s are the average of o and s, σ_o^2 and σ_s^2 are the variance of o and s, σ_{os} is the covariance of o and s, and $C_1 = (k_1 L)^2$ and $C_2 = (k_1 L)^2$ stabilize the division with weak denominator, with $L = 256$, $k_1 = 0.01$, and $k_2 = 0.03$.

5. Experimental results and discussions

The RGB histogram based image segmentation experiment is implemented in Matlab R2010a on an Intel Dual Core 1.6 GHz CPU, 1.5GB RAM running window XP. The implemented segmentation procedures are a revised form of the segmentation technique given at Matlab central webpage†. The proposed method is tested on standard RGB test images (481 X 321sized), such as Butterfly, Star fish, Rhino, Horse, Flower, and Train‡. The number of thresholds (m) considered in this procedure are 2, 3, 4 and 5. For each image, and for each m, the segmentation procedure is repeated 15 times and the mean value of the trials is chosen as the set of optimal thresholds and performance measures.

Initially the BFA, LFA, and conventional FA based optimization procedure is tested on the Butterfly image for m = 2-5. Fig. 1 (a - f) shows the original image, RGB histogram, segmented image and the corresponding optimal RGB threshold values. From Fig.1 (c - f), one can observe that, the RGB image segmentation is a much more complicated problem due to the three different colour patters, namely the Red (R), Green (G) and Blue (B) components. As previously stated, the histogram of a RGB image is more complex when compared to the histogram of grey scale image. Finding an optimal threshold on such complex histogram may be a challenging task. In other words, each colour distribution should be separately analysed considering the RGB histogram, which may increase the computational time. Fig. 2 shows the convergence of firefly algorithm based for m = 5. From this it is noted that all the algorithms provide approximately similar performance. From Table 2 and Fig. 2 one can observe that the convergence of LFA is better when compared with the alternatives considered in this study.

The above said procedure is repeated for other test images shown in Table 3. This table shows original 481 × 321 sized colour images, RGB histogram, and segmented bi-level and multi-level images with Brownian search FA (BFA). The performance measure values for these images, such as objective function, PSNR, SSIM, and the CPU time are presented in Table 2. The corresponding optimal thresholds (R, G, B) are presented in Table 4.

†http://www.mathworks.com/matlabcentral/fileexchange/authors/117313
‡http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html

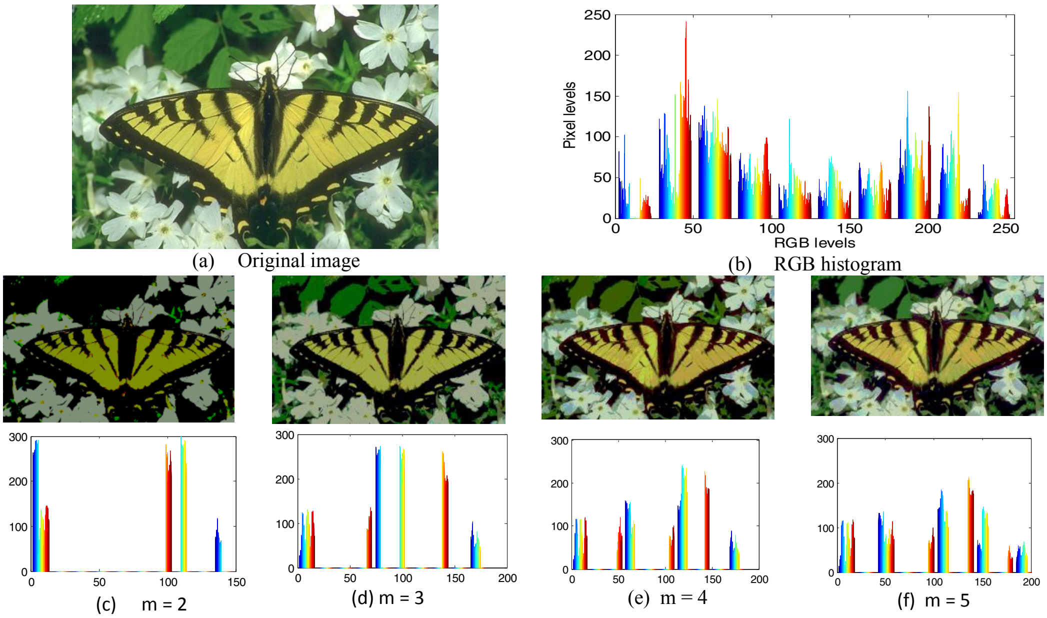
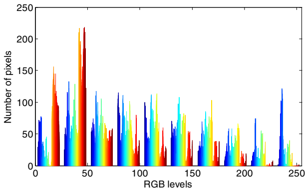
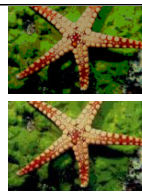
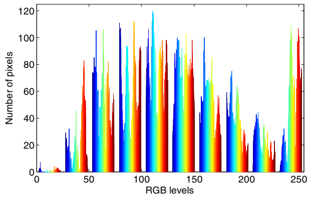
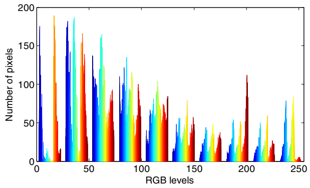
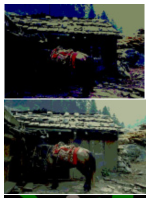
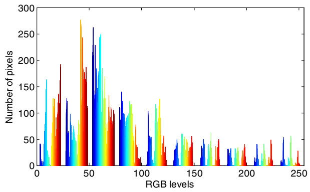
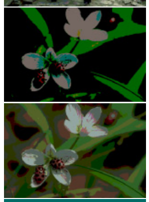
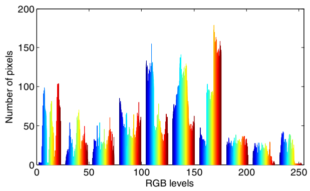


Fig.1. Segmentation of Butterfly image with BFA algorithm for m = 2-5

Table 2. Comparison of performance measure values for the RGB test images (mean value of 15 trials)

Image	m	Objective function			PSNR (dB)			SSIM			CPU time (min)		
		BFA	LFA	CFA	BFA	LFA	CFA	BFA	LFA	CFA	BFA	LFA	CFA
Butterfly	2	3515.92	3402.61	**3617.38**	10.866	10.243	**11.026**	0.6399	**0.6402**	0.6394	0.2281	0.2826	**0.2012**
	3	3629.37	3638.81	**3640.72**	14.297	**15.173**	14.927	**0.7133**	0.6936	0.7047	**0.4032**	0.4529	0.4173
	4	**3691.66**	3669.02	3690.81	17.562	17.283	**17.602**	**0.7835**	0.7669	0.7639	0.4967	0.4838	**0.4770**
	5	**3822.81**	3792.55	3811.01	19.554	19.328	**19.715**	**0.8472**	0.8317	0.8274	0.5503	**0.5138**	0.5259
Star fish	2	**1986.97**	1972.10	1985.11	11.513	**13.272**	12.267	0.7320	**0.7461**	0.7392	0.3899	0.2139	**0.2011**
	3	2017.18	2081.66	**2088.41**	14.868	14.792	**14.901**	**0.7831**	0.7706	0.7593	0.3901	**0.3103**	0.3348
	4	2107.25	**2109.91**	2098.77	**18.382**	18.281	18.332	0.8032	0.7996	**0.8106**	0.4825	**0.3628**	0.3915
	5	**2251.73**	2178.24	2201.62	19.191	20.037	**20.097**	**0.8529**	0.8274	0.8461	0.5765	0.5100	**0.47729**
Rhino	2	2004.99	2081.84	**2107.28**	9.881	**11.368**	11.206	0.6837	**0.7106**	0.7083	0.2925	0.26398	**0.2337**
	3	**2216.72**	2205.22	2192.77	13.463	**14.122**	13.974	0.7153	**0.7342**	0.7311	0.4107	**0.4092**	0.4099
	4	2251.33	**2267.18**	2222.90	16.182	16.001	**16.189**	0.7316	**0.7628**	0.7528	0.4829	0.4415	**0.43978**
	5	**2388.16**	2371.97	2382.28	**18.068**	17.926	17.874	0.7829	**0.8152**	0.7902	0.5719	0.5081	**0.4866**
Horse	2	2635.11	**2671.03**	2587.99	10.517	**12.015**	11.739	0.6402	0.7261	**0.7264**	0.2688	**0.2179**	0.2510
	3	**2688.04**	2683.31	2660.37	**14.701**	14.826	14.519	0.7026	**0.7418**	0.7302	0.3519	**0.28934**	0.3218
	4	**2717.37**	2716.03	2700.83	16.576	16.478	**17.005**	0.7792	**0.7902**	0.7886	0.4820	0.3826	**0.3775**
	5	**2782.70**	2763.44	2746.67	18.269	20.027	**20.157**	**0.8218**	0.8142	0.8213	0.5792	**0.4811**	0.5337
Flower	2	1159.57	**1302.61**	1288.92	11.284	**13.721**	12.826	0.6820	0.7227	**0.7301**	**0.3017**	0.3122	0.3108
	3	1420.23	**1472.71**	1392.44	**16.168**	14.916	16.026	0.7211	**0.7529**	0.7329	0.3725	**0.3518**	0.3597
	4	**1681.16**	1592.88	1562.39	**21.174**	20.177	20.291	0.7938	**0.8111**	0.8102	0.4826	0.4114	**0.4092**
	5	**1690.00**	1623.71	1607.35	20.844	**21.002**	20.926	0.8315	0.8268	**0.8331**	0.5639	**0.5297**	0.5442
Train	2	1829.01	1803.55	**1831.63**	12.648	12.579	**12.739**	0.6826	**0.6901**	0.6883	0.3721	0.3301	**0.3100**
	3	**1903.28**	1873.77	1894.00	14.282	14.138	**14.620**	0.6869	**0.7132**	0.7039	0.4028	**0.3877**	0.3891
	4	**1937.42**	1903.18	1917.22	**18.548**	18.207	18.442	0.7385	0.7835	**0.7893**	0.4927	0.4110	**0.4072**
	5	1975.56	1955.28	**1977.61**	20.031	20.379	**20.715**	0.8193	0.8352	**0.8374**	0.5783	0.5117	**0.4871**

Table 3. Test images, RGB histogram, and segmented images

Name	Original Image	Histogram	Segmented images m = 2 m = 4	 m = 3 m = 5
Star fish				
Rhino				
Horse				
Flower				
Train				

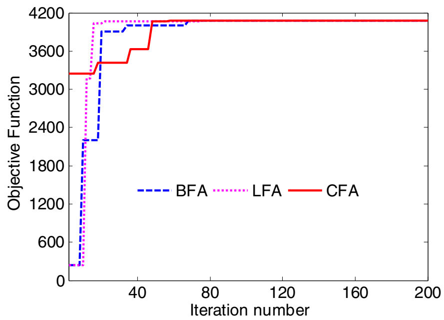


Fig. 2. Convergence of FA search

Table 4. Optimal threshold values obtained for the RGB images with firefly algorithms

Image	m	BFA			LFA			CFA		
		R	G	B	R	G	B	R	G	B
Butterfly	2	14,98	7,107	4,136	16,102	8,115	4,143	16,100	9,105	6,138
	3	13,69,144	6,96,174	3,82,167	15,71,149	7,93,171	3,85,164	15,68,147	8,98,177	4,80,168
	4	12,51,105, 146	6,71,124, 178	3,64,108, 172	13,49,107, 145	5,74,126, 181	3,61,110, 175	13,50,104, 144	6,72,123, 179	3,61,111, 175
	5	12,54,96, 133,167	5,52,114, 153,192	2,46,107, 140,179	11,56,99, 136,171	4,50,115, 151,195	2,41,109, 146,184	11,57,95, 135,168	4,54,116, 151,191	2,44,109, 148,184
Star fish	2	18,86	13,115	7,147	17,91	14,103	8,142	19,77	13,119	11,152
	3	16,78,124	10,89,162	7,69,175	15,81,127	12,75,160	7,61,181	17,60,127	11,81,167	8,77,174
	4	15,71,122, 138	8,80,131, 153	5,51,94, 194	12,68,120, 133	10,65,137, 158	4,47,90, 188	14,58,129, 128	9,77,134, 150	6,50,91, 198
	5	12,53,103, 135,164	5,33,107, 148,210	2, 49,92, 124,172	9,51,111, 141,162	7,30,101, 142,216	2, 45,84, 121,193	10,49,87, 133,166	7,31,112, 140,212	4, 51,91, 127,170
Rhino	2	17,95	14,134	12,155	17,91	15,137	13,152	16, 99	13,131	10,153
	3	15,61,137	13,79,174	10,43,158	14,66,134	13,82,171	11,41,165	15,58,132	11,71,167	9,47,10
	4	12,43,111, 152	11,68,139, 186	8,35,98, 204	12,39,116, 147	11,71,134, 182	8,38,102, 196	14,41,104, 147	9,66,133, 180	8,31,104, 201
	5	10,44,97, 145,177	7,28,107, 148,202	5, 39,105, 166,221	10,49,92, 147,170	8,31,101, 142,193	4, 33,111, 174,208	12,48,93, 140,172	7,33,112, 141,196	6, 31,102, 161,218
Horse	2	11,118	7,112	2,123	14,113	8,118	3,132	10,120	7,115	4,1131
	3	14,81,171	9,94,148	3,108,160	13,77,167	7,89,142	4,111,163	11,84,166	10,91,145	2,112,162
	4	16,67,125, 181	10,81,120, 157	4,78,114,1 70	12,66,121, 180	8,86,122, 164	6,71,112, 173	9,62,128,1 78	8,84,125, 163	2,71,104, 176
	5	18,54,95,1 38,196	12,71,112, 130,171	3,66,107,1 42,184	12,50,102, 145,193	10,76,117, 135,178	3,62,112, 144,187	13,44,99, 131,186	10,76,116 , 133,178	3,63,97, 152, 186
Flower	2	13,155	12,141	4,132	14,161	12,148	4,132	16,162	14,145	7,148
	3	15,106,16	10,76,152	6,130,178	13,111,16	11,73,150	6,130,178	14,112,172	11,68,160	6,126,167
	4	17,93,130, 188	9,85,127, 183	5,101,139, 176	11,73,128, 182	9,75,121, 180	5,101,139, 176	15,98,135, 181	10,82,128 , 177	5,95,127, 163
	5	14,90,115, 154,198	9,46,88, 124,189	4,98,135,1 61,177	15,91,121, 151,186	9,38,83, 128,184	4,98,135,1 61,177	14,91,112, 151,190	9,38,68, 119, 183	4,91,109, 158,170
Train	2	16,166	12,145	5,138	18,152	14,144	5,138	17,164	13,137	8,151
	3	15,122,18	11,101,17	4,115,169	16,124,18	12,122,17	4,115,169	16,120,176	11,97,166	6,111,143
	4	14,136,17 5,197	8,70,128, 188	5,102,148, 184	12,133,17 0,182	10,76,123, 180	5,102,148, 184	14,129,168 ,184	10,71,122 , 179	5,101,135, 177
	5	16,118,16 0,180,202	6,72,112, 175,196	4,85,127, 165,188	14,108,15 3,172,192	11,70,102, 176,189	4,85,127, 165,188	14,108,164 ,177,204	9,70,104, 163,183	4,65,97, 145,191

From these results, it is notable that despite small differences, all algorithms seem to reach the vicinities of the optimal solution. For all the tested images with various threshold levels, the convergence time of both LFA and FA seem better than BFA. On the other hand, the overall J_{max} (objective function) values obtained with the BFA are generally superior when compared to the alternatives.

6. **Conclusions**

In this paper, a new multi-level segmentation technique based on RGB histogram is proposed using Brownian search based Firefly Algorithm (BFA), Lévy search based Firefly Algorithm (LFA), and conventional Firefly Algorithm (FA). The proposed techniques are used to solve Otsu's problem for delineating multilevel threshold values. The segmentation procedure is validated using both qualitative and quantitative analysis, including traditional measures, such as objective function, PSNR, SSIM, and CPU time, which are evaluated by converting the segmented colour image into a grey scale image. Results demonstrate that the LFA and FA algorithms depict a faster convergence when compared to BFA, while the latter is able to achieve a superior final objective function.

References

1. Agrawal S, Panda R, Bhuyan S, Panigrahi BK. Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, *Swarm and Evolutionary Computation*, 2013; 11: 16–30.
2. Akay BA. Study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 2013; 13(6): 3066–3091.
3. Ghamisi P, Couceiro, MS, Benediktsson JA. Classification of hyperspectral images with binary fractional order Darwinian PSO and random forests. *SPIE Remote Sensing*, 2013; 88920S-88920S-8.
4. Ghamisi P, Couceiro MS, Benediktsson JA, Ferreira N M F. An efficient method for segmentation of images based on fractional calculus and natural selection. *Expert Syst. Appl*., 2012; 39 (16):12407– 12417.
5. Ghamisi P, Couceiro MS, Martins, FML, Benediktsson JA. Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization. *IEEE T. on Geoscience and Remote sensing*,2014; 52(5):2382-2394.
6. Lee SU, Chung SY, Park RHA. Comparative Performance Study Techniques for Segmentation, *Computer Vision, Graphics and Image Processing*, 1990; 52 (2) : 171 - 190.
7. Manikantan K, Arun BV, Yaradonic DKS. Optimal Multilevel Thresholds based on Tsallis Entropy Method using Golden Ratio Particle Swarm Optimization for Improved Image Segmentation, *Procedia Engineering*, 2012; 30 :364 – 371.
8. Otsu NA. Threshold selection method from Gray-Level Histograms, *IEEE T. on Systems, Man and Cybernetics*, 1979; 9 (1): 62-66.
9. Pal NR, Pal SK. A review on image segmentation techniques, *Pattern Recognition*, 1993; 26 (9): 1277 – 1294.
10. Su Q, Hu Z. Color Image Quantization Algorithm Based on Self-Adaptive Differential Evolution, *Computational Intelligence and Neuroscience*, Vol. 2013, Article ID 231916, 8 pages.
11. Rajinikanth V, Sri Madhava Raja N, Latha K. Optimal Multilevel Image Thresholding: An Analysis with PSO and BFO Algorithms. *Aust. J. Basic and Appl. Sci.*, 2014; 8(9): 443-454.
12. Sarkar S, Das S. Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy – A Differential Evolution Approach. *IEEE T. on Image Processing*, 2013.; 22(12): 4788-4797.
13. Sezgin M, Sankar B. Survey over Image Thresholding Techniques and Quantitative Performance Evaluation, *Journal of Electronic Imaging*, 2004; 13(1): 146 - 165.
14. Sri Madhava Raja, N.; Rajinikanth, V.; and Latha, K. Otsu Based Optimal Multilevel Image Thresholding Using Firefly Algorithm, *Modelling and Simulation in Engineering*, vol. 2014, Article ID 794574, 17 pages.
15. Sri Madhava Raja N, Suresh Manic K, RajinikanthV. Firefly Algorithm with Various Randomization Parameters: An Analysis, In B.K. Panigrahi et al. (Eds.): SEMCCO 2013, Part.1, *Lecture notes in computer science* (LNCS 8297), 2013;110–121.
16. Taher F, Werghi N, Al-Ahmad H, Donner C. Extraction and Segmentation of Sputum Cells for Lung Cancer Early Diagnosis. *Algorithms*, 2013; 6: 512-531.
17. Yang XS. Firefly algorithm, Lévy flights and global optimization, In Research and Development in Intelligent Systems XXVI, Springer-Verlag London, UK, 2010; 209–218.
18. Yang XS. Firefly algorithm, stochastic test functions and design optimization, *International Journal of Bio-inspired Computation*, 2010; 2(2): 78- 84.
19. Yang XS. Firefly algorithms for multimodal optimization, Stochastic Algorithms, Foundations and Applications, SAGA 2009, *Lecture Notes in Computer Sciences* 5792, 2009; 169–178.
20. Yang XS. *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, UK, 2nd edition, 2011.