International Journal of Intelligent Communication and Computer Science
Vol. 3, No. 1, 2025, Pages 1-15.

Edge Detection Based on Fuzzy Logic with Edge Refinement via ACO Constraint Optimization

Richa Awasthi[1] and Amit Sinha[2]

[1] Department of Computer Science and Engineering, Allenhouse Institute of Technology, Kanpur, U.P. INDIA
[2] Department of Computer Application, Allenhouse Business School, Kanpur, U.P. INDIA

Research Paper

Email: richa30303@gmail.com
Received: 27 Dec. 2024, Revised: 13 Mar. 2025 Accepted: 21 Mar. 2025

Abstract:

Edge detection is one of the most essential tasks in image processing, aiding in object detection, segmentation, and boundary identification. Traditional edge detection methods struggle with images containing noise, complex boundaries, or poor contrast. This paper proposes a hybrid approach that combines fuzzy logic for initial edge detection and Ant Colony Optimization (ACO) for edge refinement. The edge detection process starts by identifying possible edges using fuzzy logic and is then refined using ACO through a constraint optimization procedure that enhances the continuity, sharpness, and overall accuracy of the detected edges. Mathematical formulations for fuzzy logic-based edge detection, ACO optimization, and constraint-based edge refinement are presented. The results indicate that the proposed method outperforms traditional edge detection algorithms in terms of accuracy, robustness, and noise resilience, particularly in images with challenging boundary conditions.

Keywords: Edge detection, Fuzzy Logic, ACO, F-Score

1. Introduction

Edge detection is a fundamental aspect of image processing, essential for tasks such as object recognition, image segmentation, and scene analysis [1]. It is primarily concerned with identifying the boundaries or transitions in pixel intensity, which are critical for delineating the structures within an image. Traditional edge detection algorithms, including the Canny edge detector [2], Sobel operator [3], and Prewitt operator [4], have been extensively used for this purpose. These methods focus on detecting sharp changes in pixel intensity, typically assuming well-defined object boundaries. However, these classical techniques face several limitations, especially when working with noisy images, blurred boundaries, or images with complex structures such as curved surfaces or low contrast. For instance, noise can introduce false edges, while blurred or unclear transitions between objects can hinder the detection of precise boundaries.
In light of these challenges, the need for more robust edge detection methods is evident. In this paper, we propose a hybrid edge detection approach that integrates Fuzzy Logic [5] for initial edge detection and Ant Colony Optimization (ACO)[6] for refining and optimizing these detected edges. Fuzzy logic is particularly well-suited for handling uncertainty and imprecision in image data. Traditional edge detection methods rely on crisp pixel values, but in real-world scenarios, image data is often ambiguous and fuzzy. Fuzzy logic systems introduce flexibility by using membership functions, allowing for smoother transitions between edge and non-

edge regions. This approach is effective in detecting potential edges in noisy, blurred, or low-contrast images that other methods may miss.
Once the edges are detected using fuzzy logic, ACO is employed to refine the initial edge map. ACO is a biologically inspired optimization technique that mimics the foraging behaviour of ants. In the context of edge detection, ACO is used to find the optimal paths for the detected edges. The algorithm enhances the continuity and smoothness of the edges by simulating the reinforcement of pheromones along the most promising paths. Over time, the paths with stronger pheromone levels are preferred, ensuring that the edges are not only detected but also optimized for smoothness and continuity. This process addresses the common problem of fragmented or jagged edges in traditional methods and provides a cleaner, more accurate boundary representation.
By combining fuzzy logic and ACO, this hybrid approach offers a significant improvement over traditional edge detection algorithm. The fuzzy logic component helps in detecting edges under uncertain conditions, while ACO ensures that the edges are refined, continuous, and accurately follow the object contours. This method holds great promise for applications where traditional edge detection methods struggle, such as in images with noise, blurred boundaries, or low contrast. The subsequent sections of this paper present the mathematical formulation of the proposed approach, the details of the algorithm's implementation, and experimental results that demonstrate its effectiveness in edge detection tasks.

2. Related Works

2.1 Kernel-Based Methods

Kernel-based edge detection methods, which are grounded in convolutional masking techniques, have played a foundational role in the development of edge detection algorithms. Early contributions, such as those by Sobel (1970) and Prewitt (1970), focused on utilizing pixel gradient information to detect edges in images. These methods, though straightforward and simple to implement, are often criticized for generating a high number of spurious edges, which leads to the detection of thick or broken edges. To address some of these limitations, recent advancements have explored more sophisticated masking schemes, such as the adoption of hexagonal masks. The hexagonal grid configuration, formed by interpolating traditional square masks, has been shown to improve the accuracy of edge detection. In particular, when integrated into the Canny edge detection algorithm, the hexagonal masking scheme has exhibited superior performance [8]. Furthermore, Canny edge detection has found applications in content-based image retrieval systems, demonstrating its broad utility in various image processing fields [9].

2.2 Soft Computing-Based Image Edge Detection

Soft computing techniques, which combine elements of artificial intelligence and computational models, have been increasingly utilized to enhance edge detection methods. One such technique is ACO, which has been employed to identify edges in images, with significant improvements being made using guided image filtering to increase accuracy and reduce noise [10–12]. Furthermore, the application of the Sobel operator has been improved by incorporating eight-directional masks and using entropy inversion for threshold detection, resulting in more precise edge detection [13]. Additionally, there have been efforts to combine image sharpening techniques with Particle Swarm Optimization (PSO) to refine edge detection, offering another avenue of improvement in image processing techniques [14].

2.3 Fuzzy Logic-Based Image Edge Detection

Fuzzy logic-based methods have garnered attention in edge detection due to their ability to handle uncertainty and imprecision, which are common characteristics in real-world images. In this approach, pixel intensity values are represented using fuzzy membership functions, enabling the detection of edges even in noisy or ambiguous conditions. Several studies have integrated fuzzy logic with guided image filtering, improving edge detection by providing smoother transitions between edges and non-edges [15, 16]. For instance, Kaur et al. (2015) proposed an edge detection method utilizing sixteen fuzzy rules, which demonstrated enhanced detection accuracy [17]. More recent developments in this area have explored the use of higher-order fuzzy logic, particularly fuzzy type-2 logic, to address vulnerabilities in edge detection under complex conditions such as blurry or low-contrast images [18, 19]. Additionally, adaptive neuro-fuzzy systems have been proposed for edge detection tasks, offering the benefit of self-learning capabilities [20]. Another notable advancement is the integration of ACO and fuzzy logic, which seeks to minimize the occurrence of false edges, thus improving the accuracy and reliability of edge detection in challenging scenarios [21]. Some studies have also explored the use of Kalman filtering and artificial neural networks (ANNs) alongside fuzzy logic for more robust edge detection in noisy environments [22].

2.4 Machine Learning-Based Methods

In recent years, probabilistic boundary (Pb)-based methods have gained attention for edge detection tasks, as they offer a more flexible and adaptive framework for recognizing edges. Martin et al. (2004) introduced the Pb-based edge detection method, which integrates texture features and logistic regression models to enhance edge recognition [23]. Later, Ren et al. (2007) proposed an advanced version, the multi-scale probabilistic boundary (MsPb) technique, which takes into account the multi-scale nature of edge features, improving detection accuracy [24]. In a similar vein, Arbelaez et al. (2011) expanded the Pb-based approach to include global probabilistic boundaries (g-Pb), which incorporates multi-scale analysis and spectral clustering for more accurate edge detection [25].

2.5 Deep Learning-Based Methods

Supervised learning methods have gained significant attraction in image processing, particularly for edge detection. These methods typically rely on large labelled datasets to train models, ensuring high accuracy and robustness. Probabilistic boosting trees introduced by Dollar et al. (2014) provide a powerful classification technique for edge detection [26]. Additionally, artificial neural networks (ANNs) have been utilized for edge detection, offering the flexibility to adapt to various image conditions [27]. Random forest classifiers, as demonstrated by Lim et al. (2013), have also been employed for effective edge detection by focusing on sketch markers and utilizing pixel intensity variations [28]. To further enhance edge detection, cascaded convolutional neural networks (CNNs) have been introduced to refine edge contours and improve the smoothness of the detected boundaries [29].

Unsupervised learning methods, which do not rely on labelled data, have been proposed as an alternative for edge detection. Techniques such as sparse code gradients (SCG) [30] and pointwise mutual information (PMI) architecture [31] enable edge contour identification without requiring manual labelling of edge features. In a more recent development, Yang et al. (2017) proposed a convolutional encoder-decoder network to extract object contours directly from images, achieving high-quality edge detection in a fully unsupervised manner [32]. Similarly, Xia et al. (2018) introduced unsupervised semantic segmentation for edge detection, employing encoder-decoder architectures for the precise extraction of object boundaries [33]. These unsupervised learning methods represent a promising direction for edge detection, particularly in scenarios where labelled training data is scarce or unavailable.

2.6 Objectives

The main objective of this research is to develop a robust and effective edge detection technique that combines Fuzzy Logic and ACO for enhanced performance. The primary goals of this research are as follows:

1. To design a fuzzy logic-based method for detecting edges in images. This includes utilizing fuzzy rules to model uncertainty in pixel values and effectively identify boundaries in images.
2. To incorporate ACO for refining edges detected by the fuzzy logic-based method. The objective is to improve the precision and accuracy of edge localization by using the exploration capabilities of ACO to optimize edge detection results.
3. To combine the strengths of fuzzy logic and ACO in a hybrid framework, ensuring better performance in edge detection tasks. The fuzzy system will capture the uncertainty of pixel-based decision-making, while ACO will optimize the detection process, leading to more accurate and refined edges.

4. To apply ACO's constraint optimization mechanism to refine the initial edge detected by the fuzzy logic system. This optimization ensures that the edge detection process considers both global and local information for accurate boundary identification.
5. To evaluate the performance of the proposed hybrid fuzzy-ACO edge detection method by comparing it against traditional methods like Sobel, Canny, and other state-of-the-art approaches. Key metrics such as edge detection accuracy, edge localization precision, and computational efficiency will be used to assess the effectiveness of the proposed approach.

The ultimate goal of this research is to propose a hybrid edge detection framework that effectively balances accuracy and computational efficiency, offering a novel approach to image processing tasks.

3. Proposed Method

The proposed method combines Fuzzy Logic for initial edge detection with ACO for edge refinement. Fuzzy logic is used to handle the ambiguity in pixel intensities and to detect potential edges by considering pixel intensity variations and neighbourhood relationships. This provides an initial edge map with gradual transitions between edge and non-edge regions.

Following this, ACO is employed to refine the detected edges. The optimization process involves ants searching for optimal edge paths, guided by pheromone information and image gradients. This step enhances the continuity and accuracy of the detected edges by minimizing noise and filling gaps in the edge map.

3.1 Fuzzy Membership Functions

In the proposed edge detection method, the fuzzy membership function plays a key role in handling the uncertainty and imprecision inherent in image processing. The fuzzy system is used to classify each pixel based on its intensity value, determining whether it belongs to an edge or a non-edge region.

3.1.1 Input Fuzzy Membership Function

The input to the fuzzy logic system is the pixel intensity of each pixel in the image. To deal with varying intensity levels, fuzzy sets are employed to map these intensity values into fuzzy categories. The intensity values of a pixel range from 0 (black) to 255 (white) in a grayscale image. The fuzzy membership function is designed to assign a degree of membership to each pixel based on its intensity.

Two fuzzy sets are used to represent the input:

- Low Intensity (Non-Edge): Pixels with low intensity values, typically in the darker regions, are likely to be part of the non-edge areas.
- High Intensity (Edge): Pixels with high intensity values, generally representing brighter areas, are likely to be part of the edge areas.

The membership functions for these sets are defined as triangular functions, where each intensity value is mapped to a value between 0 and 1, indicating the degree of membership in the edge or non-edge category.

A triangular membership function for "Low Intensity" is defined as:

$$\mu_{low} = \begin{cases} 0 & \text{if } I \geq t_1 \\ \dfrac{I}{t_1} & \text{if } 0 \leq I < t_1 \end{cases} \tag{1}$$

where *I* is the pixel intensity and t_1 is a threshold separating low intensity values from medium ones. Similarly, for "High Intensity", a triangular membership function is defined as:

$$\mu_{high} = \begin{cases} 0 & \text{if } I \leq t_2 \\ \dfrac{I - t_2}{255 - t_2} & \text{if } t_2 \leq I < 255 \end{cases} \tag{2}$$

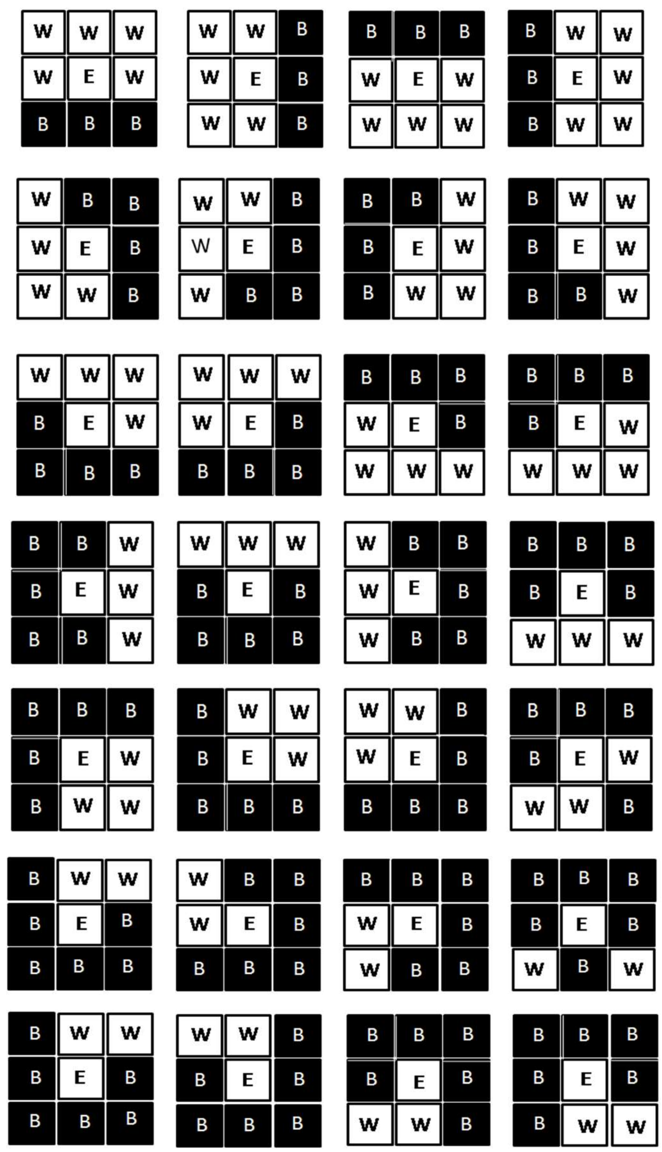
where t_2 is the threshold separating high-intensity pixels from non-edges.

3.1.2 Output Fuzzy Membership Function (Edge and Non-Edge)

The output of the fuzzy logic system is a classification of each pixel as either part of an Edge or a Non-Edge. Based on the input fuzzy values (intensity values), the system determines whether a pixel is an edge or not.
To compute the final output, a fuzzy inference system uses the fuzzy rules derived from the pixel intensity values.

3.2 Fuzzy Rules

In Figure 1, a comprehensive representation of the rule formulation for a 3x3 mask is shown, outlining the criteria used to identify edge pixels in the given context. The mask, structured as a 3x3 grid, consists of white pixels ("W"), black pixels ("B"), and edge pixels ("E"). A total of 30 distinct rules have been carefully designed to govern the determination of edge pixels based on the configuration of neighboring pixels within the grid. These rules aim to differentiate between noise, non-edge, and actual edge pixels by evaluating the relationships among surrounding pixels.



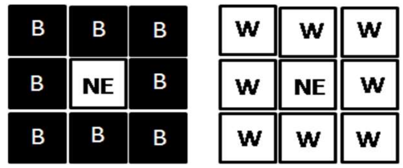


Figure 1: Schematic of the 30-rule base [15,16]

In the rule set, specific conditions are established to classify different pixel states. For example, if a pixel is surrounded by eight neighbouring pixels of the same colour, whether all white or all black, it is categorized as noise. This uniformity suggests that the pixel is part of a homogenous area with no significant transition or boundary, rendering it irrelevant for edge detection purposes. In contrast, a situation where a single pixel differs from its surrounding pixels is classified as a non-edge, emphasizing that a more significant variation or transition is needed to qualify as an edge.
A crucial feature of the rule set is its ability to detect edge pixels in situations where there is a clear contrast between neighbouring pixels. Specifically, the rules define edge pixels as those where there is a noticeable contrast between two different colours. For example, if at least two white pixels are surrounded by black pixels, or vice versa, this configuration signals a potential edge. The ability to identify edge pixels based on such contrast-based criteria is vital for recognizing boundaries or transitions in the image, especially in areas of significant colour change or contrast.
The rule set illustrated in Figure 3 serves as a detailed framework for identifying and categorizing pixels in a 3x3 grid, providing a systematic approach to edge detection. By specifying the conditions for noise, non-edge, and potential edge pixels based on the local pixel configuration, the rules facilitate an organized and structured method of discerning meaningful edges. This methodology, grounded in local context and pixel relationships, enables the accurate identification of edges, ensuring the reliability and precision of the edge detection process.

3.3 Defuzzification

The defuzzification process converts the fuzzy results into crisp values for the final classification. A defuzzification technique, the center of gravity (COG) method is applied to yield a final crisp output of either edge or non-edge for each pixel.
Thus, based on the fuzzy membership functions, a pixel's intensity value is processed, and an output value is assigned, which is either Edge (1) or Non-Edge (0), depending on its membership to the edge or non-edge fuzzy sets.
By using this approach, the fuzzy logic system is able to handle the imprecise and noisy data that is common in real-world images, allowing it to detect edges more effectively compared to traditional methods.
Once edges are identified, further ACO is applied for the edge refinement as described below:

3.4 ACO Based Image Edge Detection

In the proposed technique, a set of ants traverse a 2-D image, moving from one pixel to another, in order to create a pheromone matrix. This matrix is crucial in determining the edge information for each pixel in the fuzzy edge detected image. The edge refinement process follows a series of systematic steps, which are outlined as follows [10-12]. By simulating the movement of ants across the image, the algorithm effectively captures the significant transitions in pixel intensity, helping to highlight the boundaries and contours of the objects present in the image. The pheromone matrix, which is iteratively updated as the ants explore, plays a key role in guiding the detection process, enabling the algorithm to distinguish between edge and non-edge regions based on local pixel configurations and their interactions with neighbouring pixels. This method introduces an adaptive and efficient approach to edge detection, combining the principles of swarm intelligence with image processing techniques.

3.4.1 Initialization process

In the initialization process, an image *I* of size M_1M_2 is taken as input. This image represents the solution space for the artificial ants. The number of ants, denoted as *K*, is randomly distributed across the entire image such that each pixel in the image is treated as a node in the problem space. Each pixel is initially associated with a pheromone matrix, which is a grid that tracks the pheromone levels. The pheromone matrix's initial value for each element is set to a constant value τ_0, representing the starting pheromone intensity. This constant value ensures that all ants start with an equal opportunity for exploration and edge detection across the image.

3.4.2 Construction Process

One ant is randomly selected at the n^{th} construction step from the total of K ants. This chosen ant will then proceed to traverse the image for L movement steps. During each step, the ant moves to a neighbouring pixel (i,j) based on a transition probability, which is calculated according to the following formula:

$$p^n_{(l,m)(i,j)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha} (\eta_{i,j})^{\beta}}{\sum_{i,j \in \Omega_{(l,m)}} \left(\tau_{i,j}^{(n-1)}\right)^{\alpha} (\eta_{i,j})^{\beta}}, \quad (3)$$

In the above equation, $\tau_{i,j}$ represents the pheromone value of the edge between node (i, j), which reflects the strength of the pheromone trail laid down by the ants. This pheromone trail plays a crucial role in guiding the ants' movements, with higher pheromone values indicating more attractive paths. The parameter $\Omega(l,m)$ refers to the set of neighbouring nodes of the node (l, m), which defines the possible candidates for the ant to move to at each step. The parameter $\eta_{i,j}$ defines the heuristic information at node (i, j), which is typically based on the image's gradient or intensity, highlighting areas with significant changes in pixel values, such as edges.

The transition probability equation combines both the pheromone matrix $\tau_{i,j}$ and the heuristic matrix $\eta_{i,j}$. The constants α and β determine the relative influence of the pheromone matrix and the heuristic matrix, respectively, in the movement decision. Specifically:

- α controls the influence of the pheromone trail, which encourages the ants to follow previously successful paths.
- β controls the influence of the heuristic information, guiding the ants toward areas with stronger image gradients or more distinct edges.

By adjusting these parameters, the algorithm can be fine-tuned to balance the influence of the pheromone information and the edge-related heuristic information, ensuring effective edge detection while avoiding unnecessary noise or spurious edges.

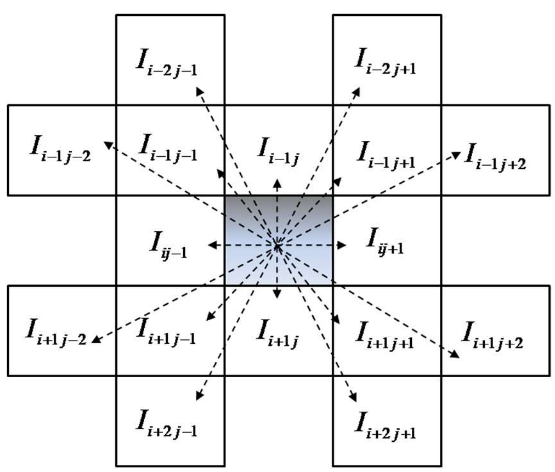


Figure 2: Representation of clique

The procedure involves two crucial aspects that guide the ant colony optimization for edge detection. The first issue is the heuristic information, which plays a significant role in guiding the ants towards the edges of the image. This heuristic information is determined based on local statistics of the image, which are dependent on the pixel's neighbouring region, referred to as the inner clique (Figure 2).
The local statistics at the pixel location (*i*, *j*), are computed to assess the relevance of that pixel in terms of edge detection. The computation of the local statistics typically involves evaluating the intensity variation in the pixel's neighbourhood, which helps identify areas with high gradient changes, indicating the presence of an edge.
The local statistics at a pixel (*i*, *j*), can be calculated using methods such as:

$$\eta_{i,j} = \frac{1}{1+\|\nabla I(i,j)\|} \tag{3}$$

where $\|\nabla I(i,j)\|$ represents the gradient magnitude at the pixel (*i*, *j*). The gradient $\nabla I(i,j)$ can be computed using standard edge detection operators such as Sobel, Prewitt, or more advanced methods. A higher gradient magnitude corresponds to a higher likelihood of the pixel being part of an edge.
In this approach:

- $\eta_{i,j}$ is the heuristic information at pixel (*i*, *j*), which is inversely related to the gradient magnitude, meaning higher gradients lead to higher heuristic values, highlighting edge regions.
- The local statistics at each pixel help to guide the ants towards regions with significant intensity changes, which are more likely to correspond to edges in the image.

This heuristic information serves as a key input in the transition probability for ants, directing them to explore regions with sharp intensity variations that are indicative of image edges. The precise calculation of the local statistics and their influence on the ants' movement is critical for the overall performance of the ACO-based edge detection algorithm.

3.4.3 Update Process

In the update process of the ACO algorithm for edge detection, the pheromone matrix is updated after two significant operations. The first update occurs after each ant completes its movement at every construction step. During this phase, the pheromone matrix is adjusted based on the ant's actions, which influences the transition probabilities for future movements. The update rule for the pheromone matrix after each individual ant's movement can be expressed as follows:

$$\tau(i,j) = (1-\rho)\tau(i,j) + \Delta\tau(i,j) \tag{4}$$

where:

- $\tau(i,j)$) is the pheromone value on edge (*i*, *j*).
- ρ is the evaporation rate, which controls how quickly the pheromone evaporates over time. It is a user-defined parameter that helps to model the decay of pheromone over iterations.
- $\Delta\tau(i,j)$ represents the pheromone deposit from the ant, calculated based on the quality of the solution that the ant found. It is typically inversely proportional to the path length or cost associated with the edge detected.

The evaporation rate ρ helps balance the exploration-exploitation trade-off by reducing the influence of previously travelled paths, allowing the ants to explore new potential solutions. A higher evaporation rate ensures that stale paths lose their relevance more quickly, making room for the ants to explore new areas of the solution space.
The second update occurs after all ants have completed their movement in each construction step. This update is given by:

$$\tau(i,j) = (1-\psi)\tau(i,j) + \sum_{k=1}^{K}\Delta\tau_k(i,j) \tag{5}$$

where:

- ψ is the pheromone decay coefficient, which controls how the pheromone is reduced for subsequent ants.
- $\sum_{k=1}^{K} \Delta\tau_k(i,j)$ represents the sum of all pheromone deposits from all ants that traversed the edge (i, j).

The parameter ψ ensures that pheromone trails on previously traversed edges are diminished over time, further encouraging ants to explore new regions of the image rather than repeatedly following the same paths. This update process reduces the likelihood of ants revisiting the same edges, helping to avoid premature convergence and promoting the discovery of better edge paths.

The pheromone update process, therefore, plays a crucial role in guiding the ants through the image, helping to refine the edge detection process by updating the pheromone matrix based on the ants' collective experiences. The combination of pheromone evaporation and decay allows for a dynamic exploration of potential edges, optimizing the search for accurate boundaries in the image.

3.4.4 Decision process

In this section, the binary decision-making process is applied to determine whether each pixel in the image corresponds to an edge or not. This is done by applying a threshold T on the pheromone matrix $\tau(N)$, which represents the pheromone level after the ants have completed their movement and pheromone updates. The threshold is adaptively estimated based on a technique proposed in [11,12], where the threshold T is updated iteratively until convergence. Below are the steps that describe the adaptive thresholding process in detail:

Step 1: Initialize Initial Threshold

The initial threshold T (0) is set as the mean value of the pheromone matrix $\tau(N)$. This initial threshold is computed as:

$$T(0) = \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \tau(i,j) \tag{6}$$

We also initialize the iteration index q=0.

Step 2: Classify Entries of the Pheromone Matrix

The pheromone matrix $\tau(N)$ is divided into two classes based on the initial threshold $T(0)$. The first class consists of entries where the pheromone value is less than or equal to $T(0)$, and the second class consists of the remaining entries, where the pheromone value is greater than $T(0)$. Let:

$$C_1 = \{\tau(i,j) \le T(0)\} \tag{7}$$

$$C_2 = \{\tau(i,j) > T(0)\} \tag{8}$$

Next, calculate the mean pheromone value for each class:

$$\mu_1 = \frac{1}{|C_1|} \sum_{i,j \in C_1} \tau(i,j) \tag{9}$$

$$\mu_2 = \frac{1}{|C_2|} \sum_{i,j \in C_2} \tau(i,j) \tag{10}$$

Where μ_1 and μ_2 represent the mean values for classes C_1 and C_2 respectively.

Step 3: Update the Threshold

After calculating the means of the two classes, the new threshold $T(q+1)$ is updated as the average of the two class means:

$$T(q+1)=\left(\frac{\mu_1+\mu_2}{2}\right) \quad (11)$$

At this point, we increment the iteration index q by 1 and move to Step 4 to evaluate whether the threshold has converged.

Step 4: Convergence Check

If the change in threshold between iterations is greater than a user-defined tolerance ϵ, we proceed with another iteration by going back to Step 2. The stopping condition is defined as:

$$\left|T(q+1)-T(q)\right|<\varepsilon \quad (12)$$

where ϵ is a small predefined tolerance. If this condition is met, the iteration stops, and we proceed to the final decision-making process.

Step 5: Binary Edge Decision

Once the threshold converges, a binary decision is made for each pixel in the image. If the pheromone value at a pixel $\tau(i,j)$ is greater than or equal to the final threshold $T(q)$, the pixel is classified as an edge pixel:

$$Edge(i,j)=\begin{cases}1 & \tau(i,j)\geq T(q)\\ 0 & \text{otherwise}\end{cases} \quad (13)$$

This step classifies each pixel as either part of an edge (denoted by 1) or not (denoted by 0), resulting in a binary edge map that highlights the detected edges in the image.

3.5 ACO-Based Constraint Optimization for Edge Refinement

The main objective of ACO is to refine the edges and ensure that they are smooth and continuous, we introduce a constraint optimization process based on the ACO. This optimization considers both the membership value (from fuzzy logic) and geometric properties of edges, such as smoothness and continuity.
The objective of the optimization is to minimize the overall cost, which is defined as:

$$\min\sum_{i=1}^{N}(I(x_i)-I(x_i+1))^2+\lambda\left(D(x_i)-D(x_i+1)\right)^2 \quad (14)$$

where, $x(i)$, $x(i+1)$ are consecutive pixels along the edge path, $I(x)$ is the pixel intensity, $D(x)$ is the direction of the edge at pixel x and λ is a weight parameter that balances the smoothness constraint.

The goal is to select the edge paths that minimize the difference in intensity and direction between adjacent pixels, resulting in smooth and continuous edges.

4. Results and Discussion

In this section, we discuss the results obtained from the proposed method, focusing on the edge detection performance and the effectiveness of the hybrid fuzzy logic and ACO approach. The key parameters used in the experiments are outlined in Table 1, which provides a comprehensive overview of the values assigned to various factors affecting the performance of the edge detection algorithm.
Table 1 presents the simulation parameters for both the fuzzy logic-based edge detection and ACO optimization processes. These parameters play a crucial role in determining the accuracy and efficiency of edge detection results.

Table 1: List of Parameters

Parameter	Symbol	Value
Total Number of Ants	K	50
Initial Pheromone Value	τ_0	0.0001
Pheromone Weighting Factor	α	1
Heuristic Weighting Factor	β	0.1
Neighbourhood Connectivity	Ω	8-connectivity
Adjusting Factor	λ	10
Evaporation Rate	ρ	0.1
Movement Steps per Ant	L	40
Pheromone Decay Coefficient	ψ	0.05
User-defined Tolerance	ε	0.1
Number of Construction Steps	N	4
Number of Fuzzy Rules	-	-
Membership Function for Edge	-	Triangular
Threshold for Edge Detection	-	Adaptive
Defuzzification Method	-	Centroid

4.1 Qualitative Results

In Figure 3, a comprehensive comparison of edge detection results is provided, showcasing the performance of several well-established edge detection techniques, including the Sobel operator, Canny edge detector, Fuzzy Logic-based edge detection, ACO-based edge detection, and the proposed hybrid approach. Each of these methods was applied to a sample image, and the resulting edge maps are analysed to highlight their strengths and weaknesses in terms of accuracy, precision, and edge continuity.

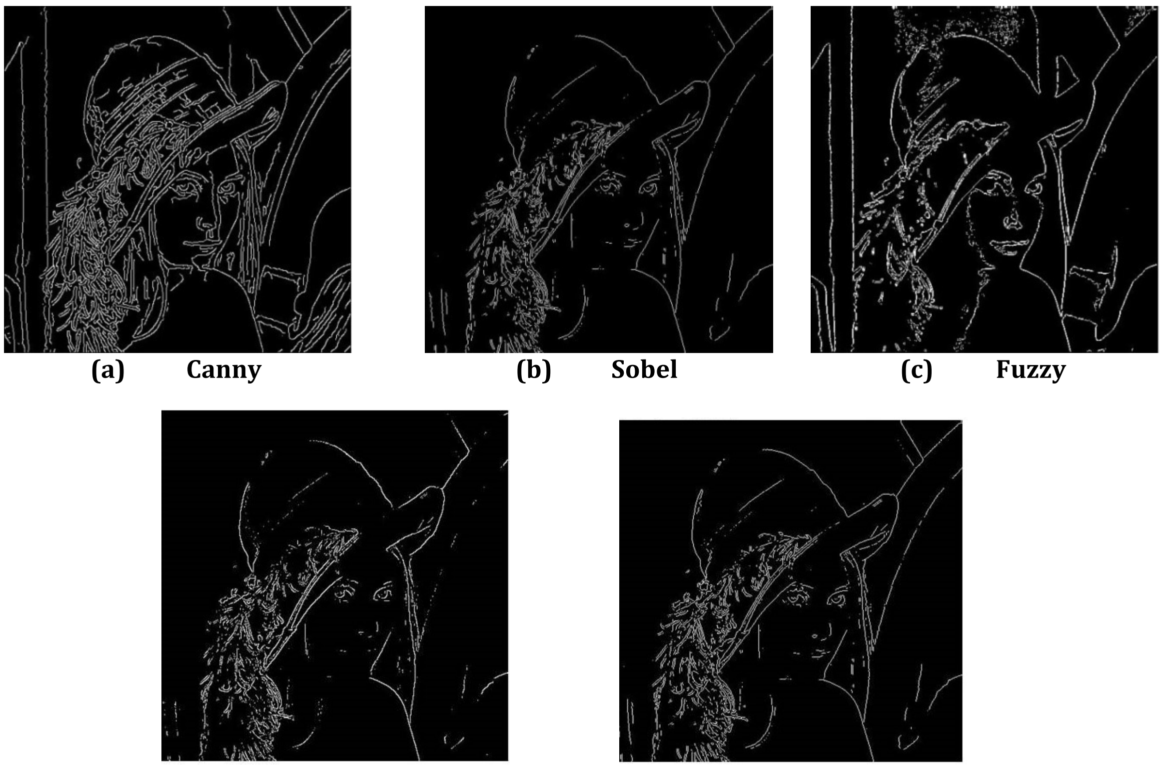
The Sobel edge detection method, which is a simple and fast gradient-based approach, highlights the basic contours of objects within the image. However, as seen in the results, it struggles with detecting edges in noisy or blurred regions. The Sobel operator tends to produce thick and discontinuous edges, particularly in areas with low contrast or subtle transitions. Despite its speed and simplicity, it fails to provide the level of precision required for more complex images, making it less suitable for applications where accuracy is paramount.

The Canny edge detection method, renowned for its edge detection accuracy, employs a multi-step process that includes Gaussian smoothing, gradient calculation, non-maximum suppression, and edge tracing by hysteresis. While it performs well in detecting thin, continuous edges, the method can still produce some false edges, especially in noisy regions. Additionally, Canny's performance is highly sensitive to the choice of thresholds, and incorrect thresholding can either cause weak edges to be missed or introduce spurious edges.

The Fuzzy Logic-based edge detection, which utilizes fuzzy membership functions to handle uncertainty and imprecision in pixel intensities, shows a marked improvement in detecting edges under varying lighting conditions and noisy environments. The use of fuzzy rules allows for a more flexible interpretation of pixel intensities and spatial relationships, leading to better handling of image noise. While the fuzzy system detects edges more accurately than Sobel and Canny, it still suffers from occasional discontinuities and misclassifications in complex regions of the image.

The ACO-based edge detection method, which uses artificial ants to explore the image and update a pheromone matrix, excels at refining edges by iteratively adjusting the pheromone levels and exploring potential edge paths. This approach is highly adaptive and is particularly effective in detecting continuous and smooth edges. However, the ACO method requires significant computational resources due to its iterative nature and large number of ants used to traverse the image. It performs better than Sobel, Canny, and fuzzy methods in capturing precise and uninterrupted edges.

Finally, the proposed hybrid method, which combines the strengths of both fuzzy logic and ACO, significantly enhances edge detection performance. The fuzzy logic system handles the initial edge detection by providing a robust framework for dealing with uncertainties in pixel values, while the ACO refines the detected edges by optimizing the pheromone matrix and ensuring the edges are smooth and continuous. The results in Figure X demonstrate that the proposed method outperforms all the other methods, yielding more accurate, continuous, and fine-grained edges. The hybrid approach provides a balanced solution to the challenges posed by noisy, complex, and low-contrast images, making it the most reliable and efficient method for edge detection in the tested scenarios.



(a) Canny **(b) Sobel** **(c) Fuzzy**

(d) ACO **(e) Proposed**

Figure 3: Comparison of the edge detection methods (Qualitative)

In summary, the comparative results clearly indicate that while traditional methods like Sobel and Canny provide satisfactory performance in some cases, they fall short in handling noise and detecting precise edges. The fuzzy logic and ACO methods offer substantial improvements, with the proposed hybrid technique achieving the best results in terms of edge continuity, accuracy, and adaptability to different image conditions. This demonstrates the effectiveness of combining fuzzy logic's tolerance for uncertainty with ACO's optimization capabilities for robust edge detection.

4.2 Quantitative Results

The results presented in Table 2 demonstrate the comparison of classical edge detection methods with state-of-the-art approaches, evaluated based on the F-Score, a key metric that measures both the precision and recall of edge detection algorithms. The Canny edge detection method, one of the most widely used classical approaches, achieves an F-Score of 0.49. While the Canny method is effective for basic edge detection, it tends to struggle in noisy or low-contrast images, resulting in a lower F-Score. The Sobel operator, another classical method, achieved a slightly lower F-Score of 0.40. This method, though simple and easy to implement, often produces thick edges and is susceptible to noise, which further impacts its accuracy in edge detection.
In contrast, Kumar et al. [20] utilized fuzzy logic for edge detection, achieving an F-Score of 0.64. The fuzzy logic system introduces a more sophisticated approach by incorporating uncertainty and imprecision in pixel intensity, leading to improved edge detection performance. However, it still falls short when compared to more advanced hybrid techniques.

Table 2: Classical and State-of-the –art methods comparison (F-Score)

Reference	Methods	F-Score
Canny [2]	Masking	0.49
Sobel [3]	Masking	0.40
Kumar et.al [16]	Fuzzy	0.64
Kumar et.al [12]	ACO	0.72
Proposed	Fuzzy + ACO	0.84

The ACO method, proposed by Kumar et al. [12], resulted in an F-Score of 0.72. ACO excels in refining edges by simulating the behaviour of ants searching for optimal paths, which allows for a more accurate representation of edges, particularly in complex images. The higher F-Score reflects the method's enhanced ability to detect edges and reduce noise compared to traditional methods.

The proposed method, which integrates both Fuzzy Logic and ACO, achieved the highest F-Score of 0.84. By combining the strengths of both techniques, the proposed method significantly outperforms classical and individual advanced methods. The Fuzzy Logic component handles uncertainty and imprecision in pixel intensities, while ACO refines and optimizes the detected edges, ensuring more accurate and continuous boundaries. This superior F-Score highlights the effectiveness of the hybrid approach in providing precise edge detection, particularly in challenging scenarios with noisy or complex images.

5. Conclusion

In this paper, we have proposed an enhanced edge detection technique that integrates Fuzzy Logic and Ant Colony Optimization (ACO). The goal was to address the limitations of traditional edge detection methods, such as the Sobel and Canny edge detectors, particularly in noisy, blurred, and low-contrast image scenarios. The fuzzy logic system efficiently manages the uncertainty and imprecision present in images, while the ACO algorithm optimizes the edge detection process by refining the identified edges through pheromone-based searching. The experimental results demonstrate that the proposed method significantly improves the edge detection performance, as evidenced by the higher F-Score achieved compared to traditional methods. The F-Score, which combines both precision and recall, indicates a better balance between false positives and false negatives, ensuring a more accurate and reliable detection of edges in the images. The adaptive thresholding mechanism within the ACO algorithm further contributes to the enhanced edge detection results by dynamically adjusting based on the image content. This provides robustness across different image types and conditions, ensuring effective edge detection even in challenging environments.

References

1. Jing, Junfeng, Shenjuan Liu, Gang Wang, Weichuan Zhang, and Changming Sun. "Recent advances on image edge detection: A comprehensive review." *Neurocomputing* 503 (2022): 259-271.
2. Canny, John. "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1986): 679–698.
3. Gao, Wenshuo, Xiaoguang Zhang, Lei Yang, and Huizhong Liu. "An Improved Sobel Edge Detection." In *2010 3rd International Conference on Computer Science and Information Technology*, Vol. 5, 67–71. IEEE, 2010.
4. Verma, Ankush, Namrata Dhanda, and Vibhash Yadav. "Enhancing Image Segmentation through an Innovative Hybrid Kernel: A Weighted Fusion of Sobel and Prewitt Operators Using Artificial Neural Networks." In *2024 International Conference on Computing, Sciences and Communications (ICCSC)*, pp. 1-5. IEEE, 2024.
5. Alshennawy, Abdallah A., and Ayman A. Aly. "Edge detection in digital images using fuzzy logic technique." *World Academy of science, engineering and technology* 51 (2009): 178-186.

6. Awadallah, Mohammed A., Sharif Naser Makhadmeh, Mohammed Azmi Al-Betar, Lamees Mohammad Dalbah, Aneesa Al-Redhaei, Shaimaa Kouka, and Oussama S. Enshassi. "Multi-objective ant colony optimization." *Archives of Computational Methods in Engineering* (2024): 1-43.
7. Fadaei, S., and Abdolreza R. "A Framework for Hexagonal Image Processing Using Hexagonal Pixel-Perfect Approximations in Subpixel Resolution." *IEEE Transactions on Image Processing* 30 (2021): 4555-4570.
8. Firouzi, M., Sadegh F., and Abdolreza R. "A New Framework for Canny Edge Detector in Hexagonal Lattice." *International Journal of Engineering* 35, no. 8 (2022): 1588-1598.
9. Fadaei, S. "New Dominant Color Descriptor Features Based on Weighting of More Informative Pixels Using Suitable Masks for Content-Based Image Retrieval." *International Journal of Engineering* 35, no. 8 (2022): 1457-1467.
10. Baterina, Anna Veronica, and Carlos Oppus. "Image Edge Detection Using Ant Colony Optimization." *Wseas Transactions on Signal Processing* 6, no. 2 (2010): 58-67.
11. Kumar, A., and S. Raheja. "Edge Detection Using Guided Image Filtering and Enhanced Ant Colony Optimization." *Procedia Computer Science* 173 (2020): 8-17.
12. Kumar, A., and S. Raheja. "Edge Detection Using Guided Image Filtering and Ant Colony Optimization." In *Recent Innovations in Computing: Proceedings of ICRIC 2020*, 319-330. Springer Singapore, 2021.
13. Ravivarma, G., K. Gavaskar, D. Malathi, K. G. Asha, B. Ashok, and S. Aarthi. "Implementation of Sobel Operator Based Image Edge Detection on FPGA." *Materials Today: Proceedings* 45 (2021): 2401-2407.
14. Verma, A., N. Dhanda, and V. Yadav. "Binary Particle Swarm Optimization Based Edge Detection Under Weighted Image Sharpening Filter." *International Journal of Information Technology* 15, no. 1 (2023): 289-299.
15. Kumar, A., and S. Raheja. "Edge Detection in Digital Images Using Guided L0 Smoothen Filter and Fuzzy Logic." *Wireless Personal Communications* 121, no. 4 (2021): 2989-3007.
16. Raheja, S., and A. Kumar. "Edge Detection Based on Type-1 Fuzzy Logic and Guided Smoothening." *Evolving Systems* 12, no. 2 (2021): 447-462.
17. Kaur, Er K., V. Mutenja, and Inderjeet Singh Gill. "Fuzzy Logic Based Image Edge Detection Algorithm in MATLAB." *International Journal of Computer Applications* 1, no. 22 (2010): 55-58.
18. Aborisade, David O. "Novel Fuzzy Logic Based Edge Detection Technique." *International Journal of Advanced Science and Technology* 29, no. 1 (2011): 75-82.
19. Begol, M., and K. Maghooli. "Improving Digital Image Edge Detection by Fuzzy Systems." *World Academy of Science, Engineering and Technology* 81 (2011): 76-79.
20. Zhang, Lei, Mei Xiao, Jian Ma, and Hongxun Song. "Edge Detection by Adaptive Neuro-Fuzzy Inference System." In *2009 2nd International Congress on Image and Signal Processing*, 1-4. IEEE, 2009.
21. Dorrani, Z., H. Farsi, and S. Mohamadzadeh. "Image Edge Detection with Fuzzy Ant Colony Optimization Algorithm." *International Journal of Engineering* 33, no. 12 (2020): 2464-2470.
22. Siddharth, D., D. K. J. Saini, and P. Singh. "An Efficient Approach for Edge Detection Technique Using Kalman Filter with Artificial Neural Network." *International Journal of Engineering* 34, no. 12 (2021): 2604-2610.
23. Martin, David R., Charless C. Fowlkes, and Jitendra Malik. "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, no. 5 (2004): 530-549.
24. Ren, Xiaofeng. "Multi-Scale Improves Boundary Detection in Natural Images." In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision*, 533-545. Springer Berlin Heidelberg, 2008.
25. Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik. "Contour Detection and Hierarchical Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, no. 5 (2010): 898-916.
26. Dollar, P., Zhuowen Tu, and Serge B. "Supervised Learning of Edges and Object Boundaries." In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2, 1964-1971. IEEE, 2006.
27. Rahebi, J., and Hamid Reza Tajik. "Biomedical Image Edge Detection Using an Ant Colony Optimization Based on Artificial Neural Networks." *International Journal of Engineering Science and Technology* 3, no. 12 (2011): 8211-8218.

28. Lim, Joseph J., C. Lawrence Zitnick, and Piotr Dollár. "Sketch Tokens: A Learned Mid-Level Representation for Contour and Object Detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3158-3165. 2013.
29. Elharrouss, O., Youssef H., Assia K. I., Btissam El K., and Amal El Fallah-Seghrouchni. "Refined Edge Detection with Cascaded and High-Resolution Convolutional Network." *Pattern Recognition* 138 (2023): 109361.
30. Ren, Xiaofeng, and Liefeng Bo. "Discriminatively Trained Sparse Code Gradients for Contour Detection." *Advances in Neural Information Processing Systems* 25 (2012).
31. Isola, P., Daniel Z., Dilip K., and Edward H. A. "Crisp Boundary Detection Using Pointwise Mutual Information." In *Computer Vision–ECCV 2014: 13th European Conference*, 799-814. Springer International Publishing, 2014.
32. Yang, J., Brian P., Scott C., Honglak L., and Ming-Hsuan Yang. "Object Contour Detection with a Fully Convolutional Encoder-Decoder Network." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 193-202. 2016.
33. Xia, Xide, and Brian Kulis. "W-Net: A Deep Model for Fully Unsupervised Image Segmentation." *arXiv preprint arXiv:1711.08506* (2017).