EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Enhanced Edge Detection through Binary Particle Swarm Optimization and L_0 Guided Filtering

Ankush Verma*[1], Namrata Dhanda[2] and Vibhash Yadav[3]

[1]Research Scholar, Amity Institute of Information Technology, Amity University Uttar Pradesh, Lucknow Campus, Lucknow (India)
[2]Amity School of Engineering and Technology, Amity University Uttar Pradesh, Lucknow, Campus, Lucknow (India)
[3]Rajkiya Engineering College, Banda (India)

Abstract

Detecting edges is a critical task in image processing, forming the backbone of many computer vision applications such as object recognition, image segmentation, and scene understanding. This paper introduces a novel edge detection method that synergistically combines Binary Particle Swarm Optimization (BPSO) with L_0 Guided Filtering to address the challenges of detecting edges in noisy and complex images. The motivation for this approach stems from the need to enhance edge detection accuracy and robustness, particularly in environments where traditional methods struggle with noise and detail preservation.

The innovation of our method lies in the integration of BPSO with L_0 Guided Filtering. BPSO is employed to optimize the parameters crucial for edge detection by exploring the solution space efficiently through binary particles. This optimization process helps in selecting the most effective parameters that significantly influence edge detection performance. Once the optimal parameters are identified, they are utilized within the L_0 Guided Filtering framework. L_0 Guided Filtering is a sophisticated technique known for its ability to preserve fine image details while effectively suppressing noise and artifacts. By combining these two techniques, our method benefits from the efficient search capabilities of BPSO and the edge-preserving properties of L_0 Guided Filtering.

The results of our experimental evaluations on benchmark datasets highlight the effectiveness of the proposed method. Compared to traditional edge detection techniques, our approach demonstrates superior edge localization, providing more accurate and well-defined edges. Additionally, it shows improved robustness to noise, making it particularly valuable for applications in challenging imaging conditions. The method's ability to adapt to diverse image characteristics and its efficient parameter optimization contribute to its enhanced performance.

Keywords: Edge detection, Guided Filtering, BPSO

Received on 5 May 2024, accepted on 02 September 2024, published on 29 October 2024

doi: 10.4108/eetsis.6282

1. Introduction

In the realm of computer vision, edge detection stands as a fundamental and pivotal task, forming the cornerstone for numerous image processing applications. The ability to accurately identify and delineate edges within images is paramount for tasks such as object recognition, scene understanding, and image segmentation [1]. The pursuit of robust edge detection methodologies has spurred the exploration of innovative techniques that can navigate the challenges posed by diverse image characteristics, including

*Corresponding author. Email: newankushphd@gmail.com

noise, variations in illumination, and complex structural details [2]. This paper introduces a novel approach to edge detection, amalgamating the strengths of Binary Particle Swarm Optimization (BPSO) [3] with the precision of L_0 Guided Filtering. Edge detection, as a computational challenge, requires a delicate balance between preserving fine details and suppressing noise, necessitating techniques that exhibit adaptability and resilience across a spectrum of image scenarios. BPSO, a variant of the well-established Particle Swarm Optimization (PSO) [4] algorithm, introduces a binary encoding scheme for particles exploring the solution space. This binary nature endows BPSO with an efficient and rapid convergence toward optimal solutions, making it a compelling candidate for parameter optimization in edge detection scenarios. The integration of BPSO within the framework of L_0 guided filtering [5] seeks to harness the collective intelligence of particles to enhance the adaptability of the filter, ultimately refining the edge detection process. L_0 guided filtering, known for its edge preserving characteristics, provides a valuable foundation for our proposed methodology.

This paper details the formulation, implementation, and evaluation of the BPSO Based Edge Detection under L_0 Guided Filtering. Through comprehensive experimentation on benchmark datasets, we explore the efficacy of our proposed approach, highlighting its potential to significantly advance the state of the art in edge detection. The integration of BPSO and L_0 guided filtering not only contributes to the technical aspects of edge detection but also holds promise for enhancing the performance of computer vision systems across a multitude of applications.

2. Literature Survey

The field of image processing continually evolves with the pursuit of enhanced techniques for edge detection—a critical task pivotal to numerous applications in computer vision. Classical methods such as Canny [6] and Sobel [7] have laid the foundation for edge detection, each with its strengths and limitations. However, contemporary advancements explore innovative approaches that transcend the conventional frameworks. This compilation delves into a diverse set of edge detection methods, ranging from classical to cutting edge, including those inspired by bio inspired paradigms like Ant Colony Optimization (ACO) [8] and fuzzy logic [9]. Additionally, it explores the integration of optimization algorithms such as Particle Swarm Optimization (PSO) with techniques like guided filtering and sharpening, showcasing the versatility and adaptability of these methods across various image processing scenarios. Each referenced method is examined in detail, shedding light on its unique contributions, strengths, and limitations, providing a comprehensive overview of the current landscape of edge detection methodologies.

Canny Edge Detection [6]: The Canny edge detection method, proposed by John Canny, is a classical technique widely used in image processing and computer vision. This method is renowned for its ability to identify edges with high precision and suppress noise effectively. Operating in multiple stages, Canny edge detection involves gradient computation, non-maximum suppression, and edge tracking by hysteresis. Despite its effectiveness, one drawback is the acceptance of a significant number of false edges. This issue arises due to the method's reliance on convolutional kernels and its sensitivity to threshold parameters.

Sobel Operator [7]: The Sobel operator, a classical edge detection method, is particularly known for its simplicity and computational efficiency. It operates by convolving an image with Sobel kernels to calculate the gradient magnitude. While Sobel is adept at highlighting edges, it possesses limitations. In certain scenarios, genuine edges may be erroneously discarded, leading to an incomplete representation of the image's structure. This drawback is attributed to the method's sensitivity to predefined thresholds, which can affect the trade off between sensitivity and specificity.

ACO Based Edge Detection [8]: Edge detection techniques inspired by Ant Colony Optimization (ACO), as highlighted in reference [8], introduce a bio inspired paradigm into image processing. ACO algorithms mimic the foraging behavior of ants to optimize paths, and when applied to edge detection, they exhibit characteristics such as adaptability and robustness. The method referenced in [8] demonstrates minor sensitivity to noise, good preservation of edge connectivity, and the absence of double edges. This indicates its potential for providing reliable edge information while mitigating some of the limitations associated with classical methods.

Fuzzy Edge Detection [9]: Fuzzy edge detection methods leverage fuzzy logic to address the challenges encountered by classical approaches. By incorporating fuzziness into the edge detection process, these methods aim to enhance adaptability and robustness in handling uncertainties. The fuzzy edge detection technique, as referenced by [9], is acknowledged for its moderate noise tolerance, good edge connectivity preservation, and a balanced approach to handling double edges. Fuzzy logic allows for a more flexible interpretation of image features, making it particularly valuable in scenarios where precise boundaries may be challenging to define.

ACO+ Guided Filtering [10, 11]: The combination of Ant Colony Optimization (ACO) with guided filtering, as described in reference [10], presents an innovative approach to edge detection. Guided filtering, known for its ability to enhance image details while preserving edges, is synergistically integrated with ACO to leverage the strengths of both paradigms [11]. The referenced method achieves minor sensitivity to noise, good edge connectivity preservation, and the absence of double edges. This collaborative approach aims to harness the advantages of ACO for optimization while leveraging guided filtering for effective edge enhancement.

Fuzzy+ L0 Guided Filtering [12, 13]: The method introduced in reference [12] combines fuzzy logic with L_0 guided filtering, showcasing a novel strategy for edge detection. Fuzzy logic introduces adaptability to handle uncertainties, while L_0 guided filtering contributes to edge preserving contrast enhancement [13]. The referenced method exhibits moderate sensitivity to noise, excellent preservation of edge connectivity, and the presence of significant double edges. This suggests a successful integration of fuzzy logic and guided filtering to achieve a balanced and robust edge detection outcome.

PSO Based Edge Detection [14]: PSO, as applied to edge detection and referenced in [14], introduces a heuristic optimization approach inspired by the collective behavior of particles. This method leverages the collaborative exploration of solution space to enhance edge detection [15]. The PSO based edge detection method demonstrates moderate sensitivity to noise, average preservation of edge connectivity, and the absence of double edges. This suggests that PSO brings optimization capabilities to edge detection while maintaining a reasonable balance in handling noise and preserving edge structures.

PSO+ 5% Sharpening Edge Detection: The variant of PSO edge detection with 5% sharpening, as referenced in the specific configuration [16], introduces a sharpening technique to enhance the edge detection outcome. The method achieves moderate sensitivity to noise, average preservation of edge connectivity, and the absence of double edges. The incorporation of sharpening aims to improve the clarity and definition of detected edges, showcasing the adaptability of PSO in accommodating additional image enhancement strategies.

PSO+ 10% Sharpening Edge Detection: Similarly, the PSO based edge detection method with 10% sharpening, as outlined in [16], introduces a higher degree of sharpening to the edge detection process. This variant achieves moderate sensitivity to noise, good preservation of edge connectivity, and the absence of double edges. The increased sharpening level aims to further accentuate the detected edges, offering a configurable approach to address specific image processing requirements.

Despite the effectiveness of soft computing methods in edge detection, the limitations of these techniques are acknowledged due to their reliance on a singular optimal solution, leading to the rejection of many true edges [8]. To mitigate these limitations, researchers have turned to guided image filtering as a complementary enhancement for weak edges [10, 14]. Recent advancements have also integrated guided image filtering with fuzzy based edge detection [12]. Given the shortcomings identified in the literature, our research article aims to address the following key requirements effectively:

1. Improve edge connectivity.
2. Enhance the uniformity of edge width.
3. Minimize inaccuracies to achieve edge detection results closely aligned with the ground truth.

Through this endeavor, we aim to contribute to the advancement of edge detection methodologies, striving for improved accuracy, connectivity, and qualitative analyses.

Other State of the art methods [17-20]:

The multi-stream and multi-scale fusion network proposed in [17] represents an advanced method in edge detection, aiming to enhance the detection performance by addressing the complexity of edge features and their variations across scales. Edge detection with transformers represents a significant advancement by leveraging the self-attention mechanisms of transformer architectures to capture global context and intricate edge details. However, this approach can be computationally demanding and requires substantial training data, making efficient model design and integration with existing techniques important considerations for optimal results [18].

Refined edge detection with a cascaded and high-resolution convolutional network involves a sophisticated approach that improves edge accuracy by utilizing a series of convolutional layers arranged in a cascaded fashion. This approach addresses challenges like detail loss and noise sensitivity, providing more accurate and reliable edge maps [19].

An exclusive U-Net designed for fine and crisp edge detection utilizes the U-Net architecture's strengths in capturing detailed features through its encoder-decoder structure and skip connections. However, this architecture is very complex [20].

3. Introduction to PSO

PSO is an ingenious optimization algorithm rooted in the collective behavior of social organisms, particularly inspired by the coordinated movements observed in flocks of birds. The underlying principle of PSO is to emulate the collaborative and adaptive nature of these social entities, where individuals work together to navigate and discover optimal paths or locations. In the realm of PSO, the optimization process unfolds within a multi-dimensional search space. Each participant in the optimization, referred to as a "particle," embodies an individual entity within the swarm. The particle's position in this multi-dimensional space represents a potential solution to the optimization problem at hand. The movement of particles within the space is orchestrated by two fundamental components: personal experience and social interactions.

Personal experience involves the particle's historical knowledge of its own best-performing positions. The algorithm assigns a fitness value to each particle based on the quality of its position in the search space. Social interactions come into play as particles exchange information about their optimal positions with neighboring particles. This collaborative information sharing enables the swarm to

collectively converge toward promising regions in the search space.

The iterative nature of PSO allows particles to dynamically adjust their positions over successive generations, optimizing their solutions based on the feedback received from personal experiences and shared insights within the swarm. As a result, PSO demonstrates a remarkable ability to explore diverse regions of the solution space efficiently.

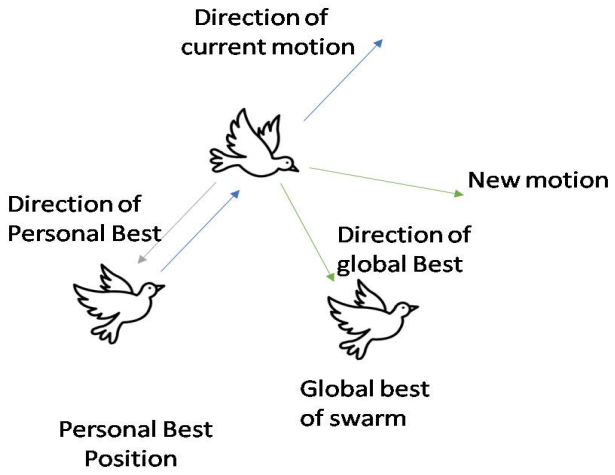

Figure 1. Dynamics Unveiled: Illustrating Particle Velocity Update and Movement in a Schematic Diagram

In the context of optimization, PSO represents each potential solution as a "particle" in a multi-dimensional search space. The movement of each particle is governed by two key components: personal experience and social interactions. Mathematically, the position of a particle i at iteration t is represented by $X_i^t = (x_{i1}^t, x_{i2}^t, ..., x_{iD}^t)$ where D is the dimensionality. The update equation for the velocity of each particle is given by:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (Pbest_i - X_i^t) + c_2 r_2 (Gbest - X_i^t) \quad (1)$$

Here, V_i^t is the velocity of particle i at iteration t, ω is the inertia weight, c_1 and c_2 are acceleration constants, r_1 and r_2 are random values between 0 and 1, $Pbest_i$ is the best position the particle i has visited so far, and Gbest is the best position found by any particle in the swarm.

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

The algorithm iteratively refines the positions and velocities of particles, allowing the swarm to collectively converge toward optimal solutions. The procedure persists until a termination condition is satisfied, such as reaching a specified maximum number of iterations or attaining a satisfactory solution.

4. Proposed Method

The block diagram delineating the architecture of the proposed guided L_0 smoothing based edge detection technique is illustrated in Figure 1. This innovative approach is designed to enhance edge detection through a structured sequence of operations. The method revolves around three fundamental components within the guided L_0 smoothing framework: the L_0 smoothing filter, L_0 gradient minimization, and guided filtering.

L0 Smoothing Filter: The initial phase of the guided L_0 smoothing method involves the application of the L_0 smoothing filter to the input image. This filter plays a pivotal role in reducing noise and enhancing the overall smoothness of the image, contributing to a more robust foundation for subsequent processing steps.

L0 Gradient Minimization: Following the L_0 smoothing stage, the method incorporates L_0 gradient minimization as a critical step. This process focuses on minimizing gradients within the image, aiming to preserve edges while attenuating unnecessary details. The emphasis on gradient minimization contributes to the extraction of salient features, laying the groundwork for precise edge detection.

Guided Filtering: The guided filtering serves as a key element in the guided L_0 smoothing method. This stage refines the image by leveraging guidance from an additional reference image, enhancing the overall coherency and preserving important edge information. The guided filter aids in achieving a fine balance between smoothing and edge preservation.

PSO Based Edge Detection: Subsequently, the processed image undergoes a progressive stage where it is passed through a PSO based edge detection mechanism. This computational technique harnesses the principles of swarm intelligence to identify and highlight edges within the image. The PSO based edge detection step contributes to the finalization of the edge map. The collaborative integration of these components within the guided L_0 smoothing framework, culminating in the PSO based edge detection, results in a comprehensive technique for robust and accurate edge detection. Figure 1 serves as a visual guide, encapsulating the sequential flow of operations and the interplay of components within this sophisticated method. This methodology aims to enhance the quality and precision of edge detection in the realm of image processing.

4.1 L_0 Guided Image Filtering

Utilizing smoothness noise in an image can be mitigated through image smoothing, which reduces amplitude variation and aids in diminishing undesired edges. However, this process may inadvertently suppress some prominent edges. Various techniques have been proposed over time to achieve edge preserving smoothing [21, 22].

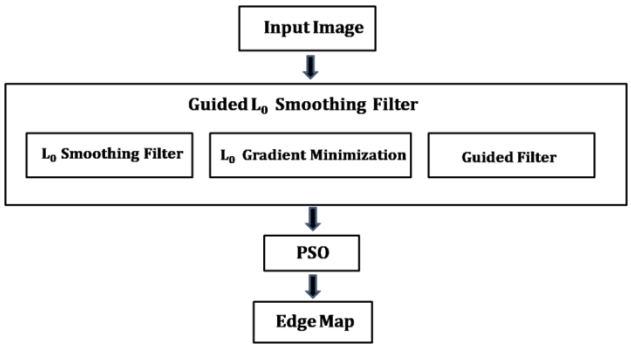

Figure 2. Block diagram for Guided L_0 smoothing filtering and PSO based edge detection process

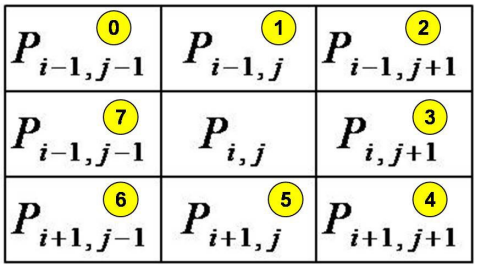

Figure 3. Representation of 3×3 image block with pixel coding

In a similar vein, guided image filtering was introduced [21]. Another method for edge preservation is the L_0 smoothing filter [22]. A more refined filter, namely the Guided L_0 smoothing filter, was proposed by combining these two methods [22] (Figure 2). For a clearer illustration, considering a 3×3 block image with the central pixel positioned at (i, j), denoted as $P_{i,j}$,, and the values and positions of the surrounding pixels are depicted in Figure 3.

4.2 L_0 smoothing filter

The number of the absolute difference in intensity values between neighbouring pixels can be expressed mathematically as

$$pg(I) = \#\left\{i, j \middle| \left|P_{i,j} - P_{i-1,j}\right| \neq 0\right\} \quad (3)$$

The L_0 norm of the gradient is the measure of the disparity in pixel values. The non-zero-pixel difference is denoted by Δ. In terms of pixel gradient (pg) for input pixel (P_i) and its smoothened version as P_i^* the objective function can be written as

$$\min_I \sum_i \left(P_i - P_i^*\right)^2 \quad \text{s.t.}, pg(I) = \Delta \quad (4)$$

The, more appropriate objective function considering weight controlling parameter (γ) would be

$$\min_P \sum_i \left(P_i - P_i^*\right)^2 + \gamma\, pg(I) \quad (5)$$

4.3 L_0 gradient minimization

The L_0 norm, when strategically optimized, offers a tailored approach to generate an output image characterized by piecewise constancy through L_0 gradient minimization [18]. This optimization technique proves particularly effective in accentuating prominent edges by augmenting the abruptness of the transitions while simultaneously mitigating the influence of low amplitude structures. The formulation expressed in Equation 5, can be alternatively represented as follows:

$$\min_I \left\|P_i - P_i^*\right\|_2^2 + \gamma \left\|\nabla P_i\right\|_0 \quad (6)$$

The notation ∇P_i signifies the gradient of the function P_i. To handle the term $\left\|\nabla P_i\right\|_0$ effectively, an auxiliary variable ψ is introduced. Consequently, equation (6) can be transformed into the following minimization expression. This modification allows for a more flexible and nuanced optimization approach, facilitating a comprehensive treatment of the associated variables and enabling a refined exploration of the solution space. The introduction of the auxiliary variable ω serves to enhance the adaptability of the minimization process and contributes to a more intricate and expressive formulation of the mathematical model.

$$\min_{P,\psi} \left\|P_i - P_i^*\right\|_2^2 + \beta \left\|\psi - \nabla P_i\right\|_0 + \gamma \left\|\psi\right\|_0 \quad (7)$$

Here, β is controlling parameter and the degree of smoothness is handled by γ.

4.4 Guided filter

The Guided smoothing filter emerges as a specialized edge preserving smoothing filter renowned for its exceptional performance, particularly in the proximity of edges [21]. One noteworthy aspect of its versatility lies in the ability to utilize an alternative reference image (I_g) to fulfill the guidance image prerequisite. This flexibility allows for customization based on specific requirements or considerations, enhancing the adaptability of the filtering process.

To illustrate this concept further, let's refer to Figure 3. Within this context, we can focus on a chosen window ω_u. The number of pixels in the chosen window is determined by a specific mathematical expression (eqn. 8), denoting the spatial extent over which the guided filter operates. This spatial consideration is crucial for understanding the local context in which the filtering takes place, contributing to the filter's effectiveness in preserving edges while smoothing the surrounding regions.

$$|P| = \sum_{\omega_u} P_{i,j} \,. \tag{8}$$

The resulting image, guided by the specified parameters, is represented as follows:

$$I_{out} = \frac{1}{|P|} \sum_{P_u | i \in \omega_u} \left(a_u I_i + b_u \right), \tag{9}$$

In the preceding equation, the values for a_u and b_u are derived as follows:

$$a_u = \frac{\frac{1}{|P|} \sum_{P_u | i \in \omega_u} I_g I_i - E\left[I_g\right] E\left[I_i\right]}{\sigma_u^2 + \varepsilon}, \tag{10}$$

$$b_u = E\left[I_i\right] - a_u \left[I_g\right]$$

In this context, the parameter ε serves as a regularization parameter.

4.5 Guided L_0 smoothing filter

The concept of L_0 gradient minimization, stands out as a technique that enhances image sharpness while preserving prominent edges [23]. In a complementary vein, X. Ding et al. proposed a novel extension known as the guided L_0 smoothing filter [24]. This innovative method integrates the principles of guided filtering with L_0 smoothing, resulting in an advanced image processing approach that is adept at simultaneously sharpening the image and preserving dominant edges. The guided L_0 smoothing filter, as delineated by Ding and collaborators, represents a synergistic combination of edge preserving qualities and smoothing efficacy, contributing to a refined and versatile tool for image enhancement [24].

Utilizing the expression I^u, the parameter Ω^u is optimized through the following procedure:

$$\min_{\delta^u} \beta^u \left\| \nabla I^u - \Omega^u \right\|_2^2 + \lambda \left\| \Omega^u \right\|_0 \quad (u = 1,2,3....) \tag{11}$$

In the given equation, the term Ω^u is defined as follows:

$$\Omega^u = \begin{cases} 0 & \nabla I^u \le \dfrac{\lambda}{\beta} \\ \nabla I^u & others \end{cases} \tag{12}$$

Now that both Ω^u and I^u are known, the next step involves evaluating I^{u+1} using:

$$\min_{I^{u+1}} \left\| I^{u+1} - I^* \right\|_2^2 + \beta^u \left\| \Omega^u - \nabla I^{u+1} \right\|_2^2 \tag{13}$$

Inserting eqn. 12 in eqn. 13 we get,

$$\min_{I^{u+1}} \left\| I^{u+1} - I^* \right\|_2^2 + \beta^u \left\| \nabla I^{u+1} - O.* \nabla I^u \right\|_2^2 \tag{14}$$

where,

$$O = \begin{cases} 0 & \Omega^u = 0 \\ 1 & \Omega^u \neq 0 \end{cases} \qquad (u = 1,2,3.....) \tag{15}$$

The presented equation constitutes a convex optimization problem, and its solution can be obtained through the application of least squares and the Fast Fourier Transform (FFT) technique. A comprehensive elucidation of the approach is provided in [24], offering detailed insights into the methodology and the intricacies of employing least squares and FFT for the resolution of the convex optimization problem at hand.

$$I^{u+1} = \mathrm{T}_r^{-1} \left(\frac{T_r\left(I^*\right) + \beta \left(T_r\left(\partial_x^T\right) T_r\left(\Omega_x^u\right) + T_r\left(\partial_y^T\right) T_r\left(\Omega_y^u\right) \right)}{T_r(1) + \beta \left(T_r\left(\partial_x^T\right) T_r\left(\partial_x\right) + T_r\left(\partial_y^T\right) T_r\left(\partial_y\right) \right)} \right) \tag{16}$$

In the aforementioned expression, the inverse Fast Fourier Transform is symbolically represented as T_r^{-1}, while the difference operators in the 'x' and 'y' directions are denoted by ∂x and ∂y respectively. These operators serve to compute partial derivatives with respect to the respective spatial coordinates, contributing to the overall formulation of the mathematical expression.

Finally, using, Ω^u and s^u we obtain s^{u+1} as

$$\begin{cases} \min\limits_{s^{u+1}} \left\| s^{u+1} - s^* \right\|_2^2 + \beta^u \left\| \nabla s^{u+1} - O.* \nabla s^u \right\|_2^2 \\ O = \begin{cases} 0 & \Omega^u = 0 \\ 1 & \Omega^u \neq 0 \end{cases} \qquad (u = 1,2,3.....) \end{cases} \tag{17}$$

The solution of equation (17) is:

$$s^{k+1} = \mathrm{T}_r^{-1} \left(\frac{T_r\left(s^*\right) + \beta \left(T_r\left(\partial_x^T\right) T_r\left(O.* \nabla s_x^k\right) + T_r\left(\partial_y^T\right) T_r\left(O.* \nabla s_y^k\right) \right)}{T_r(1) + \beta \left(T_r\left(\partial_x^T\right) T_r\left(\partial_x\right) + T_r\left(\partial_y^T\right) T_r\left(\partial_y\right) \right)} \right) \tag{18}$$

Algorithm:

Input: Image s^*, guided image I^*, parameters λ, $\beta_0, \beta_{\max}$

Get: $I^1 \leftarrow I^*, s^1 \leftarrow s^*, \beta^1 \leftarrow \beta_0, u \leftarrow 1,$

do:

Using I^u, get Ω^u for in (11);

Using I^u and Ω^u, obtain I^{u+1} in (13);

Using s^u and Ω^u, obtain s^{u+1} in (17);

$\beta \leftarrow u\beta, u++;$

Until $\beta_0 > \beta_{\max}$

Get smoothened image: s.

Finally, the edge sharpened image can be obtained as

$$s_S = s + \chi[s - \underbrace{s \otimes G_{L_0}}_{\text{Smoothening}}] \tag{19}$$

On the guided edge sharpened image with predefined threshold 'T', the homogeneity is calculated as [25] (refer Figure 4)

$$H_p = \begin{cases} \max\left|P_{i,j} \otimes G_{L_0} - P_{m,n} \otimes G_{L_0}\right|, m \in \{i-1, i, i+1\}, n \in \{j-1, j, j+1\} & \text{if } > \mathrm{T} \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

The partial derivatives of the pixels in 'x' and 'y' directions can be evaluated as

$$\Delta_x P(s) = P(i+1, j) \otimes G_{L_0} - P(i, j) \otimes G_{L_0}$$
$$\Delta_y P(s) = P(i, j+1) \otimes G_{L_0} - P(i, j) \otimes G_{L_0} \quad (21)$$

The gradient is obtained as

$$\nabla\left(P(i, j, s)\right) = \sqrt{\left(\Delta_x P(s)\right)^2 + \left(\Delta_y P(s)\right)^2} \quad (22)$$

4.6 PSO Based Edge Detection

Consider a scenario where a pixel displays a non-zero gradient, and envision this pixel as a prospective starting point for the initial position of a particle. In this context, pixels with non-zero gradients act as the instigators for the PSO algorithm, initiating a dynamic optimization process that seeks to refine and enhance the overall array of pixels for a specified objective or criterion. The crucial stages in PSO based edge detection encompass:

1. Defining the procedure for encoding a particle;
2. formulating a fitness function and
3. Appraising the fitness of the arc.

This iterative process effectively refines the selection and placement of particles, optimizing the detection of edges within the image by iteratively adjusting their positions based on the fitness evaluations of the corresponding arcs.

4.6.1 Encoding of Particles

Every particle is given a numerical assignment within the range of 1 to 8, signifying the specific direction of movement from one pixel to the adjacent pixel in the computational process. This assignment serves as a directional indicator for the particle's traversal across the image grid, contributing to the overall dynamics of the algorithm. Consequently, a unique encoding $\langle d_1, d_2, \ldots\ldots d_{\max} \rangle$ is assigned to each particle in the population.

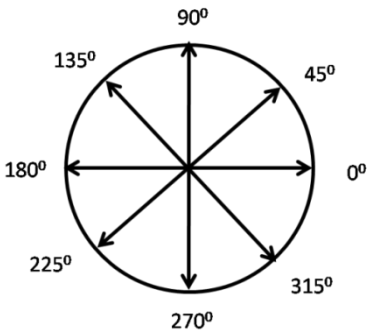

Figure 4. Directions of Particle Movement (Total of 8 Directions)

In Figure 4, a depiction of the eight movement directions is provided. To elucidate, consider the starting point at pixel A; the ensuing movement proceeds in the direction of 270^0 degrees, as indicated in Figures 5. Referring to Figure 3, the encoding for pixel movement from position $P_{i,j}$ to $P_{i+1,j}$ is specifically represented as 5. Extending this principle, the systematic encoding of the entire arc is conducted by aligning the movement direction of the arc with the corresponding pixel coding (Figure 6). This method ensures a comprehensive and consistent encoding process for the entire trajectory of the arc.

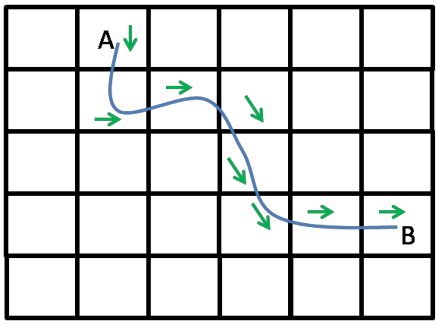

Figure 5. Illustration of a continuous edge arc between two points (A, B)

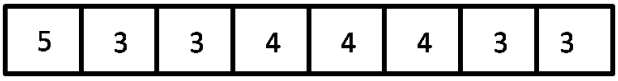

Figure 6. Illustration of Encoding Scheme

4.6.2 Arc characteristics

In the realm of image processing, it's observed that pixels situated along a given edge consistently exhibit a relatively uniform intensity. This commonality in intensity levels along an edge forms the basis for leveraging two key characteristics when analyzing arcs within an image: homogeneity and uniformity.

Homogeneity (H_f), in this context, refers to the degree of similarity or consistency in pixel intensities along an arc (f). When an arc possesses high homogeneity, it implies that the pixel values along the corresponding edge are relatively uniform, contributing to a cohesive and visually continuous appearance. This characteristic is instrumental in identifying and emphasizing regions where pixel intensity remains consistent signifying a clear and discernible edge. The H_f in horizontal and vertical direction is given by

$$H_f = \sum_{P_k \in f} H_{P_k} \quad (23)$$

In other directions (diagonal) is

$$H_f = \frac{1}{\sqrt{2}} \sum_{P_k \in f} H_{P_k} \quad (24)$$

Uniformity, on the other hand, addresses the even distribution and regularity of pixel intensities along the arc. An arc demonstrating high uniformity implies that the variations in pixel intensities are consistent and well distributed, reinforcing the visual continuity of the edge. The uniformity attribute is crucial for ensuring that the detected edges exhibit a smooth and aesthetically pleasing quality, contributing to the overall clarity and perceptual coherence of the image.
The H_f in horizontal and vertical direction is given by

$$U_f = \sum_{k=1}^{A_L-1} \left| I_{P_{k+1}} - I_P \right| \tag{25}$$

In other directions (diagonal) is

$$U_f = \frac{1}{\sqrt{2}} \sum_{k=1}^{A_L-1} \left| I_{P_{k+1}} - I_P \right| \tag{26}$$

4.6.3 PSO algorithm's objective function

An objective function is devised in this context, embodying the twin qualities of homogeneity and uniformity. It is in horizontal and vertical direction is given by

$$F_f = \begin{cases} \dfrac{\left(H_f - U_f\right)}{H_f} & if\ F_f \geq \left(\dfrac{1}{1+e^{-H_f}}\right) \\ -\infty & otherise \end{cases} \tag{27}$$

In other directions (diagonal) is

$$F_f = \begin{cases} \dfrac{\left(H_f - U_f\right)}{\sqrt{2}H_f} & if\ F_f \geq \left(\dfrac{1}{1+e^{-H_f}}\right) \\ -\infty & otherise \end{cases} \tag{28}$$

5. Results

The validity of the proposed method is evaluated using computer simulation. Simulation is done in MATLAB(R) 2015 (a) with RAM of 4 GB. Both qualitative and quantative methods are used for the comparisons of results. The list of parameters and their value is shown in Table 1.

Table 1. List of Parameters and Value

Parameters	Values
ε	0.01
χ	5%, 10% and 20%
Population Size	20
Directions	8
Iterations	50
Database	BSD, Lena Image

Using the Berkeley Segmentation Dataset (BSD) alongside the Lena image offers a robust approach for evaluating edge detection algorithms. The BSD dataset provides a diverse collection of images with varying complexities and textures, allowing for a thorough assessment of an algorithm's performance across different real-world scenarios. It includes ground truth edge maps that are crucial for quantitative evaluation of edge detection accuracy. In contrast, the Lena image, with its well-defined edges and smooth gradients, serves as a controlled benchmark for visual inspection and demonstration purposes. By combining the challenging conditions of BSD with the clarity of Lena, researchers can achieve a comprehensive evaluation of edge detection methods, balancing both real-world applicability and idealized test cases.

Performance Metrics:

Pratt's Figure of Merit (FoM)
The Pratt's FoM evaluates edge location exactness in edge detected image in comparison to ground truth image, by measuring the displacement of edge points that are detected from an ideal edge. The FoM is characterized by

$$FoM = \frac{1}{\max(I_{gt}, I_{ed})} \sum_{i=1}^{I_{ed}} \frac{1}{1+\mu d^2(p, I_{gt})} \tag{29}$$

Here,
I_{gt} = ideal edge points (ground truth)
I_{ed} = edge points detected
d = displacement of detected edges from ideal edges
μ = scaling constant.

F-Score
The F-Score is given by

$$F = \frac{2TP}{2TP + FP + FN} \tag{30}$$

Where TP is true positive and FP is false positive and FN is false negative.

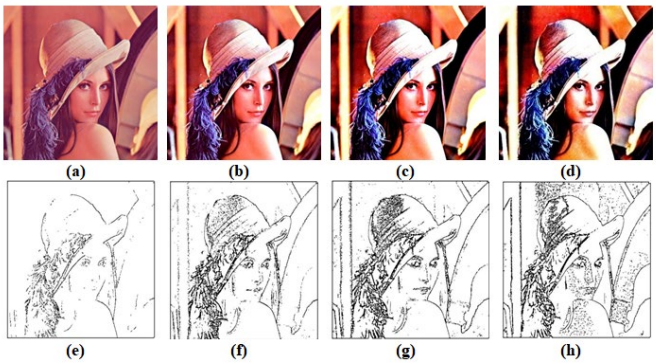

Figure 7. (a) Original Lenna Image (b) 5% Sharpened Image (c) 10% Sharpened Image (d) 20% Sharpened Image (e) Detached edges original image (f) Detached edges 5% sharpened image (g) Detached edges 10% sharpened image (h) Detached edges 20% sharpened image

Figure 7 presents a detailed exploration of the effects of different sharpening percentages on the well-known Lenna image. The original Lenna image acts as a reference point in panel (a), while subsequent panels (b) through (d) showcase the image sharpened at 5%, 10%, and 20%, respectively. These variations in sharpening percentages allow for a visual understanding of how enhancing image details impacts the overall clarity and sharpness. The detached edges extracted from both the original and sharpened images are highlighted in panels (e) through (h), providing a closer look at the accentuated features and boundaries. Importantly, the 10% sharpened image, as illustrated in panel (g), demonstrates a judicious equilibrium between heightened details and the mitigation of undesirable artifacts, culminating in an aesthetically pleasing result. It is noteworthy that in the original image, certain edges are erroneously dismissed, indicating a potential limitation in edge detection. This underscores the delicate trade-off in image sharpening, where excessive enhancement may lead to the exclusion of genuine edges. On the contrary, the 20% sharpened image, as depicted in panel (d), reveals a significant increase in noise. This heightened noise level becomes pronounced as a consequence of the intensified sharpening process, compromising the visual quality of the image by introducing unwanted artifacts. The observed trade-offs between sharpening levels shed light on the intricate balance required to achieve optimal image enhancement, where a 10% sharpening ratio emerges as a favorable compromise, striking a harmonious blend between heightened details and minimized artifacts.

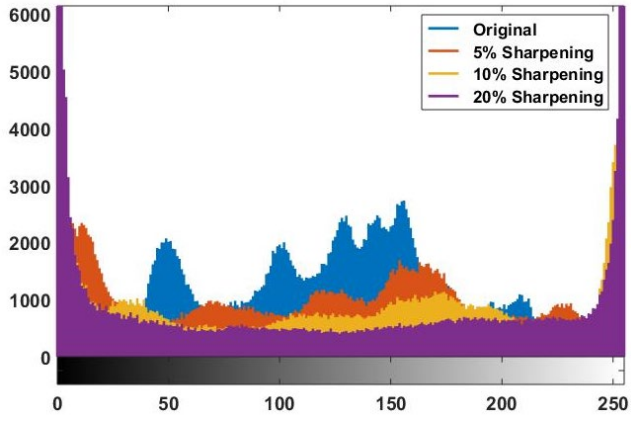

Figure 8. Histogram of Lenna image under various sharpening levels

In Figure 8, we delve into a comprehensive analysis of the Lenna image's histogram across different sharpening levels. The histogram serves as a graphical representation of the pixel intensity distribution within the image, providing valuable insights into the impact of sharpening on the overall tonal characteristics. Each subfigure within Figure 8 corresponds to a distinct sharpening level, ranging from subtle to more pronounced enhancements.

As we examine the histograms, it becomes apparent how the pixel intensity values are redistributed with varying degrees of sharpening. Sharpening, by nature, tends to accentuate the contrast and fine details in an image. In the context of the Lenna image, the histograms depict how this sharpening process influences the spread of pixel intensities.

Notably, as the sharpening levels increase, there is a discernible shift in the histogram towards higher intensity values. This shift signifies the augmentation of image details and the creation of more distinct tonal variations. However, it's crucial to observe whether this augmentation remains within a visually acceptable range or if it leads to undesirable artifacts such as clipping or saturation.

Furthermore, by examining the histogram patterns across different sharpening levels, we gain a nuanced understanding of how pixel intensities are modified, helping us assess the overall impact on image tonality. This detailed histogram analysis in Figure 8 contributes to a comprehensive evaluation of the consequences of sharpening, allowing for informed decisions about the appropriate level of enhancement for achieving the desired visual outcomes.

In Figure 9, an insightful comparison of state-of-the-art edge detection methods is presented, shedding light on the strengths and limitations of each technique. The Canny edge detection method is observed to be plagued by false edges, introducing inaccuracies in the delineation of edges.

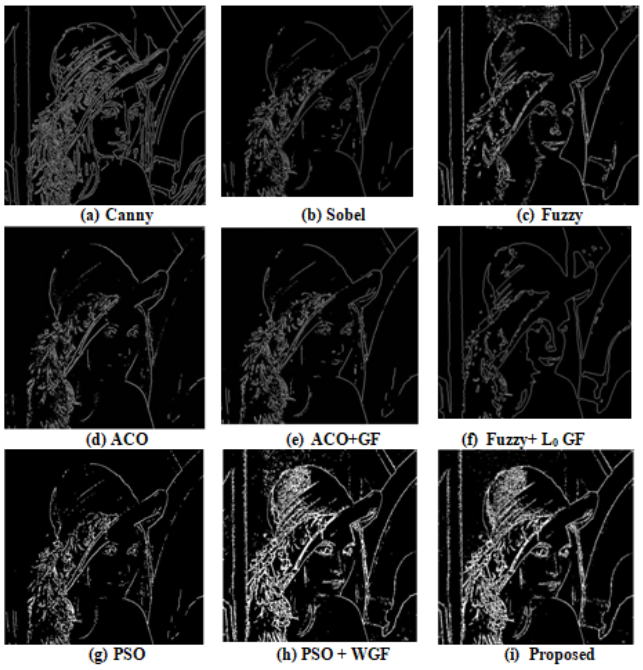

Figure 9. Comparison of the State-of-the-art edge detection methods

Meanwhile, the Sobel method exhibits a tendency to reject correct edges, potentially leading to an underestimation of the true edge structure. The Fuzzy method, as depicted in the above-hat area, demonstrates vulnerability to noise, highlighting its susceptibility to inaccuracies in edge detection. The ACO method not only rejects correct edges but also suffers from broken edges, compromising the overall

integrity of the detected edges. The introduction of ACO+ GF addresses the issue of broken edges, providing a notable improvement in edge detection accuracy. The Fuzzy+ L_0 Guided Filtering method emerges as a superior option among the discussed techniques, showcasing improved performance. However, it is noted for incorrectly detecting face boundaries, indicating a potential limitation in accurately identifying specific features. Comparatively, the PSO method outperforms ACO but exhibits edges that are less sharp, potentially impacting the precision of edge detection. The addition of Weighted Guided Filtering (WGF) to PSO, denoted as PSO+ WGF, proves to be an enhancement over the standalone PSO, showcasing improved results. Notably, the proposed method surpasses the aforementioned techniques in terms of edge detection. The edges are reasonably well connected, demonstrating a favorable balance between sensitivity and specificity. Moreover, the proposed method exhibits a commendable tolerance to noise, further solidifying its efficacy in real-world scenarios. This comparative analysis in Figure 9 underscores the significance of the proposed method as a promising advancement in edge detection methodologies, offering improved edge connectivity and noise tolerance compared to existing state-of-the-art techniques.

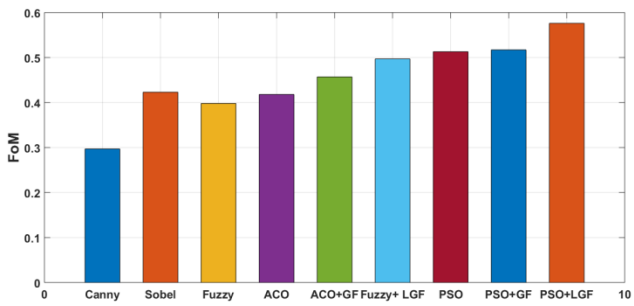

Figure 10. Comparison of FoM for the state-of-the-art methods

The Figure 10 presents a detailed comparison of various image processing methods, each accompanied by its corresponding Figure of Merit (FoM) value. The Canny edge detector, known for its accuracy in identifying edges, yielded a FoM value of 0.297. The Sobel operator, utilizing convolution with Sobel kernels for edge detection, demonstrated a FoM value of 0.423, reflecting its computational efficiency. Fuzzy image processing, incorporating fuzzy logic to address uncertainty in image data, garnered a FoM value of 0.398. ACO, a nature-inspired optimization algorithm, achieved a FoM value of 0.418 when applied to image processing tasks. ACO combined with guided filtering resulted in ACO+ Guided Filtering, yielding an improved FoM value of 0.457. The integration of fuzzy processing with L_0 guided filtering, termed Fuzzy + L_0 Guided Filtering, produced a FoM value of 0.497, showcasing enhanced performance. PSO, another nature-inspired algorithm, achieved a FoM value of 0.513. Variants of PSO with additional sharpening effects demonstrated varying FoM values: PSO+ 5% sharpening (0.517), PSO+ 10% sharpening (0.511), the proposed method, PSO+ L_0 Guided Filtering (5% sharpening) (0.521), and PSO+ L_0 Guided Filtering (10% sharpening) with the highest FoM value of 0.576. The table thus provides a comprehensive overview of the methods and their FoM values, facilitating a comparative analysis of their performance in the specific image processing context considered in the study.

The Figure 11 offers a comprehensive comparison of various image processing methods, each accompanied by its corresponding F score, a metric commonly used in classification tasks that combines precision and recall. The Canny edge detector, known for its accuracy in detecting edges, achieved an F score of 0.49. The Sobel operator, employing convolution with Sobel kernels for edge detection, demonstrated an F score of 0.40, reflecting its computational efficiency in this specific application. Fuzzy image processing, which incorporates fuzzy logic to handle uncertainty in image data, exhibited a notable F score of 0.64, indicating its effectiveness in the given context. ACO, a nature-inspired optimization algorithm applied to image processing, yielded a higher F score of 0.72, showcasing improved performance.

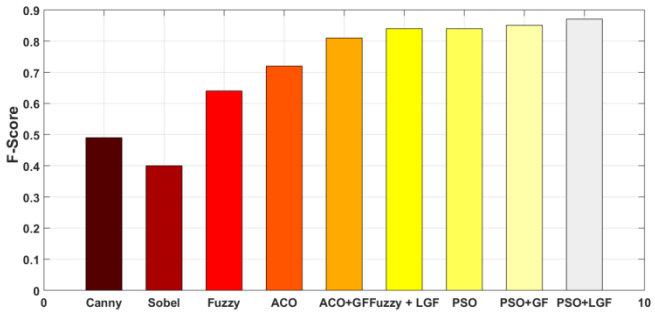

Figure 11. Comparison of F-Score for the state-of-the-art methods

The combination of ACO with guided filtering, denoted as ACO+ Guided Filtering, further enhanced the F score to 0.81. The fusion of fuzzy processing with L_0 guided filtering, termed Fuzzy+ L_0 Guided Filtering, resulted in a higher F score of 0.84, suggesting improved accuracy and recall. PSO, another nature-inspired algorithm applied to image processing, achieved a commendable F score of 0.84. Variants of PSO with additional sharpening effects demonstrated incremental improvements in F scores: PSO+ 5% sharpening (0.843), PSO+ 10% sharpening (0.851), PSO+ L_0 Guided Filtering (5% sharpening) (0.867), and PSO+ L_0 Guided Filtering (10% sharpening) with the highest F score of 0.871. In summary, the table provides valuable insights into the performance of each image processing method based on the F score, enabling a comparative analysis of their accuracy and recall in the specific image processing context considered in the study.

Figure 12. BSD Butterfly image

In Figure 12, the BSD butterfly image is presented, which is utilized as a benchmark for comparing various edge detection algorithms. This image, sourced from the BSD, features a complex and richly detailed scene with intricate edges and textures. By using this image, we are able to evaluate and compare the performance of different edge detection methods under controlled conditions, assessing their ability to accurately identify and delineate edges in a diverse and challenging context

Figure 13. Comparison various edge detection method for BSD image dataset

In Figure 13, various edge detection methods are compared to evaluate their performance. Figure 13(a) presents the ground truth image, which serves as a reference for assessing the accuracy of the different edge detection techniques. Figure 13(b) shows the results of the Sobel edge detection method. The Sobel detector is known for its simplicity and computational efficiency, making it effective for detecting edges with significant contrast. However, it is highly sensitive to noise, which can lead to the detection of false edges, and it often struggles with accurate edge localization and thinning. In Figure 13(c), the Canny edge detection method is displayed. Canny is renowned for its robustness against noise and its precise edge localization, thanks to its use of Gaussian smoothing, gradient calculation, non-maximum suppression, and hysteresis thresholding. Despite these strengths, it has higher computational complexity and is sensitive to parameter choices, which can affect edge detection performance, especially in low-contrast regions.
Figure 13(d) illustrates the results obtained using the PSO method. PSO can optimize edge detection parameters or criteria, potentially enhancing the results compared to traditional methods. However, it can suffer from convergence issues, leading to suboptimal solutions, and is computationally expensive, requiring careful tuning of its parameters. Figure 13(e) presents the results of combining PSO with a WGF. This combination aims to leverage the strengths of both PSO and WGF, improving edge detection by refining edge details through weighted gradients. While this approach can enhance edge detection, it introduces added complexity and may result in slower processing speeds, along with challenges in parameter tuning. Finally, Figure 13(f) shows the results of the proposed method. This new approach seeks to overcome the limitations of existing techniques by integrating innovative methods or combining multiple approaches to improve edge detection performance. Although promising, the proposed method needs thorough validation and benchmarking to confirm its advantages and address any potential implementation challenges.

Table 2. Comparison with state-of-the-art DNN methods

Method/Reference	F-Score
Msmsfnet [17]	0.767
Edter [18]	0.84
Refined Edges [19]	0.853
U-Net [20]	0.861
Proposed	0.871

Table 2 provides a comparative analysis of various state-of-the-art deep neural network (DNN) methods for edge detection, focusing on their F-scores, which measure the balance between precision and recall. The Msmsfnet [17] achieves an F-score of 0.767, reflecting its capability in detecting edges but indicating room for improvement. Edter [18] improves upon this with an F-score of 0.84, showing better performance in capturing accurate edges. Refined Edges [19] further advances edge detection accuracy with an F-score of 0.853, offering a notable improvement in precision and recall. The U-Net model [20] surpasses previous methods with an F-score of 0.861, demonstrating its robust performance in detailed edge detection. The proposed method leads the comparison with an F-score of 0.871, highlighting its superior ability to detect fine and accurate edges, thus setting a new benchmark in the field.

5. Conclusions

In conclusion, this paper introduces a novel approach to edge detection by integrating BPSO with L_0 Guided Filtering. The collaborative strategy aims to tackle the intricate task of precise edge detection in noisy and complex images by capitalizing on the strengths of both BPSO and L0 guided filtering. The proposed methodology initiates with the BPSO algorithm's initialization, utilizing binary particles to navigate the solution space and optimize crucial parameters for effective edge detection. These optimized parameters are then applied within the L_0 guided filtering framework, a sophisticated edge-preserving filter renowned for its capacity to retain fine details while effectively mitigating noise. The synergy between BPSO and L_0 guided filtering exhibits enhanced adaptability to diverse image characteristics, thereby fortifying the overall robustness of edge detection. The binary nature of BPSO expedites exploration of the solution space, leading to quicker convergence to optimal parameters. Simultaneously, L_0 guided filtering ensures edge-preserving smoothing, contributing to the suppression of undesired artifacts. Experimental assessments conducted on benchmark datasets underscore the efficacy of the proposed method compared to traditional edge detection techniques. The outcomes demonstrate superior edge localization and reduced sensitivity to noise, emphasizing the practical potential of BPSO-based Edge Detection under L_0 Guided Filtering in real-world applications. This presented approach represents a valuable contribution to the progression of edge detection methodologies, showcasing its capacity to enhance the performance of computer vision systems across various domains. The innovative fusion of BPSO and L_0 guided filtering offers a promising avenue for advancing edge detection techniques, opening new possibilities for more robust and accurate computer vision applications. In the proposed method parameter tuning is complex, requiring careful adjustment of numerous settings for optimal performance. Additionally, while the approach improves edge detection, it may still be sensitive to noise, necessitating effective noise reduction strategies.

References

[1] Torre, Vincent, and Tomaso A. Poggio. "On edge detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 (1986): 147-163.

[2] Jeong, Chiyoon, Hyun S. Yang, and KyeongDeok Moon. "A novel approach for detecting the horizon using a convolutional neural network and multi-scale edge detection." *Multidimensional Systems and Signal Processing* 30, no. 3 (2019): 1187-1204.

[3] Nezamabadi-pour, Hossein, Majid Rostami-Shahrbabaki, and Malihe Maghfoori-Farsangi. "Binary particle swarm optimization: challenges and new solutions." *CSI J Comput Sci Eng* 6, no. 1 (2008): 21-32.

[4] Wang, Dongshu, Dapei Tan, and Lei Liu. "Particle swarm optimization algorithm: an overview." *Soft computing* 22 (2018): 387-408.

[5] Ranjan, Rakesh, and Vinay Avasthi. "Edge Detection Using Guided Sobel Image Filtering." *Wireless Personal Communications* 132, no. 1 (2023): 651-677.

[6] Canny J. 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence. (6):679 98.

[7] Vincent, O. Rebecca, and Olusegun Folorunso. "A descriptive algorithm for Sobel image edge detection." In *Proceedings of informing science & IT education conference (InSITE)*, vol. 40, pp. 97-107. 2009.

[8] Ari, Samit, Dipak Kumar Ghosh, and Prashant Kumar Mohanty. "Edge detection using ACO and F ratio." *Signal, Image and Video Processing* 8, no. 4 (2014): 625 634.

[9] Seng, N. H., Z. Samad, and N. M. Nor. "A 3 Pixel Fuzzy Mask for Edge Detection." In IOP Conference Series: Materials Science and Engineering, vol. 530, no. 1, p. 012023. IOP Publishing, 2019.

[10] Kumar, Akshi, and Sahil Raheja. "Edge detection using guided image filtering and enhanced ant colony optimization." *Procedia Computer Science* 173 (2020): 8 17.

[11] Kumar, Akshi, and Sahil Raheja. "Edge detection using guided image filtering and ant colony optimization." In *The international conference on recent innovations in computing*, pp. 319 330. Springer, Singapore, 2020.

[12] Kumar, Akshi, and Sahil Raheja. "Edge Detection in Digital Images Using Guided L_0 Smoothen Filter and Fuzzy Logic." *Wireless Personal Communications* 121, no. 4 (2021): 2989 3007.

[13] Raheja, Sahil, and Akshi Kumar. "Edge detection based on type 1 fuzzy logic and guided smoothening." *Evolving Systems* 12, no. 2 (2021): 447 462.

[14] Eberhart, Russell, and James Kennedy. "Particle swarm optimization." In Proceedings of the IEEE international conference on neural networks, vol. 4, pp. 1942 1948. 1995.

[15] Chen, Dongyue, Ting Zhou, and Xiaosheng Yu. "A new method of edge detection based on PSO." In International Symposium on Neural Networks, pp. 239 246. Springer, Berlin, Heidelberg, 2012.

[16] Verma, Ankush, Namrata Dhanda, and Vibhash Yadav. "Binary particle swarm optimization based edge detection under weighted image sharpening filter." *International Journal of Information Technology* 15, no. 1 (2023): 289-299.

[17] Liu, Chenguang, Chisheng Wang, Feifei Dong, Xin Su, Chuanhua Zhu, Dejin Zhang, and Qingquan Li. "Msmsfnet: a multi-stream and multi-scale fusion net for edge detection." *arXiv preprint arXiv:2404.04856* (2024).

[18] Pu, Mengyang, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. "Edter: Edge detection with transformer." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1402-1412. 2022.

[19] Elharrouss, Omar, Youssef Hmamouche, Assia Kamal Idrissi, Btissam El Khamlichi, and Amal El Fallah-Seghrouchni. "Refined edge detection with cascaded and high-resolution convolutional network." *Pattern Recognition* 138 (2023): 109361.

[20] An, Ying, Junfeng Jing, Xuewei Li, Jiaqi Zhang, and Junmin Bao. "An exclusive U-net for fine and crisp edge detection." *Multimedia Tools and Applications* 83, no. 18 (2024): 54657-54672.

[21] He, Kaiming, Jian Sun, and Xiaoou Tang. "Guided image filtering." In European conference on computer vision, pp. 1 14. Springer, Berlin, Heidelberg, 2010.
[22] Sun, Xiankun, Huijie Liu, Shiqian Wu, Zhijun Fang, Chengfan Li, and Jingyuan Yin. "Low light image enhancement based on guided image filtering in gradient domain." International journal of digital multimedia broadcasting 2017 (2017).
[23] Xu L., Lu C., Xu Y., & Jia J. 2011. Image smoothing via L_0 gradient minimization. ACM Transactions on Graphics.30(6):174.
[24] Ding X, Chen L, Zheng X, Huang Y, & Zeng D. 2016. Single image rain and snow removal via guided L_0 smoothing filter. Multimedia Tools and Applications. 75(5):2697-71.
[25] Dagar, Naveen Singh, and Pawan Kumar Dahiya. "Edge detection technique using binary particle swarm optimization." *Procedia Computer Science* 167 (2020): 1421 1436.