An Optimal Fuzzy System for Edge Detection in Color Images using Bacterial Foraging Algorithm

Om Prakash Verma, *Member, IEEE,* Anil Singh Parihar

***Abstract*—This paper presents a fuzzy system for edge detection, using Smallest Univalue Segment Assimilating Nucleus (SUSAN) principal and Bacterial Foraging Algorithm (BFA). The proposed algorithm fuzzifies the Univalue Segment Assimilating Nucleus (USAN) area obtained from the original image, using a USAN area histogram based Gaussian membership function. A parametric fuzzy intensifier operator (FINT) is proposed to enhance the weak edge information, which results in another fuzzy set. The fuzzy measures: fuzzy edge quality factors and sharpness factor are defined on fuzzy sets. BFA is used to optimize the parameters involved in fuzzy membership function and FINT. The fuzzy edge map is obtained using optimized parameters. The adaptive thresholding is used to de-fuzzify the fuzzy edge map to obtain a binary edge map. The experimental results are analyzed qualitatively and quantitatively. The quantitative measures: Pratt's FOM, Cohen' Kappa, Shannon's Entropy and edge strength similarity based edge quality metric, are used. The quantitative results are statistically analyzed using t-test. The proposed algorithm outperforms many of the traditional and state-of-art edge detectors.**

***Index Terms*— Edge detection, Fuzzy intensifier, Fuzzifier, Edge sharpness, Fuzzy entropy, Bacterial Foraging, USAN area.**

I. Introduction

THE edge is characterized by change in intensity with some direction behavior. The edges represent all non-chromatic information contained within an image. In many applications such as segmentation, object detection, edge detection becomes the essential step. Even in case of noise removal, de-blurring, etc. the edge information plays a significant part. The edges can be divided into the two categories, i.e. strong or prominent edges and weak or fine edges. There are a number of methods available for edge detection, which are able to detect prominent edges. Most of the edge detectors available in the literature, are unable to detect fine or weak edges. In this work, we have proposed an optimal fuzzy system to detect strong and weak edges present within the image.

In literature, many edge detectors have been proposed. Canny [1] proposed an optimal edge detector, it smoothens image using Gaussian convolution and directional gradients to obtain edge information. Similarly Sobel, Prewitt, Roberts and LoG are also gradients based edge detectors [2]. Prewitt operator uses horizontal and vertical gradients, while Sobel uses the same with more weight to central pixel. Roberts uses a method of computing diagonal gradients, i.e. cross gradients to obtain edge information. The LoG edge detector uses Gaussian convolution and Laplacian to approximate the second derivative for edge detection. It has been noticed that, the edge detector involving Gaussian convolution may suffer from the problems like proper localization of edges, vanishing edges and false edges [3]. Salinas *et al.* [4] proposed regularization to obtain fused edge map and overcome "ill-posed" problem of previous edge detectors such as Canny. Perona *et al.* [5] proposed an anisotropic diffusion based algorithm, which produces the sharp region boundary to get better quality edges as compared to other Gaussian based methods. Molina *et al.* [6] presented an edge detector using finite sample images form Gaussian scale-space. The Sobel method is used to produce the edge map of each image. The final edge map is obtained by using a course-to-fine tracking algorithm.

The advancement in fuzzy theory, has contributed towards the growth of many fuzzy based edge detectors. The vagueness present in edge detection, i.e. location and visible clarity of the edges can be dealt with fuzzy reasoning. Bezdek *et al.* [7] used fuzzy logic to model geometric feature based functions for edge detection. The fuzzy reasoning is applied for edge in noisy images [8]. The morphological edge extraction method is integrated with local fuzzy properties and minimization of fuzzy entropy [9-10], are used for edge detection. Jinbo *et al.* [11] proposed a multilevel fuzzy edge detection algorithm. Yishu *et al.* [12] abstracted the image features using multi-scale wavelet and fuzzy C-mean clustering algorithm for edge detection. Bustince *et al.* [13] proposed an edge detector based on interval-valued fuzzy sets (interval type-2 fuzzy). The algorithm uses the difference of maximum and minimum in the neighborhood, to represent the uncertainty measure. It uses interval valued t-norm, t-conorm and entropy to detect the edges. In [14], Barrenechea *et al.* use the similar concept as in [13], to produce fuzzy edge images. The edges are obtained from fuzzy images using binarization and thinning. Molina *et al.* [15] presented an approach to generate fuzzy edge images by fuzzifing the gradient magnitude of image. The algorithm uses various parametric membership functions to capture the uncertainty in the edges. The fuzzy edge images are binarized to obtain edges. Most of the fuzzy based edge detectors uses de-fuzzification of the

Manuscript received on April 15, 2015; first revision on September 16, 2015; second revision on November 15, 2015, accepted on March 10, 2016.

Om Prakash Verma is with the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India (e-mail: opverma.dce@gmail.com).

Anil Singh Parihar is with the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India (e-mail: parihar.anil@gmail.com).

fuzzy edge map to obtain binary edges. Pinto *et al.* [16], proposed a thresholding algorithm based on Atanassov's intuitionistic fuzzy sets and Pagola *et al.* [17] presented a fuzzy thresholding algorithm.

Nature-inspired optimization algorithm has got attention from many researchers in the field of image processing. Passino [18] proposed an algorithm based on foraging behavior of bacteria, named bacterial foraging algorithm (BFA). The BFA has evolved significantly since then and find many applications in image enhancement and edge detection [19-21]. In [19], BFA is used to optimize various parameters for enhancement of the under exposed and over exposed region in the color images. Verma *et al.* [20] proposed a modification to the BFA algorithm for edge detection. The probabilistic derivative approach of ant colony optimization (ACO) [22], is used for determining the direction of movement of a bacterium. The edge map is obtained using the optimal path travelled by the bacteria. This approach results in some discontinuities in edges as well as the double edges. Verma *et al.* [21] proposed an edge detection algorithm in the noisy images using fuzzy derivative and BFA. In this algorithm, the movement of bacteria is governed by fuzzy derivate. This approach fails to detect the weak edges present within the image. Setayesh *et al.* [23] proposed an edge detection technique based on particle swarm optimization (PSO) [24]. The authors proposed two modifications of PSO to optimize the two sets of parameters in smooth region and edge pixels. Etemad and White [25] proposed an ACO inspired edge detection approach. In this approach, the authors have defined type-I pheromone deposited by ant movement and type-II pheromone determined by the Euclidean norm of the image gradient. The optimal path of ants is decided by these two pheromones. Sun *et al.* [26] proposed an edge detection technique based on the theory of universal gravity. In this approach, pixel intensity is considered as gravitational mass. Malina *et al.* [27] proposed another edge detector based on law of universal gravity. The algorithm calculates membership value of the edge fuzzy set using gravitational force between pixels. The effective detection of weak edges is still a challenging task. Most of the methods involve derivatives for edge detection, thus, suffers from the noisy and false edge map. A family of edge detectors uses Gaussian convolution, which results in dislocation of edges. The weak edges get vanished and false edges appear.

In the present work, we have developed an algorithm, which is capable of detecting weak edges along with strong edges. The proposed algorithm uses the USAN area [28] histogram based Gaussian membership function [29] to extract primitive edge information. This algorithm uses the integration of derivatives, which reduces the effect of noise. We propose a parametric fuzzy intensification operator (FINT) to enhance the weak edge information. Fuzzy measures are introduced to assess the quality of weak and strong edges. The sharpness factor and fuzzy entropy are used to define the objective function for optimization. BFA is used to get an optimized fuzzy edge map, which is further de-fuzzified to get binary edge map using adaptive thresholding. Following are the key contributions of our work:

1. A parametric fuzzy intensifier FINT to enhance weak edge information is proposed.
2. The fuzzy quality measures for strong and weak edges present in the image, are introduced. Further the sharpness factor is defined using fuzzy quality measures.
3. The effective utilization of BFA to obtain optimized fuzzy edge, which resulted in binary edge map after de-fuzzification.

The organization of the paper is as follows: Section II introduces the SUSAN principle to calculate the USAN area. Section II describes the fuzzification of USAN area matrix and FINT operation. In Section IV, the fuzzy measures are defined for sharpening the weak and the strong edges. The optimization using BFA is also described in this section. De-fuzzification process to obtain a binary edge map is described in section V. The experimental results and their analysis are presented in section VI. The conclusion is drawn in section VII.

II. Computation of USAN Area

The USAN (Univalue Segment Assimilating Nucleus) area is calculated using SUSAN (Smallest Univalue Segment Assimilating Nucleus) principle [28], with a mask of 37 pixels. The 37 pixel circular mask approximation in a digital image, is shown Fig. 1. The central pixel is compared with other pixels within the mask [28] using the following equation:

$$C(r,r_0)=\exp\left(-\left(\frac{I(r)-I(r_0)}{t}\right)^{\delta}\right) \quad where\ r\neq r_0 \qquad (1)$$

Where $I(\mathrm{r})$ is the intensity of a pixel at position *r*, within the mask, $I(\mathrm{r}_0)$ is the intensity of the nucleus of mask at position r_0, and t is brightness threshold. The parameter t is used to yield a proper representation for similar pixels in the neighborhood. The suitable value of parameter t is found experimentally to be 20 for 8-bit images. The value of δ is taken as '6' experimentally [21, 28] as it yields good results for nearly all types of images. The USAN area k at position r_0 is calculated as:

$$k(r_0)=\sum_r C(r,r_0) \qquad (2)$$

The value k is approximated to the nearest integer. The USAN area attains a maximum value in smooth region and its value decreases as edges appear within the mask. The variation of USAN area represents the edge information in the image. The USAN area involves integration of differences, therefore the noise effect is reduced. Applying the above process to the each pixel results in a USAN area matrix. The value of USAN area is high for flat regions and low for the edges, but USAN area alone is unable to detect the fine features within the image, especially weak edges. We apply the fuzzy theory to capture the fine features of the image.

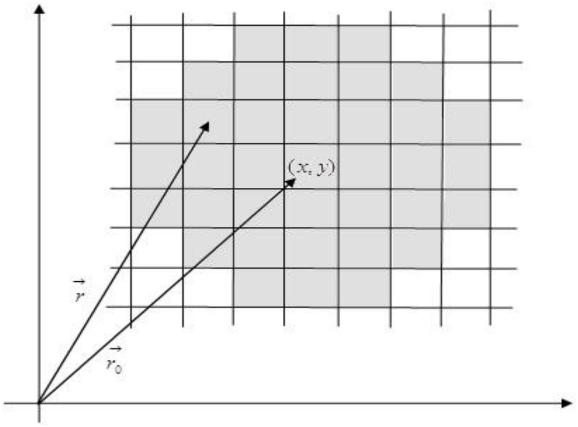


Fig. 1. Circular Mask approximation in digital images, with 37 pixels.

III. Fuzzification

We fuzzify USAN area matrix to obtain a fuzzy property [30], which reflects how much a pixel looks like an edge-pixel. We construct a histogram $h(k)$ of USAN area, which is the total number of pixels having USAN area k. The probability p of having USAN area k is given by:

$$p(k) = \frac{h(k)}{\sum h}; \qquad k = 0, 1,\ldots,36. \tag{3}$$

The USAN area histogram satisfies the following condition:

$$\sum_{k=0}^{L-1} p(k) = 1 \tag{4}$$

Where L is the total number of pixels in a mask i.e. 37. The USAN area matrix, obtained from spatial domain is transformed into fuzzy domain using histogram-based Gaussian membership function. This membership function for fuzzification of USAN area matrix is given as:

$$\mu_1(k) = \exp\left(-\left(\frac{(k-k_{\min})^2}{2f_h^2}\right)\right) \tag{5}$$

Where $k_{\min}$ is the minimum USAN area. The fuzzifier parameter f_h is determined as:

$$f_h^2 = \frac{1}{2}\frac{\sum_{k=0}^{L-1}(k-k_{\min})^4\, p(k)}{\sum_{k=0}^{L-1}(k-k_{\min})^2\, p(k)} \tag{6}$$

The minimum value of k *i.e.* $k_{\min}$ represents the strong-edge pixels as USAN area is minimum when strong edge is present. Thus, it gives a measure of deviation of the USAN area k from $k_{\min}$. The use of the ratio (normalized value of moments) rather than the standard variance of k gives robustness against deviations due do non-edge features and noise, but at the same time it may leave weak edges undetected. Since, $\mu_1(k)$ involves only one parameter f_h, it gives little control to the user. Therefore, to detect the weak edges present in the image, another fuzzy set is obtained using a parametric fuzzy intensification operator (FINT) on fuzzy set generated by $\mu_1(k)$. The membership function for intensifier with parameters α, β and γ is given by:

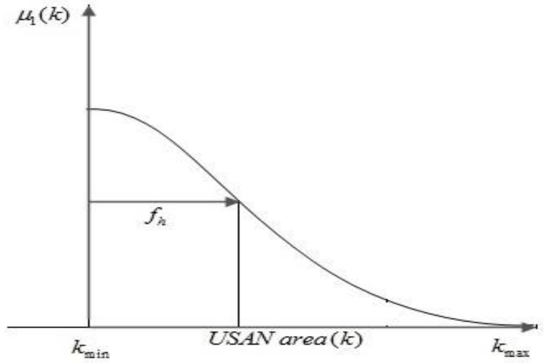


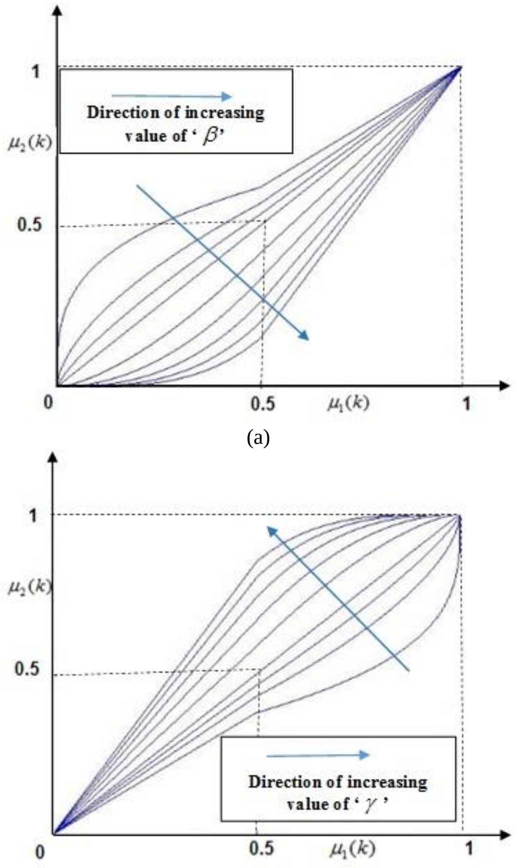
Fig. 2. Histogram-based Gaussian fuzzy membership function.

$$\mu_2(k) = \begin{cases} \alpha[\mu_1(k)]^{\beta}, & 0 \le \mu_1(k) < 0.5 \\ 1-\alpha[1-\mu_1(k)]^{\gamma} & 0.5 \le \mu_1(k) \le 1 \end{cases} \tag{7}$$

The membership function $\mu_2(k)$ needs to be continuous at $\mu_1(k) =$ 0.5, therefore parameters α, β and γ are related in following manner:

$$\alpha = \left[\, 2^{-\beta} + 2^{-\gamma} \,\right]^{-1} \tag{8}$$

Moreover, the parameters α, β and γ needs to be positive to keep the membership value $\mu_2(k)$ between 0 and 1. The strong edges will have a high membership value (in fuzzy set obtained by $\mu_1(k)$ with respect to weak edges as later have higher USAN area. The crossover point, where $\mu_1(k)$ takes the value 0.5 may be considered as a region of division for strong and weak edges. The crossover point will depend on the value of fuzzifier f_h. The application of fuzzy intensification operator (FINT) modifies membership values. The parameters α, β and γ control the shape of the membership function, hence provides more flexibility. The shape of membership function $\mu_2(k)$ becomes straight line for $\alpha = \beta = \gamma = 1$ i.e. no modification. Since, the parameters α, β and γ are related as per Eqn. (8), only two of the three parameters can be varied at a time. The variation in membership function $\mu_2(k)$, while keeping one of the parameters at a fixed value, are shown in Fig. 3. The variations in the value of these parameters alter the membership value of each pixel. The value of fuzzifier f_h, also plays an important role as it decides the crossover point of $\mu_1(k)$, therefore obtaining optimum values of the parameters α, β, γ and f_h becomes imperative.



(a)

(b)

Fig. 3. Fuzzy membership function intensification operator curves (a) with γ =1, (b) with β =1.

IV. Fuzzy Measures and Optimization

To optimize the parameters α , β , γ and f_h , evolutionary optimization technique, namely bacterial foraging algorithm is used in the present work. Other optimization algorithms such as particle swarm, gravitational search, artificial colony, etc. will also work with proposed algorithm. To facilitate the optimization process, the fuzzy measures are defined in the next sections. Further, the objective function for optimization is constructed using these fuzzy measures and fuzzy entropy.

A. Fuzzy Measures

The qualitative assessment of the edge map is required in the optimization process as it tries to achieve best possible edge quality. Assessing the quality of edge map and formulating an ideal measure for it, which can detect all the edges and their exact location, is always a challenge in the edge detection literature. Nearly, all the existing measures leaves some vagueness in one or other situations. The fuzzy theory capture this vagueness; therefore, a fuzzy measure for qualitative assessment becomes instrumental. In the present approach, fuzzy measures are defined, which estimates the edge map quality. The fuzzy edge sharpness measures are defined for both fuzzy sets, i.e. fuzzy sets obtained by $\mu_1(k)$ and $\mu_2(k)$ to measure edge quality of strong and weak edges. The fuzzy edge sharpness factor for the weak edges of the image is given by:

$$F_w = \sum_{k=0}^{L-1} [\mu_2(k) - c]^2 \, p(k) \tag{9}$$

Where c is the crossover point of the membership function $\mu_1(k)$. The fuzzy edge sharpness factor indeed depicts, how far the membership values of $\mu_1(k)$ are stretched with respect to the crossover point c after applying the fuzzy intensification operator FINT. Therefore, the desired value of F_w should be obtained by tuning the parameters α , β , γ , and f_h . The average fuzzy edge sharpness factor for the weak edges of the image is given by:

$$F_{avgW} = \sum_{k=0}^{L-1} [\mu_2(k) - c] p(k) \tag{10}$$

Similarly, the fuzzy edge sharpness factor for the strong edges of the image is given by:

$$F_S = \sum_{k=0}^{L-1} [\mu_1(k) - c]^2 \, p(k) \tag{11}$$

The average fuzzy edge sharpness factor for the strong edges of the image is given by:

$$F_{avgS} = \sum_{k=0}^{L-1} [\mu_1(k) - c] p(k) \tag{12}$$

The fuzzy sharpness factor gives a measure of deviation from the crossover point of fuzzy set and average fuzzy edge sharpness factor gives the overall sharpness of the edges with in the image. The ratio of the absolute average fuzzy edge sharpness factor to the fuzzy edge sharpness factor is defined as the fuzzy edge quality factor. The fuzzy edge quality factor for the weak edges of the image is given by:

$$Q_W = \frac{F_{avgW}}{F_W} \tag{13}$$

The fuzzy edge quality factor for the strong edges of the image is given by:

$$Q_S = \frac{F_{avgS}}{F_S} \tag{14}$$

It can be observed from above definitions neither very low nor very high value of the quality factor represents a good quality edge map. A very low fuzzy edge quality factor results due to very few edges, i.e. poor edge detection. Similarly, a high fuzzy edge quality factor, i.e. membership values have a large deviation from the critical point. Therefore, a high value of the quality factor is due to a large number of false edges in the edge map. Thus the range of values of the fuzzy quality factors for good quality of the edge map exists in between two extreme values. In the present approach, the ratio of these factors is termed as the sharpness factor (section IV.B). The above definitions are coined to capture the extent of the uncertainty present in the edges. These fuzzy measures are used for the computation of the objective function for optimization. The quality of the edges is ascertained by a quantitative analysis based on the fuzzy edge sharpness measures and the fuzzy entropy (section IV.C).

B. Sharpness Factor

The normalized fuzzy edge sharpness factor, called the sharpness factor, is defined to evaluate the fuzzy edge sharpness. The sharpness factor is defined as:

$$S_f = \frac{Q_S}{Q_W} \tag{15}$$

The sharpness factor cannot be used directly as objective function in the optimization process as it will result in parameters (α, β, γ and f_h) pertaining minimum or maximum value of S_f. A very low (high) value of S_f may be due to low (high) value of Q_S and high (low) value of Q_W, but both are undesirable. It is experimentally found that the sharpness factor should lie in between 1.0 and 1.5 for a good quality edge map. The desired value of the desired sharpness factor is denoted as S_{df}.

C. Fuzzy Entropy

In the process of fuzzification, the spatial domain edge information is transformed in the fuzzy domain using USAN area. In the fuzzy domain, the measure of uncertainty or randomness associated with the image is characterized by fuzzy entropy. The fuzzy entropy [31-32] of fuzzy set obtained by $\mu_2(k)$ is defined using Shannon's function as:

$$E = \frac{-1}{L \ln 2} \sum_{k=0}^{L-1} [\mu_2(k) \ln(\mu_2(k)) + (1-\mu_2(k)) \ln(1-\mu_2(k))] \tag{16}$$

The fuzzy entropy of an image characterizes the edge information present in the image. The optimization of fuzzy entropy should result in the desired value of the parameters α, β, γ and f_h involve in the membership functions.

D. Objective Function for Optimization

The optimization is performed to obtain suitable values of parameters α, β, γ and f_h such that both strong and weak edges are detected properly. The entropy function and the edge sharpness factor S_f both express the edge quality, hence these must be optimized together. The minimization of fuzzy entropy would reduce the randomness in edge map, but a very low value of fuzzy entropy would result in loss of some actual edges, which is undesirable. Therefore, optimize the entropy function E subject to the constraint $S_{df} = S_f$. An objective function is defined as:

$$J = E + \sigma \left| S_{df} - S_f \right| \tag{17}$$

Where $\sigma =$ 0.5 is chosen experimentally. The objective function is optimized for four parameters α, β, γ and f_h i.e. four dimensional search space has to be explored for optimum value. The search space needs to satisfy the condition in Eqn. (8) and α, β, γ takes only positive values. The initial value of fuzzier is calculated from image information using Eqn. (6), which controls the shape of the membership function $\mu_1(k)$.

E. Bacterial Foraging algorithm

The bacterial foraging algorithm, introduced by Passino [18] in the year 2002, is one of the nature-inspired evolutionary optimization algorithms, which caught great attention in recent times. The bacterial foraging algorithm is inspired from the foraging strategy of the *E. coli* bacterium. The *E. coli* is made up of plasma cell and contains "cytoplasm" and "nucleoid". At a given temperature and sufficient food bacterium gets longer and divided into daughters (reproduction). In search of food the motion patterns (taxes) of bacteria, govern by chemical attractants (nutrient region) and repellants (noxious region), are called chemotaxis. The *E. coli* bacterium has a control system, which facilitates the search of food and avoids the noxious region by changing the direction of movement. The foraging behavior of the bacteria is mimicked into a mathematical model in the BFA. The major steps of BFA are: Chemotaxis, Swarming, Reproduction, Elimination and Dispersion. An objective function $J(i, j, k, l)$ is defined to represent the fitness (cost) of the i^{th} bacterium at the position $\theta^i(j,k,l)$ in j^{th} chemotaxis, k^{th} reproduction and l^{th} elimination-dispersion step. The positions of bacteria may be initialized randomly or by using some given information.

Chemotaxis*:* The bacteria search for food, i.e. nutrient rich region and movement of bacteria is governed by nutrient concentration in the search space. The nutrient concentration in search region is represented as a cost function $J(\theta)$ at any position θ. The bacterium moves to attain a better nutrient region, i.e. to reduce cast value $J(\theta)$. Let the step size of movement in random direction is $\lambda(i)$ for the i^{th} bacterium. If direction of movement takes the bacterium in a better nutrient region, it continues to move in the same direction as long as cost function reduces or maximum number of steps N_s reaches. If movement of the bacterium takes in the region of constant nutrients or noxious region, the next step is taken in some random direction, termed as tumble. The random vector $\Delta(i)$ represents the direction of movement, and the movement of a bacterium is given by:

$$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + \lambda(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{18}$$

Swarming*:* The bacteria use a cell to cell signaling to swarm together. It signals to attract other bacteria in a nutrient rich region. The bacterium also repels the nearby bacterium, to avoid over consumption of its food by others. These factors are incorporated in BFA by using attractant and repellent parameters.

Reproduction*:* The health of a bacterium is measured as the value of the objective function at a position. The bacteria are sorted with respect to health, and the healthier bacteria are split into two, in the same position. To make the total number of bacteria same, half of the bacteria are reproduced, and the other half gets eliminated.

Elimination and dispersal*:* This step redistributes the bacteria population in such a way that bacteria in low nutrient or noxious regions are eliminated and dispersed randomly to explore the regions which might be remain unexplored. The number of eliminations and dispersals kept same in number to

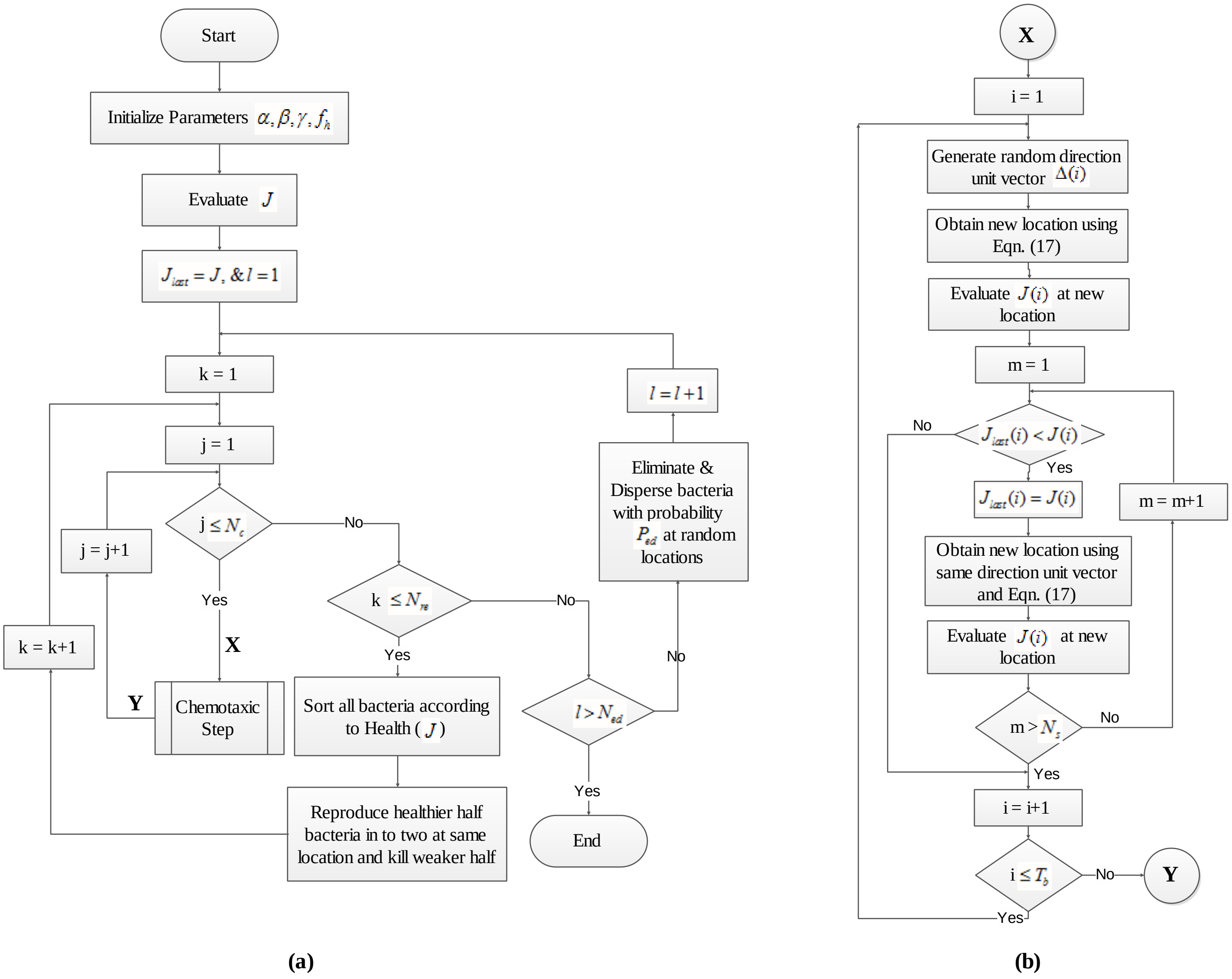

Start
Initialize Parameters $\alpha, \beta, \gamma, f_h$
Evaluate J
$J_{last} = J$, & $l = 1$
k = 1
j = 1
$j \le N_c$
No
Yes
X
j = j+1
k = k+1
Y
Chemotaxic Step
$k \le N_{re}$
No
Yes
Sort all bacteria according to Health (J)
Reproduce healthier half bacteria in to two at same location and kill weaker half
$l > N_{ed}$
No
Yes
End
$l = l + 1$
Eliminate & Disperse bacteria with probability P_{ed} at random locations
(a)
X
i = 1
Generate random direction unit vector $\Delta(i)$
Obtain new location using Eqn. (17)
Evaluate $J(i)$ at new location
m = 1
No
$J_{last}(i) < J(i)$
Yes
$J_{last}(i) = J(i)$
m = m+1
Obtain new location using same direction unit vector and Eqn. (17)
Evaluate $J(i)$ at new location
$m > N_s$
No
Yes
i = i+1
$i \le T_b$
No
Y
Yes
(b)

Fig. 4. (a) Flow chart of bacterial foraging algorithm, (b) Flow chart of Chemotaxis step.

make total bacteria count same.

F. Initialization of position of bacteria and the parameters of the BFA

The The bacteria are placed randomly in four dimensional search space created by α , β , γ and f_h i.e. position of i^{th} bacterium θ^i is defined by a quartet (α , β , γ , f_h). The initial positions of all bacteria are taken randomly by taking random values of α , β , γ . Although, the values of parameters α , β , γ are always positive and should follow the relation given by Eqn. (8). The value of f_h is calculated by Eqn. (6), i.e. initially all bacteria are placed in positions having the same value of f_h . The initial values of BFA parameters are set as follows: Number of bacteria T_b =15, Swimming length N_s =5, Number of iterations in a chemotactic loop N_c =10, Number of reproduction steps N_{re} =6, Number of elimination and dispersal events N_{ed} = 4, probability of elimination/dispersal p_{ed} =0.26. Although, the initial position of bacterium has no effect in optimize value of parameters, but a wise selection of positions of bacteria can reduce execution time significantly. The number of bacteria involves another tradeoff between edge quality and complexity. A wide range of values is tried on different images and it is concluded that it needs to keep the optimum number of iterations to execution time reasonable, but at the same time not sacrificing edge map quality. Although, initial values of parameters α , β , γ and f_h does not affect the quality of edge map obtained as these are optimized through BFA.

V. De-fuzzification: Adaptive Thresholding

The fuzzification, using optimized parameter results in the edge map with fuzzy membership, termed as the fuzzy edge map. To obtain a well define an edge map, i.e. a classical set having pixels that truly belongs to edges only. Therefore, we need to de-fuzzify the fuzzy set to a classical set.

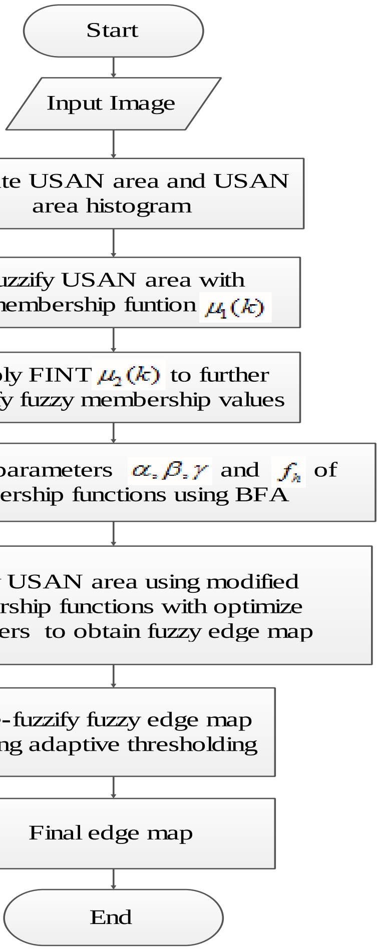

Start
Input Image
Calulate USAN area and USAN area histogram
Fuzzify USAN area with membership funtion $\mu_1(k)$
Apply FINT $\mu_2(k)$ to further modify fuzzy membership values
Optimize parameters α, β, γ and f_h of membership functions using BFA
Fuzzify USAN area using modified membership functions with optimize parameters to obtain fuzzy edge map
De-fuzzify fuzzy edge map using adaptive thresholding
Final edge map
End

Fig. 5. Block diagram of the proposed approach.

In other words a proper fuzzy set is converted to two-value membership fuzzy set i.e. binarization. This classical set is obtained using adaptive thresholding of the fuzzy edge map. The adaptive thresholding, unlike the global threshold algorithms, allows a user to exact out a higher intensity (membership value) object from the region with varying intensity background in an image. The de-fuzzification/binarization is performed through an adaptive thresholding by analyzing the membership value of each pixel with respect to neighborhood membership values. Therefore, each image pixel is judged adaptively by the localized criteria. The pixels having membership value greater than the local threshold are marked as "object" and the rest of the pixels are considered to be "background". To achieve a binary edge map, object pixels and background pixels are assigned a value of '1'and '0', respectively. This creates a classical set, i.e. a fuzzy set having membership value '0' and '1' only. In the context of adaptive thresholding, the value of the threshold depends on the particular regions of the fuzzy edge map, hence resulting in adaptive de-fuzzification. A small region within an image is assumed to have approximately uniform intensity in Chow and Kaneko [33] and threshold depends on the local intensity. In an alternative approach, local statistical parameters in the neighborhood of a pixel, are examined to find the local threshold. The statistical measure of the neighborhood to be considered may depend on the type of image. In simplest measures, it could be *mean* of the local intensity distribution; or the median value; or the median / the mean of the minimum and maximum values, etc. In the present approach the threshold is taken as the average value of mean and median of the local intensity distribution of the fuzzy edge map. The neighborhood size of a pixel need to be sufficient to enclose enough of the foreground and background pixels, but at the same time not too large to lose the advantage of the local statistic. When the variance of the local region is too low the local threshold is replaced with the global threshold to avoid threshold become too low in case of the smooth region. Fig. 5 represents the block diagram of the proposed approach.

VI. Experimental Results and Analysis

The performance of the algorithm is compared against some well-established edge detection methods, namely Canny, Sobel, Prewitt, Roberts, LoG, and some of the latest edge detection methods described in BFED [20], FEDBF [21], PSODE [23], ACED [25], EDUG [26], GEDT [27], IVFE [13], Canny-gs [15], WFIE [14], M-Sobel [6]. The BFED and FEDBF are based on bacterial foraging algorithm, PSODE is based on particle swarm optimization, ACED is based on ant colony optimization, EDUG and GEDT are based on theory of gravitation. We have also compared our work with some state of art edge detectors based on the fuzzy theory [13-15] and multiscale edge detector [6]. The human judgment is supposed to be the best measure in case of edge detection, but at the same time it can be subjective. Moreover, human visual system (HVS) has its own limitations; therefore, the quantitative measures are also engaged to judge the performance of the proposed approach. The color image edge detection may be performed by applying the proposed algorithm to each component, i.e. red, green, and blue. But, our algorithm uses the HSI color model to represent the color image. In this color model, Hue (H), and Saturation (S) represents chromatic information, whereas intensity (I) represents achromatic information.

TABLE I
Initial and Optimal Value of Parameters for Lena Image

Parameter	Initial Value (Random)	Optimized Value for "Lena"	Optimized Value for "Cameraman"
α	5	3.6348	2.9674
β	10	17.4827	15.6352
γ	1.2	0.8732	1.3172
f_h	15	24.0658	21.5961

The edge information is non-chromatic in nature; therefore, only the intensity component is required for edge detection in nearly all practical applications. The use of HSI model over RGB model makes the proposed algorithm universally applicable, i.e. for both gray and color images. The image data set for performance analysis consists of various benchmarks,

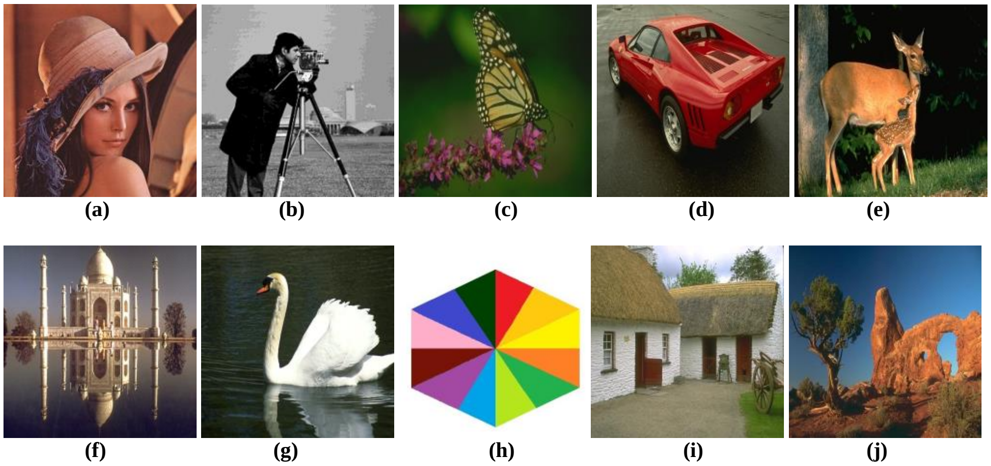


Fig. 6. Test images: (a) Lena (b) Cameraman (c) Butterfly (d) Car (e) Deer (f) Taj (g) Swan (h) Hexagon (i) Hut, (j) Tree

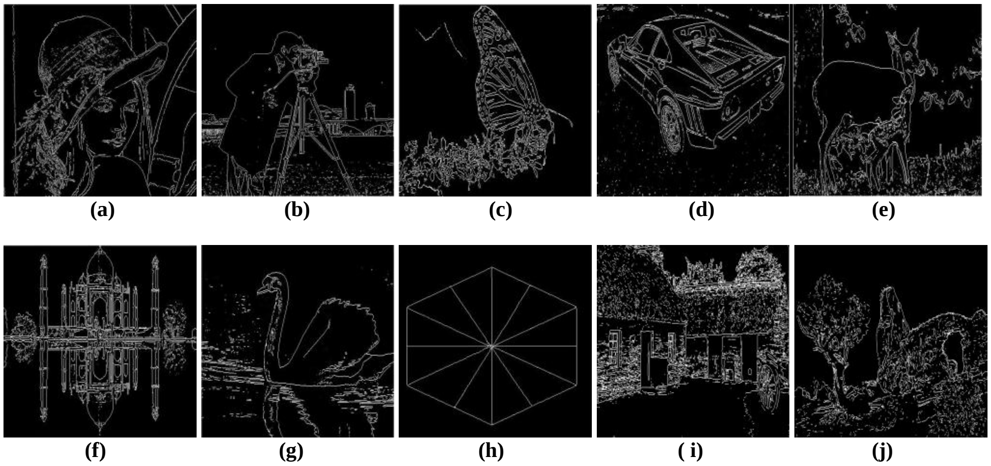


Fig. 7. (a) Lena (b) Cameraman (c) Butterfly (d) Car (e) Deer (f) Taj (g) Swan (h) Hexagon (i) Hut, (j) Tree

natural and synthetic images. We use Berkeley segmentation data set of more than 100 images as a main source of images. Ten such images and corresponding edge maps obtained by proposed algorithm, are shown in Fig. 6 and Fig. 7 respectively. The values of parameters used for Lena and Cameraman images are shown in Table I. The initial values of the parameters is same for all images. This is because, the initial values do not have much effect on the algorithm as BFA optimizes parameters according to image information. Although, if parameters are initialized with proper values, optimization process will take lesser time. The visual analysis of edge maps in Fig. 7 shows, that the proposed filter is able to detect nearly all edges present in images. It may be noted that the proposed approach detects strong and weak edges efficiently. It is evident from the results shown in Fig. 7 that in the region with fine features, only prominent edges are detected. In the highly detailed region, if all the edges are detected, then edge information becomes noisy in nature. Therefore, it is desirable to adjust the parameters in such a way that only prominent edges get detected. In the proposed approach, the parameters are tuned as per nature of the image (i.e. edge information present in the image), also the adaptive thresholding is used to suppress the unwanted edges. The edge maps by well-established and state-of-art edge detectors are shown in Fig. 8 and Fig. 9 for ‘Lena’ and ‘Cameraman’ images. The results of the Canny edge detector, in Fig. 8(a) and 9(a), shows that it gives good edge connectivity, but there are a large number of false edges or unwanted edges. This makes edges map noisy and if the threshold is reduced many prominent edges goes missing. The Sobel edge detector, in Fig. 8(b) and 9(b), on the other hand, gives the lesser number of false edges, but the edge connectivity is poor. Similarity the results for other traditional edges detectors, i.e. Prewitt, Roberts and LoG in Fig 8(c-e) and 9(c-e), shows that either some prominent edges are missing or too many false edges are present in the edge map. Nearly, all the tradition methods shows poor edge connectivity, if the threshold is high and noisy edge map, if the threshold is low. Finding an optimum threshold for each type of image is a critical issue in all these algorithms. The SUSAN edge detector gives thick edges as shown in Fig. 8(f) and 9(f), as the detector works on USAN area alone. The SUSAN edge detector is unable to detect weak edges present in the images. The detector BFED [20] is based on combination of BFA and ACO. Since the algorithm BFED

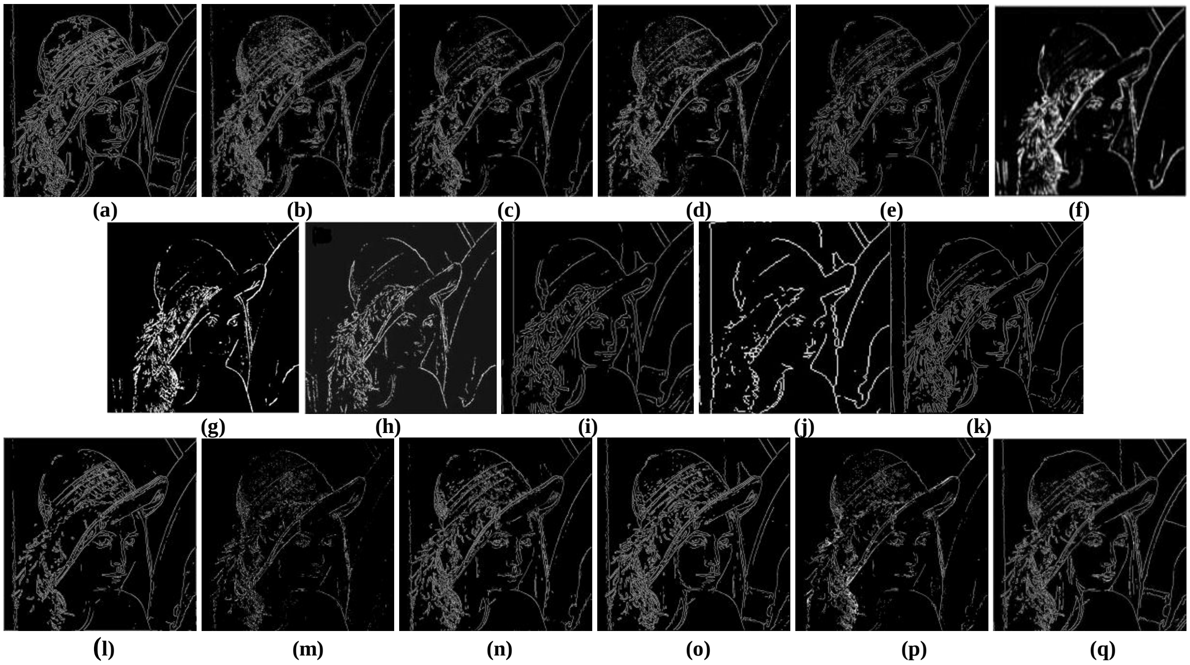


Fig. 8. (a) Canny (b) Sobel (c) Prewitt (d) Roberts (e) LoG (f) SUSAN (g) BFED (h) FEDBF (i) PSODE (j) ACED (k) EDUG (l) GEDT (m) IVFE (n) M-Sobel (o) Canny-gs (p) WFIE (q) Proposed Algorithm

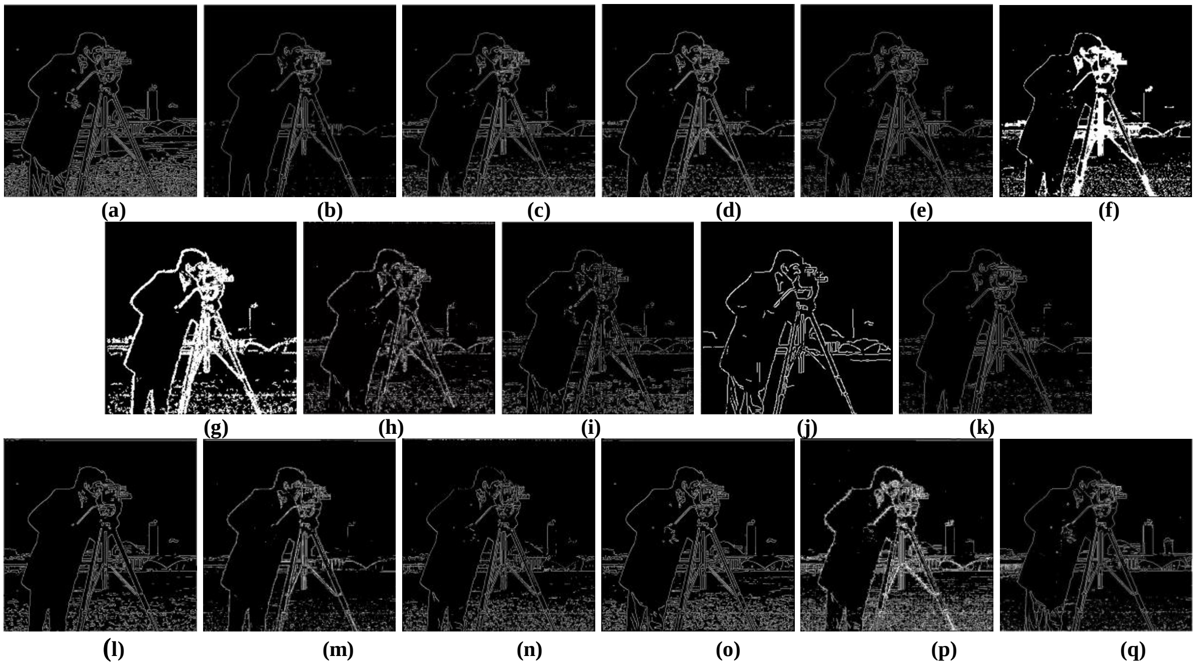


Fig. 9. (a) Canny (b) Sobel (c) Prewitt (d) Roberts (e) LoG (f) SUSAN (g) BFED (h) FEDBF (i) PSODE (j) ACED (k) EDUG (l) GEDT (m) IVFE (n) M-Sobel (o) Canny-gs (p) WFIE (q) Proposed Algorithm

utilizes the ability of BFA to find the optimum path, which is considered as edge map, therefore, mostly strong edges are detected. Also results in Fig. 8(g) and 9(g) shows some disconnected edges. The algorithm in FEDBF [21], also uses BFA to find the edge map, and the fuzzy reasoning is used to find the direction of an optimal path (edge). Moreover, swarming step of BFA is not considered to avoid the convergence of bacteria to only thick edges. The results of FEDBF algorithm are shown in Fig. 8(h) and 9(h). It is observed that FEDBF gives improved results as compared to other BFA based algorithm BFED, but many of the weak edges are missing. The algorithm in PSODE is based on particle swarm optimization. The results of PSODE in Fig. 8(i) and 9(i), shows that it detects more weak edges, but it has introduced some distortion in the shape of many edges. In ACED [25], the ant colony algorithm based algorithm is proposed and the results for the same are shown in Fig. 8(j) and 9(j). It is evident from these results that ACED show poor connectivity in the detected edges. The EDUG [26] is an edge detector based on law of universal gravity, and intensity is considered as gravitational mass. The results of EDUG in Fig 8(k) and 9(k), shows that edge connectivity is better, but weak edges are not properly detected. Also the edge detector GEDT [27] considers the intensity of pixel as gravitational mass. A fuzzy set of edges is constructed using gravitation force as membership degree. Canny based edge detector is applied to fuzzy edge image to obtain edges. The authors in [27] has considered various t-norms for implementation, and minimum is best suited for most of the cases. The results for GEDT are shown is Fig. 8(l) and 9(l). These edge maps are comparable to that of canny edge detector. The edge connectivity is good for strong edges, but weak edges are not detected efficiently. The problem in weak edge detection is due to weak gravitational for low intensity pixels. The edges obtained by IVFE [13] are shown in Fig. 8(m) and 9(m). The results shows some missing edges and few noisy edges. The edge detector IVFE divides edge pixels in three category using interval valued fuzzy sets. If pixels with small membership value are included in edge map, it becomes noisy, and if they are excluded, it shows poor edge connectivity. The results for multiscale edge detector M-Sobel [6] are shown in Fig. 8(n) and 9(n). Although, results are improved as compared to Sobel edge detector, yet weak edges are not properly detected. The edge detector Canny-gs [15] fuzzifies gradient magnitudes using different parametric functions. We use sigmoidal membership implementation for comparison purpose. The edge maps for Canny-gs are shown in Fig. 8(o) and 9(o). The edge maps have better connectivity, but some noisy edges are also present. The authors in WFIE [14] presented a method to generate the fuzzy edge map, but their focus is not to develop an edge detector. WFIE converts fuzzy edge map into crisp edges using binarization and thining. The results of WFIE are shown in Fig. 8(p) and 9(p). WFIE detects strong edges properly, but weak edges shows some missing edges. Although, a proper de-fuzzification may result in good edge map. The visual analysis of all above edge detectors shows that most of the edge detectors fail to detect weak edges. Majority of the edge detectors detects false edges if the threshold is small and leaves missing edges if the threshold is high. The visual perception of Fig. 8 and 9, shows the proposed algorithm performs better or at least comparable to many traditional and state-of-art edge detectors, in weak edge detection. The proposed edge detector is able to detect strong as well as weak edges accurately.

A. Quantitative Analysis

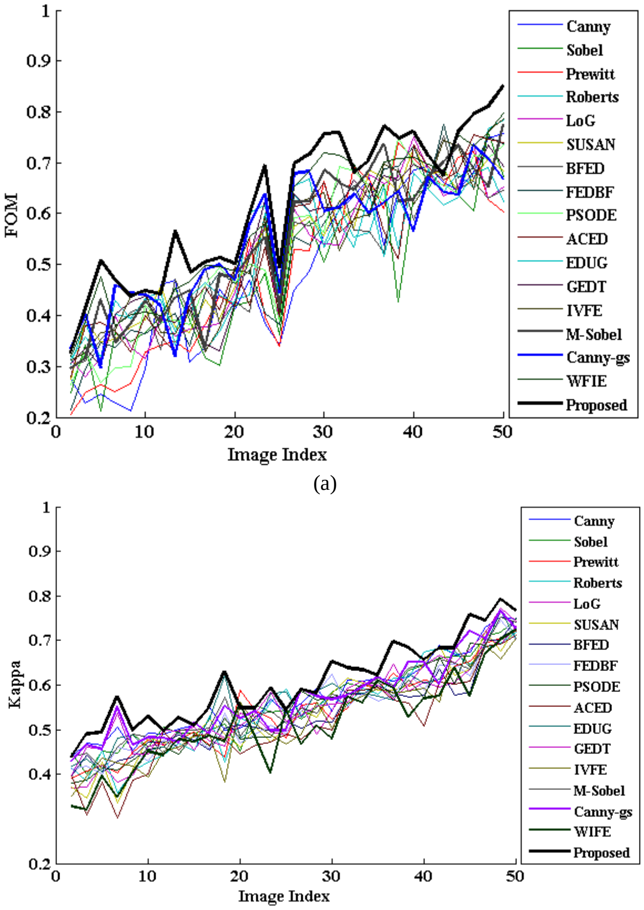
The quantitative analysis is performed using some well-known methods, i.e. Pratt's figure of merit (FOM) [34], Kappa value [35], Shannon's entropy [36], and a recently proposed method edge strength similarity [37]. Since, there is no single universally accepted method for quantitative analysis of edge maps, we have performed the qualitative analysis using all above methods.

1) Pratt's Figure of Merit (*FOM*)

Pratt's FOM [34] is one of the highly used measure to evaluate the performance of edge detectors in the literature. This measure uses an ideal edge map of an image to compare with the edges obtained by edge detector. The Pratt's FOM is defined as:

$$FOM = \frac{1}{\max\{N_I, N_D\}} \sum_{i=1}^{N_D} \frac{1}{1+\tau(d_i)^2} \tag{19}$$

Where N_I and N_D are the number of ideal and actually detected edge points respectively, d_i is the distance between i^{th} edge point detected and ideal edge point, and τ is the scaling factor with a value of 1/9 as in [34]. The expression for FOM in Eqn. (19), suggests that the ideal edge map for given image is required for exact measure. The problem of finding ideal edge map is dealt by taking synthetic images, where exact locations of the edges are known or by taking the best of the well-established methods Canny, Sobel, Prewitt, Roberts and LoG and combine them to give good approximation to ideal edge map.



(a)

(b)

Fig. 10. The quantitative metric curves of the different algorithms: (a) Pratt's FOM, (b) Cohen's Kappa.

The FOM values are calculated for the edge maps of images from the Berkeley image database, using various edge detectors. The images are sorted (in the ascending order), and indexed based on the mean value of the FOM metric by different algorithms. In Fig. 10(a), we plot the FOM curves for different algorithms. Different colors are used to distinguish various algorithm. The FOM curve for the proposed algorithm is highlighted with black color. The proposed algorithm gives highest FOM values for most of the images. However, for few images, FOM values are not much better than other traditional edge detectors. This is because Pratt's FOM method is dependent on the approximation of the ideal edge map, which is difficult to get for some images.

2) Cohen's Kappa

Cohen's Kappa [35] is used for further investigation of experimental results obtained by the proposed algorithm. Cohen's Kappa is defined as:

$$Kappa = \frac{P(a) - P(e)}{1 - P(e)} \tag{20}$$

Where $P(a)$ is observed probability of agreement and $P(e)$ is the probability of random agreement of two approaches. It can be noted from the definition of Cohen's Kappa, that the agreement due to 'chance' i.e. probability of random agreement is factored out. It makes comparison more realistic, i.e. less random. Cohen' Kappa is used for comparison of two edge maps by obtaining pixel to pixel agreement. One of the edge maps is considered as ground truth image, which is a majority image [20]. A majority image/edge map is obtained by using the edge maps of Canny, Sobel, Prewitt, Roberts and LoG. In the majority edge map, a pixel is considered as edge pixel if the majority of the edge maps indicates it to be an edge pixel. In the proposed approach, observed probability of agreement $P(a)$, is calculated as the ratio of number of pixels declared as edge pixels (by both majority edge map and other edge map in question), to the total number of pixels. The probability of random agreement $P(e)$ is obtained by adding the joint probabilities of labelling a pixel as edge pixel and non-edge pixel by both. The Cohen's Kappa is calculated for the edge maps obtained by various algorithms with respected to majority edge map, and the values are plotted in Fig. 10(b). The Kappa metric curve for the proposed algorithm is highlighted with black color. It may be observed that the proposed method has highest Kappa values for nearly all types of images. However, the Kappa value is slightly less for few images. It is because some of the pixels are declared as non-edge pixels in majority edge map, but they are termed as a weak edge pixel in the proposed algorithm. Therefore, the proposed algorithm is able to detect weak edges better than other traditional filters, which constituted majority edge map.

3) Shannon's Entropy

The quantitative measure Pratt's FOM requires the ideal edge map to compare, which is approximated with the edge maps of the well-established edge detectors. The approximated ideal edge map or majority edge map does not give much information about weak edges. Therefore, Pratt's FOM and Cohen's Kappa are a not true measure for weak edges. Thus, results are further investigated using Shannon's entropy [36], which is defined as:

$$H(I) = -\sum_{i=0}^{L-1} p_i \log_2 p_i \tag{21}$$

Where I is the image or edge map, whose entropy is to be calculated, p_i is probability of a pixel having intensity i, i.e. normalized frequency of occurrence of pixel with intensity i𝕜, L𝔽 represents intensity levels. Shannon entropy represents uncertainty, i.e. information contained within the image. Although, a higher value of entropy does not always reflect the higher edge information. This is because the entropy value may be high due to noisy/ false edges, double edges or broad edges. Also, a very low value of entropy reflects loss of the prominent edges in the edge map. To obtain a fair value of entropy of the ideal edge map, we have used the majority edge map [20] as discussed in section VI.A.2. The majority edge map should have nearly all the prominent edges, but it may not be able to fetch the information about weak edges as shown in Fig. 11. Therefore, the desired entropy should be close to the value of the entropy of the majority edge map.

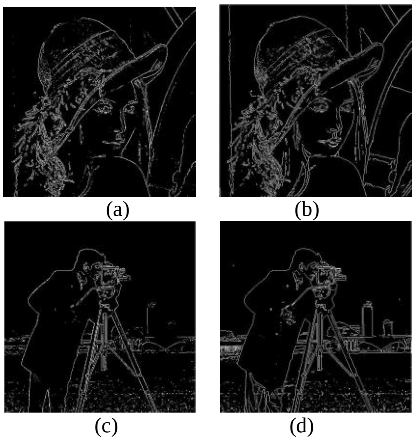

(a) (b)
(c) (d)

Fig. 11. (a) Majority edges, (b) Proposed algorithm (c) Majority edges (d) Proposed algorithm

The Shannon entropy is calculated for edge maps of the images form Berkeley's database, by different edge detectors; and plotted in Fig. 12(a). The entropy for majority edge map is also plotted for the reference. The reference curve and the proposed algorithm curve are highlighted in blue and black colors, respectively. It may be observed that the entropy curve of the proposed algorithm traces closely to that of the majority edge map. However, the proposed algorithm gives little higher values of entropy than the majority edge map. This is because of the weak-edge information contained in edge maps.

4) *Edge Strength Similarity based Edge Quality Metric* (*ESSEM*)

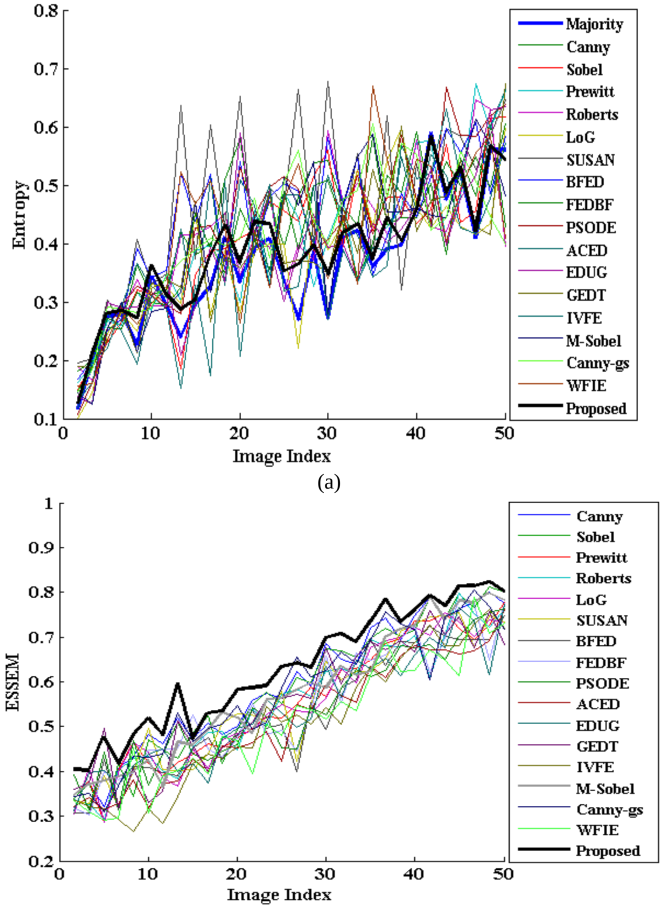
Edge strength similarity based edge quality metric (ESSEM) is based a quality metric described in [37]. It compares the quality of the distorted image with the reference/original image. In [37], the edge strength in the diagonal direction of i^{th} pixel is defined as:

$$\chi_i^{j,j+2}(I) = \left|\partial I_i^j - \partial I_i^{j+2}\right|^{\rho} \tag{22}$$

Where ∂I_i^j denotes derivative of image I at i^{th} pixel in the direction j and ρ = 0.25. The four directions of the edges are denoted as $j = 1, 2, 3, 4$ respectively as shown in Fig. 13. The total edge strength at i^{th} pixel is defined as:

$$\chi(I,i) = \max\left\{\chi_i^{1,3}(I), \chi_i^{2,4}(I)\right\} \tag{23}$$

The edge strength of distorted image is also calculated in a similar manner. Since, we have to examine the edge quality of the edge map, the edge strength of the original image is compared with the edge map.



(a)

(b)

Fig. 12. The quantitative metric curves of the different algorithms: (a) Entropy, (b) ESSEM.

The ESSEM is defined as:

$$ESSEM\,(I,\xi) = \frac{1}{MN}\sum_{i=1}^{MN}\frac{2\,\chi(I,i)\,\chi(\xi,i)+\varepsilon}{(\chi(I,i))^2+(\chi(\xi,i))^2+\varepsilon} \tag{24}$$

Where ξ is the edge map, MN is the total number of pixel present in image/edge map, and ε is scaling parameter with value 0.01.

The ESSEM metric for the edge maps obtained by various algorithms is calculated with respect to their original images and values. The ESSEM metric curves for different algorithms are plotted in Fig. 12(b), and the metric curve for the proposed algorithm is highlighted with black color. The ESSEM values do reflect the capability of the edge detector in abstracting the edge information from an image, but the exact value depends upon many factors such as nature, contents, fine details present in the image. It can be observed that the proposed algorithm gives highest values of ESSEM nearly for all the images. Thus, the ESSEM metric supports the fact that our proposed algorithm is able to detect nearly all the prominent and weak edges.

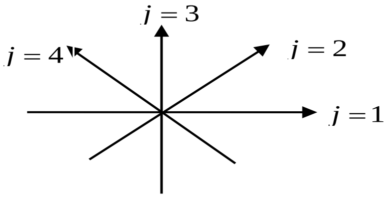


Fig. 13. (a) Directions of edges for directional derivative

B. Statistical Analysis of Quantitative Results

We perform hypothesis testing using statistical analysis of results obtained in various quantitative measures. The proposed algorithm is compared with other algorithms using the same set of images; thus, a paired t-test analysis [38] is chosen. We propose following null hypothesis (H_0) and alternate hypothesis (H_1):

H_0: The proposed algorithm does not performs better than the algorithm X.

H_1: The proposed algorithm performs better than the algorithm X.

Here method X is any algorithm against which the proposed method is being compared. The higher values of the measures FOM, Kappa, and ESSEM, shows better performance. Therefore, one-tail *t*-test is performed for these measures. However, the entropy value closer to majority edge map reflects better performance. Thus a two-tail *t*-test is performed between majority edge map and other algorithms for entropy values. The *p*-values [38] for *t*-test are shown in Table II. The *p*-value for the *t*-test between the majority edge map and the proposed approach is 1.6E-05. The *p*-values in the Table II shows that the null hypothesis is rejected in each case with at least 0.01 level of significance. The statistical t-test analysis has proved the supremacy of the proposed algorithm over well-established and state-of-art edge detectors.

C. Computational Time Analysis of Proposed Algorithm

We perform computation time analysis by evaluating execution time of the proposed algorithm and other algorithms used for comparison. The execution of algorithms are carried out in a computer with core i3 processor, 2 GB RAM and Windows 7 operating system. The implementation of the algorithms are done in MATLAB R2010b. We have executed each algorithm for 30 times for 50 test images, and an average execution time is recorded in Table III. It is observed that traditional edge detector takes almost negligible time in comparison to nearly all state-of-art edge detectors. This seems to be logical as most of the traditional algorithms involve a convolution with a kernel with few morphological operations. The fuzzy logic based edge detectors: GEDT [27], IVFE [13], Canny-gs [15], M-Sobel [6], and WFIE [14] takes more execution time in comparison to classical edge detectors as most of them are built on either Canny or Sobel edge detectors. All the edge detectors based on evolutionary algorithm take longer execution time.

TABLE II
The p-values of The Paired t-test between an Algorithm X and The Proposed Algorithm

Alg. A→	Canny	[20]	[21]	[23]	[25]	[26]	[27]	[13]	[6]	[15]	[14]
FOM	3.9E-08	6.1E-11	2.4E-09	9.6E-11	6.5E-10	1.5E-08	1.0E-10	5.4E-10	1.5E-07	2.1E-06	1.4E-10
KAPPA	6.3E-06	5.9E-11	4.3E-12	1.5E-10	1.0E-10	2.8E-08	3.4E-08	1.3E-09	1.8E-09	9.6E-08	6.3E-10
ESSEM	5.8E-08	6.8E-12	2.5E-09	1.2E-11	5.3E-15	1.9E-11	1.6E-10	1.7E-13	1.3E-09	6.1E-08	9.2E-14
ENTROP	5.5E-04	2.3E-02	2.5E-03	8.3E-02	3.6E-02	1.3E-01	7.7E-02	6.5E-01	3.2E-04	9.7E-04	3.0E-02

TABLE III
Average Computation Time per Image (in Seconds)

Algo.	[1]	[20]	[21]	[23]	[25]	[26]	[27]	[13]	[6]	[15]	[14]	Proposed
Time	0.54	**25.41**	**19.37**	**17.38**	**26.39**	2.31	1.96	2.83	3.74	2.16	1.28	**11.46**

This is due to the involvement of parameter learning. The gravitational rule based edge detectors (EDUG [26], GEDT [27]) take lesser time in comparison to other evolutionary computation based edge detector. The proposed algorithm takes longer execution time than other fuzzy based edge detectors. This is because we optimize the parameters using BFA. It is evident from the Table III that the proposed algorithm takes the smallest time among all other evolutionary computation based edge detector. As discussed earlier, most of the evolutionary based approaches try to find edges as an optimal path of search-agent (ant, particle, or bacterium). The proposed algorithm optimizes parameter based on fuzzy measures. It requires the lesser number of search agents and hence lesser computation. Although, the proposed algorithm requires longer time than traditional as well as other fuzzy based edge detectors, but there is always a tradeoff between the accuracy and computation time. The proposed algorithm is capable of detecting weak edges with higher accuracy in comparison to other edge detectors.

VII. Conclusion

We have presented a fuzzy system for edge detection. The algorithm is designed to detect strong and weak edges efficiently. We have defined a USAN area histogram-based Gaussian membership function, a new parametric fuzzy intensifier FINT, weak and strong fuzzy edge quality, and sharpness factor. The fuzzy entropy and sharpness factor are used to devise the objective function for optimization. The optimized parameters obtained from BFA are used to construct a fuzzy edge map. The fuzzy edge map is de-fuzzified using adaptive thresholding to result in binary edge map. The results obtained by the proposed algorithm are examined visually and quantitatively. It is compared with well-known traditional methods and state-of-art methods. The proposed algorithm has following key advantages:

1. The proposed algorithm detects edges with more accuracy i.e. better localization of edges, which is supported with visual analysis and quantitative analysis.
2. Improved edges connectivity.
3. The algorithm uses integration of derivatives, which makes our edge detector robust against noisy edges. Therefore, the proposed algorithm detects weak edges with almost negligible noisy edges.

The drawback of the proposed algorithm is that it takes longer execution time as compared to non-evolutionary computation base edge detectors. The computation time can be reduced significantly by using parallel computing. Theoretically in BFA, bacteria can move simultaneously in search space, but in practical implementation we need to code in sequential manner. A parallel computing hardware/software implementation would make the proposed algorithm comparatively faster. The future scope of the proposed work is to develop parallel computing based implementation to reduce the execution time.

References

[1] J. Canny, “A computational approach to edge detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 1986, vol. 8, no.1, pp. 679-698, 1986.

[2] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing,* 3rd ed., Pearson India.

[3] M. Basu, “Gaussian Based Edge Detection Methods: A Survey,” *IEEE Trans. on systems, man and cybernetics,* vol. 32, no. 3, pp. 252 – 260, 2002.

[4] R.A. Salinas, C. Richardson, M.A. Abidi, and R.C. Gonzalez, “Data Fusion: Color Edge Detection and Surface Reconstruction through Regularization,” *IEEE Trans. on Industrial Electronics,* vol. 43, no. 3, pp. 355-363, 1996.

[5] P. Perona, J. Malik “Scale-Space and Edge Detection Using Anisotropic Diffusion,” *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 629-639, 1990.

[6] C. L. Molina, B. De Baets, H. Bustince, and E. Barrenechea, “Multiscale edge detection based on Gaussian smoothing and edge tracking,” *Knowledge-Based Systems*, vol. 44, pp. 101-111, 2013.

[7] J.C. Bezdek, R. Chandrasekhar, and Attikiouzel, “A geometric approach to edge detection,” *IEEE Trans. on Fuzzy Systems*, vol. 6, no. l, pp. 52-75, 1998.

[8] F. Russo, “Edge Detection in Noisy Images Using Fuzzy Reasoning,” *IEEE Trans. on Instrumentation and Measurement,* vol. 47, no. 5, 1998.

[9] I. Bloch, “Fuzzy sets in image processing,” in *Proc. of ACM Symposium on Applied Computing*, New York, USA, pp. 175 – 179, March 6-8,1994.

[10] S. E. El-Khamy, I. Ghaleb, and N. A. El-Yamany, “Fuzzy edge detection with minimum fuzzy entropy criterion,” in *Proc. of IEEE Electrotechnical Conference MELECON*, Egypt, pp. 498-503, 7-9 May, 2002.

[11] Jinbo Wu, Zhouping Yin, “The Fast Multilevel Fuzzy Edge Detection of Blurry Images,” *IEEE signal processing letters*, vol. 14, no. 5, pp. 344-347, 2007.

[12] Yishu Zhai; Xiaoming Liu, “Multiscale Edge Detection Based on Fuzzy C-Means Clustering,” in *Proc. of 1st International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, pp. 1201-1204, June 19-21, 2006.

[13] H. Bustince, E. Barrenechea, M. Pagola and J. Fernandez, “Interval-valued fuzzy sets constructed from matrices: Application to edge detection,” *Fuzzy Sets and Systems,* vol. 160, pp. 1819-1840, 2009.

[14] E. Barrenechea, H. Bustince, B. De Baets and C. L. Molina, “Construction of Interval-Valued Fuzzy Relations with Application to the Generation of the Fuzzy Edge Images,” *IEEE trans. on Fuzzy Systems*, vol. 19, no. 5, pp. 819-830, 2011.

[15] C. L. Molina, B. De Baets, and H. Bustince, “Generating fuzzy edge images from gradient magnitudes,” *Computer Vision and Image Understanding*, vol. 115, pp. 1571-1580, 2011.

[16] P. M. Pinto, P. Couto, H. Bustince, E. Barrenechea, M. Pagola, and J. Fernandez, “Image segmentation using Atanassov’s intuitionistic fuzzy sets,” *Expert Systems with Applications*, vol. 40, pp. 15-26, 2013.

[17] M. Pagola, C. L. Molina, J. Fernandez, E. Barrenechea, and H. Bustince, “Interval Type-2 Fuzzy Sets Constructed From Several Membership Functions: Application to the Fuzzy Thresholding Algorithm,” *IEEE trans. on Fuzzy Systems*, vol. 21, no. 2, pp. 230-244, 2013.

[18] K. M. Passino, “Biomimmicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[19] M. Hanmandlu, O. P. Verma, N. K. Kumar, M. Kulkarni, “A Novel Optimal Fuzzy System for Color Image Enhancement Using Bacterial Foraging,” *IEEE Trans on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2867 – 2879, 2009.

[20] O. P. Verma, M. Hanmandlu, P. Kumar, S. Chhabra, A. Jindal, “A novel bacterial foraging technique for edge detection,” *Pattern Recognition Letters*, vol. 32, pp. 1187-1196, 2011.

[21] O. P. Verma, M. Hanmandlu, A. K. Sultania and A. S. Parihar, “*A Novel Fuzzy System for Edge Detection in noisy image using Bacterial Foraging,”* Multidimensional Systems and Signal Processing, vol. 24, no. 1, pp 181-198, 2013.

[22] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant System: Optimization by a Colony of Cooperating Agents,” IEEE *Trans. on systems, man, and cybernetics-part b cybernetics*, vol. 26, no. 1, 1996.

[23] M. Setayesh, M. Zhang, M. Johnston, “A novel particle swarm optimization approach to detecting continuous, thin and smooth edges in noisy images,” *Information Sciences*, vol. 246, pp. 28-51, 2013.

[24] J. Kennedy, R. Eberhart, “Particle Swarm Optimization,” in *Proc. of IEEE International Conference on Neural Networks* vol. 4, pp. 1942–1948, 1995.

[25] S. Ali Etemad, Tony White, “An ant-inspired algorithm for detection of edge features,” *Applied Soft Computing*, vol. 11, pp. 4883-4893, 2011.

[26] G. Sun, Qinhuo Liu, Qiang Liu, C. Ji, X. Li, “A novel approach for edge detection based on the theory of universal gravity,” *Pattern Recognition*, vol. 40, pp. 2766-2775, 2007.

[27] C. L. Molina, H. Bustince, J. Fernandez, P. Couto, and B. De Baets, “A gravitational approach to edge detection based on triangular norms,” *Pattern Recognition*, vol. 43, pp. 3730-3741, 2010.

[28] S. M. Smith, and J. M. Brady, “SUSAN: A new approach to low level image processing,” *International Journal of Computer Vision*, vol. 23, no.1, pp.45-78, 1997.

[29] M. Hanmandlu, O. P. Verma, P. Gangwar, S. Vasikarla, “Fuzzy Edge and Corner Detector for Color Images,” in *Proc. of 6th IEEE International Conference on Information Technology: New Generations,* Las Vegas, NV, pp. 1301 – 1306, April 27-29, 2009.

[30] S. K. Pal and R. A. King, “Image enhancement using smoothing with fuzzy sets”, *IEEE Trans. on System, Man ,and Cybernetics,* vol. 11, no. 7, pp. 494 – 501, 1981.

[31] A. De Luca, S. Termini, “A definition of Nonprobabilistic Entropy in the setting of fuzzy set theory,” *Information and Control*, vol. 20, pp. 301-312, 1972.

[32] H. Cheng, Y. Chen, Y. Sun, “A novel fuzzy entropy approach to image enhancement and thresholding,” S*ignal Processing*, vol. 75, pp. 277-301, 1999.

[33] C. K. Chow and T. Kaneko, “Automatic boundary detection of the left ventricle from cineangiograms,” *Computers and Biomedical Research*, vol. 5, no. 4, pp.388- 410, 1972.

[34] I. Abdou and W. Pratt, “Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors”, in *Proceedings of IEEE* vol. 67, no.5, pp. 756-766, 1979.

[35] Cohen J., “A coefficient of agreement for nominal Scales,” *Educational and Psychological Measurement,* vol. 20 no. 1, pp. 37-46, 1960.

[36] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[37] X. Zhang, X. Feng, W. Wang and W. Xue, “Edge Strength Similarity for Image Quality Assessment,” *IEEE Signal Processing Letters,* vol. 20, no. 4, pp. 319-322, 2013.

[38] D. A. Lind, W. G. Marchal, S. A. Wathen, *Statistical Techniques in Business and Economics*, 13th ed., Tata McGraw Hill Education Pvt. Ltd. India.

Om Prakash Verma (M’76) received his B.E. degree in Electronics and communication engineering from Malaviya National Institute of Technology, Jaipur, India, M. Tech. degree in Communication and Radar Engineering from Indian Institute of Technology (IIT), Delhi, India and PhD in the area of applications of soft and evolutionary computing in image processing from University of Delhi, Delhi, India. From 1992 to 1998 he was assistant professor in Department of ECE at Malaviya National Institute of Technology, Jaipur, India. He joined Department of Electronics & Communication Engineering, Delhi Technological University (formerly Delhi College of Engineering) as Associate Professor in 1998. Currently, he is Head & Professor in Department of Computer Science and Engineering, Delhi Technological University, Delhi, India. His present research interest includes: Applied Soft Computing, Nature Inspired Algorithms, Swarm Intelligent, Evolutionary Computing, and Image Processing.

Anil Singh Parihar received his B. Tech Degree in Electronics and Communication Engineering in 2005 from Priyadarshini College of India and the Master of Engineering degree in Electronics and Communication Engineering from Delhi College of Engineering, New Delhi, India, in 2008. He joined the Department of Information Technology at Delhi Technological University, Delhi, India as Assistant Professor in 2010. Currently, he is Assistant Professor in the Department of Computer Science & Engineering at Delhi Technological University, Delhi, India. His research interest includes Image Processing, Biometrics, Soft Computing and Evolutionary Algorithms.