Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-13
DOI:10.22034/cmde.2025.66702.3288

Edge detection method for color images based on F^1-transform and hierarchical fuzzy inference system

Robab Alikhani* **and Maryam Ganjeh-Alamdari**
Department of Mathematics, University of Tabriz, Tabriz, Iran.

Abstract

This study presents a novel RGB image edge detection algorithm using a first-degree polynomial fuzzy transform (F^1-transform) and a hierarchical fuzzy inference system (HFIS). The F^1-transform generates adaptable convolution kernels that enable adjustable image smoothing and create directional derivatives in four orientations, producing an optimized gradient matrix. These derivatives help connect and refine edges into smooth, continuous contours. For computational efficiency, we implement a multi-layer HFIS structure with two-input single-output fuzzy systems at each layer. This hierarchy progressively refines fuzzified gradient inputs, enhancing edge extraction while reducing noise. The method shows improved performance with additional layers due to flexible rules and diverse membership functions. Comparisons with traditional (e.g., Canny) and modern techniques demonstrate its effectiveness.

Keywords. F^1-transform, Hierarchical fuzzy inference system, Multi-input single-output fuzzy inference system, Optimized gradient matrix.
2010 Mathematics Subject Classification. 68U10, 90C70, 03B52.

1. Introduction

Edge detection is a fundamental technique in image processing for identifying boundaries, crucial for computer vision tasks. Various methods exist, including gradient-based approaches (Sobel, Prewitt, Roberts) that detect intensity changes, and the Laplacian of Gaussian, which highlights regions of rapid variation. The Canny detector is a prominent multi-stage technique involving smoothing, gradient calculation, non-maximum suppression, and hysteresis thresholding for reliable results [7, 8, 13, 20, 21, 24].

Fuzzy-based edge detection techniques offer robustness to noise and precise edge localization by effectively handling uncertainty through fuzzy logic and membership functions. These methods outperform traditional approaches in noisy environments [3, 5, 12, 16, 17, 23, 25]. Various implementations include multi-level thresholding for contrast enhancement [25], Sobel and type-2 fuzzy hybrids for uncertainty management [12], generalized type-2 fuzzy sets for detailed feature capture [16], univalue segment fuzzification with bio-inspired optimization [23], and specialized applications like retinal vessel detection [17]. Hardware-aware approaches using nanotube transistors have also been explored [5].

Fractional calculus operators enhance edge detection by capturing complex textures through fractional derivatives and filters, leveraging their ability to model long-range dependencies and non-local signal characteristics [4, 10, 14, 22, 26]. These methods excel with natural images exhibiting power-law or fractal properties. Specific applications include Caputo-Fabrizio based masks for medical imaging [14], Atangana-Baleanu operators for real images and retinal vessel detection [10, 22], and memristor-based neural networks for detailed edge extraction [26].

In the current contribution, we present an edge detection algorithm based on the first degree polynomial fuzzy transform, also known as F^1-transform, and fuzzy inference systems. This algorithm aims to detect edges in an image, while preserving edge details and being robust against noise. The F^1-transform effectively captures image gradients

Received: 15 July 2025 ; Accepted: 10 September 2025.
∗ Corresponding author. Email: alikhani@tabrizu.ac.ir.

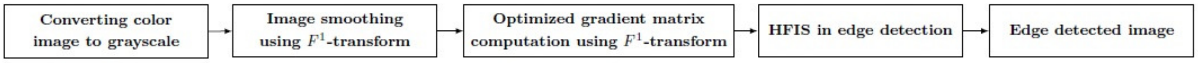


Figure 1. Structure of the proposed technique.

and intensity variations. Originally developed for single-variable functions [18], its extension to higher degrees and dimensions has enhanced its capability to model complex data patterns in image processing and pattern recognition [1, 2, 27]. Applications span approximation, filtering, and compression [1, 2, 9, 18, 19, 27]. Notably, it has been integrated into Canny edge detection by replacing traditional gradient computation with F^1-transform derivatives, followed by standard non-maximum suppression and thresholding, simplifying the algorithm and potentially improving performance [19].

Fuzzy inference systems (FIS) effectively manage real-world uncertainties in applications like control, decision support, and pattern recognition [6, 11]. The core FIS components are: fuzzification (converting crisp inputs to fuzzy sets), a rule base (if-then rules), an inference engine (applying rules), and defuzzification (producing crisp outputs). This process enables robust handling of imprecise data.

This study employs the F^1-transform with varying strides to generate multi-scale convolution masks for edge detection. These masks compute horizontal and vertical gradients, and are rotated by 45° to derive additional directional derivatives. The four directional derivatives are combined into an optimized gradient matrix by calculating the magnitude of intensity changes at each pixel. Different strides produce distinct gradient matrices, enabling multi-scale edge capture. We use the elements of different optimized gradient matrices as inputs to a fuzzy inference system and leverage the information contained in these matrices to make decisions or predictions based on fuzzy logic principles. A hierarchical fuzzy inference system (HFIS) extends traditional FIS by structuring inference across multiple layers, enhancing efficiency and scalability for complex systems. Unlike single-layer FIS, which processes all rules simultaneously, HFIS decomposes the problem into stages: each layer applies fuzzy rules to its inputs (either initial data or intermediate outputs from prior layers) and produces fuzzy outputs for subsequent processing. This reduces computational burden and improves interpretability but requires careful design of layer architecture and rule sets.

In this work, the HFIS comprises multiple layers of two-input, single-output FIS units. Initial inputs-elements from optimized gradient matrices-are fuzzified and processed in the first layer. Intermediate layers operate directly on fuzzy sets without repeated fuzzification/defuzzification, streamlining computation. Only the final output is defuzzified to yield a crisp edge detection result. This structure efficiently refines edge information while managing complexity. We called the thechnique mentioned above the HFIS edge detection method. The proposed HFIS edge detection method is evaluated against established techniques including Canny, fuzzy-based [16, 19, 23], and fractional-based methods [10, 14, 26]. Performance is assessed using standard metrics: accuracy (ACC), sensitivity (SEN), specificity (SPC), PSNR, SSIM, and FOM. The following advantages of the HFIS method are demonstrated:

- Adjustable image smoothness via F^1-transform kernels with variable strides.
- Multi-scale edge detection using optimized gradient matrices from different strides.
- Continuous edges through 4-directional derivative integration.
- Computational efficiency via HFIS with reduced rules and intermediate layer optimization.
- Improved accuracy through layered HFIS with adaptive rules and membership functions.
- Enhanced noise suppression using F^1-transform and hierarchical fuzzy processing.

2. Comprehensive theoretical analysis of the proposed algorithm

This section introduces a novel edge detection method for color images. This method utilizes the YIQ color model to extract the grayscale components from the RGB image, enabling more effective analysis. Additionally, the application of the F^1-transform aids in the creation of filters with varying sizes. These filters are then employed to both smooth the image and evaluate optimized gradient matrices. The resulting elements of these matrices serve as inputs for a fuzzy inference system. The algorithm's intricacies and steps are thoroughly explained and outlined in Figure 1, providing a comprehensive understanding of its implementation.

2.1. **Converting color image to grayscale.** Let $P = \{\big(R(x_k, y_l), G(x_k, y_i), B(x_k, y_l)\big) \mid x_k = 1, \cdots, N,\ y_l = 1, \cdots, M\}$ denote a 3rd-order tensor with $N \times M \times 3$ elements. This tensor represents a color image, where the kl-th component corresponds to the intensity values of red, green, and blue colors, respectively. In color image processing, the goal is to transform the color image into a grayscale image by separating the color information from the brightness information. Various algorithms exist for this transformation. In this paper, we adopt the YIQ color model, which entails the following representation:

$$\bar{P} = \{\big(Y(x_k, y_l), I(x_k, y_i), Q(x_k, y_l)\big) \mid x_k = 1, \cdots, N,\ y_l = 1, \cdots, M\},$$

where,

$$\begin{aligned} Y(x_k, y_l) &= 0.2999R(x_k, y_l) + 0.587G(x_k, y_l) + 0.114B(x_k, y_l), \\ I(x_k, y_l) &= 0.596R(x_k, y_l) - 0.275G(x_k, y_l) - 0.321B(x_k, y_l), \\ Q(x_k, y_l) &= 0.212R(x_k, y_l) - 0.523G(x_k, y_l) + 0.311B(x_k, y_l). \end{aligned}$$

Here, Y denotes the luminance of the image, while I and Q signify the color information.

2.2. **Image smoothing using F^1-transform.** Here, we apply the theoretical findings of the F^1-transform as presented in [2, 19]. By utilizing the F^1-transform, we can enhance our original image by generating masks or kernels specifically designed for smoothing purposes.

2.2.1. *The concepts of F^1-transform for two-variable functions.* From the literature, let fuzzy sets $\mathcal{B}_{k,l} : [a - h_u, b + h_u] \times [c - h_v, d + h_v] \to [0, 1]$ called basic functions and constitute an extended fuzzy partition of $\Omega = [a - h_u, b + h_u] \times [c - h_v, d + h_v]$.

Example 2.1. Throughout this study, we consider triangular basic functions $\{\mathcal{B}_{k,l}\}$ as defined below:

$$\mathcal{B}_{k,l}(u, v) = \begin{cases} (1 - \frac{|u-u_k|}{h_u})(1 - \frac{|v-v_l|}{h_v}) & (u, v) \in [u_k - h_u, u_k + h_u] \times [v_l - h_v, v_l + h_v], \\ 0, & o.w. \end{cases}.$$

Let $\mathcal{L}_2(\mathcal{B}_{k,l})$ denote the weighted L_2-space of bivariate functions with the kernel-weighted inner product:

$$\langle \mathfrak{f}, \mathfrak{g} \rangle_{\mathcal{B}_{k,l}} = \frac{1}{h^2} \iint_{\Omega_{k,l}} \mathfrak{f}(u, v)\mathfrak{g}(u, v)\mathcal{B}_{k,l}(u, v)\, dudv \tag{2.1}$$

where $\Omega_{k,l} = [u_{k-1}, u_{k+1}] \times [v_{l-1}, v_{l+1}]$ is the local support region.

Lemma 2.2. *Let $\mathfrak{f} \in \mathcal{L}_2(\mathcal{B}_{k,l})$. The fuzzy transform F^1 is defined as $F^1(\mathfrak{f}) = [F^1_{k,l}]_{p\times q}$ such that $F^1_{k,l}$ is defined as*

$$F^1_{k,l}(u, v) = c^{0,0}_{k,l} + c^{1,0}_{k,l}(u - u_k) + c^{0,1}_{k,l}(v - v_l),$$

in which the components of $c^{0,0}_{k,l}, c^{1,0}_{k,l}$, and $c^{0,1}_{k,l}$ are obtained as follows

$$c^{0,0}_{k,l} = \frac{1}{h_u h_v} \iint_{\Omega_{k,l}} \mathfrak{f}(u, v)\mathcal{B}_{k,l}(u, v)dudv, \tag{2.2}$$

$$c^{1,0}_{k,l} = \frac{6}{(h_u)^3 h_v} \iint_{\Omega_{k,l}} \mathfrak{f}(u, v)(u - u_k)\mathcal{B}_{k,l}(u, v)dudv, \tag{2.3}$$

$$c^{0,1}_{k,l} = \frac{6}{(h_v)^3 h_u} \iint_{\Omega_{k,l}} \mathfrak{f}(u, v)(v - v_l)\mathcal{B}_{k,l}(u, v)dudv. \tag{2.4}$$

Theorem 2.3. *For twice continuously differentiable functions $\mathfrak{f} \in \mathcal{L}_2(\mathcal{B}_{k,l})$ w.r.t. u and v in Ω. Then*

$$F^1_{k,l}(u, v) = \mathfrak{f}(u, v) + O\big(\max((h_u)^2, (h_v)^2)\big), \qquad (u, v) \in \Omega_{kl}.$$

Moreover, the coefficient approximations verify:

$$c^{0,0}_{k,l} = \mathfrak{f}(u_k, v_l) + O(\max((h_u)^2, (h_v)^2)),$$

$$c^{0,1}_{k,l} = \frac{\partial \mathfrak{f}}{\partial v}(u_k, v_l) + O\big(\max((h_u)^2, (h_v)^2)\big),$$

$$c_{k,l}^{1,0} = \frac{\partial \mathfrak{f}}{\partial u}(u_k, v_l) + O\big(\max((h_u)^2, (h_v)^2)\big).$$

Throughout this paper, we consider $h_u = h_v = h$.

2.2.2. *Image smoothing technique.* Let T be a normalized grayscale image matrix of Y with size $N \times M$. The kl-th element of T, denoted $T(x_k, y_l)$, is defined as the ratio of the corresponding element in Y to the maximum element in Y. Mathematically, this can be expressed as:

$$T(x_k, y_l) = \frac{Y(x_k, y_l)}{\max(Y(:))}, \quad \text{for} \quad x_k = 1, \cdots, N, \quad \text{and} \quad y_l = 1, \cdots, M.$$

Consider the extended triangular shaped basic functions $\mathcal{B}_{k,l}$ defined on windows of size $m \times m$ centered at (x_k, y_l), where $x_k = 1, \cdots, N$ and $y_l = 1, \cdots, M$. The strides are set as $h = \frac{m+1}{2}$. The windows are constructed by extending zero-padding to fill all empty areas of the image array. From the results of Theorem 2.3, we have

$$T(x_k, y_l) \approx c_{k,l}^{0,0}.$$

The coefficients $c_{k,l}^{0,0}$ for $k = 1, \cdots, N$ and $l = 1, \cdots, M$ can be obtained by substituting finite sums instead of integrals in Eq. (2.2) as shown below:

$$c_{k,l}^{0,0} = \frac{1}{h^2} \sum_{y=y_l-h}^{y_l+h} \sum_{x=x_k-h}^{x_k+h} T(x,y)\mathcal{B}_{k,l}(x,y). \tag{2.5}$$

We assume $h = 3$ which results in a window size of 5×5. By employing Eq. (2.5) and Example 2.1, we obtain the following mask with a size of 5×5.

$$\mathcal{W}^5 = \frac{1}{9}\begin{pmatrix} \frac{1}{9} & \frac{2}{9} & \frac{1}{3} & \frac{2}{9} & \frac{1}{9} \\ \frac{2}{9} & \frac{4}{9} & \frac{2}{3} & \frac{4}{9} & \frac{2}{9} \\ \frac{1}{3} & \frac{2}{3} & 1 & \frac{2}{3} & \frac{1}{3} \\ \frac{2}{9} & \frac{4}{9} & \frac{2}{3} & \frac{4}{9} & \frac{2}{9} \\ \frac{1}{9} & \frac{2}{9} & \frac{1}{3} & \frac{2}{9} & \frac{1}{9} \end{pmatrix}.$$

For any given value of h, we can create masks $\mathcal{W}^m$ of size $m \times m$. $\bar{T}_m$ (the smoothed image) can then be obtained by convolving T with the mask $\mathcal{W}^m$. This convolution operation is denoted by $*$, and mathematically expressed as:

$$\bar{T}_m = T * \mathcal{W}^m.$$

Here, $\bar{T}_m$ represents the smoothed version of the image T using the mask $\mathcal{W}^m$ and $\bar{T}_m(x_k, y_l) = c_{k,l}^{0,0}$.

2.3. **Optimized gradient matrix computation using F^1-transform.** In this section, we enhance the methodology by applying F^1-transform to derive four distinct masks. Utilizing these kernels, we are able to generate an optimized gradient matrix.

Based on the findings of Theorem 2.3, it can be concluded

$$\frac{\partial \bar{T}_m}{\partial x}(x_k, y_l) \approx c_{k,l}^{1,0}, \qquad \frac{\partial \bar{T}_m}{\partial y}(x_k, y_l) \approx c_{k,l}^{0,1}.$$

The coefficients $c_{k,l}^{0,1}$ and $c_{k,l}^{1,0}$, where $k = 1, \cdots, N$ and $l = 1, \cdots, M$, are derived by substituting finite sums instead of integrals in Eqs. (2.3)-(2.4) in the following manner:

$$c_{k,l}^{0,1} = \frac{6}{h^4} \sum_{y=y_l-h}^{y_l+h} \sum_{x=x_k-h}^{x_k+h} \bar{T}_m(x,y)(y - y_l)\mathcal{B}_{k,l}(x,y), \tag{2.6}$$

$$c_{k,l}^{1,0} = \frac{6}{h^4} \sum_{y=y_l-h}^{y_l+h} \sum_{x=x_k-h}^{x_k+h} \bar{T}_m(x,y)(x - x_k)\mathcal{B}_{k,l}(x,y). \tag{2.7}$$

By assuming $h = 3$, utilizing Eqs. (2.6)-(2.7) and Example 2.1, we can derive the following masks with a size of 5×5:

$$\mathcal{W}_0^5 = \frac{2}{27}\begin{pmatrix} -\frac{2}{9} & -\frac{4}{9} & -\frac{2}{3} & -\frac{4}{9} & -\frac{2}{9} \\ -\frac{2}{9} & -\frac{4}{9} & -\frac{2}{3} & -\frac{4}{9} & -\frac{2}{9} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{9} & \frac{4}{9} & \frac{2}{3} & \frac{4}{9} & \frac{2}{9} \\ \frac{2}{9} & \frac{4}{9} & \frac{2}{3} & \frac{4}{9} & \frac{2}{9} \end{pmatrix}, \qquad \mathcal{W}_{90}^5 = \frac{2}{27}\begin{pmatrix} -\frac{2}{9} & -\frac{2}{9} & 0 & \frac{2}{9} & \frac{2}{9} \\ -\frac{4}{9} & -\frac{4}{9} & 0 & \frac{4}{9} & \frac{4}{9} \\ -\frac{2}{3} & -\frac{2}{3} & 0 & \frac{2}{3} & \frac{2}{3} \\ -\frac{4}{9} & -\frac{4}{9} & 0 & \frac{4}{9} & \frac{4}{9} \\ -\frac{2}{9} & -\frac{2}{9} & 0 & \frac{2}{9} & \frac{2}{9} \end{pmatrix}.$$

The masks $\mathcal{W}_{45}^5$ and $\mathcal{W}_{135}^5$ are derived by rotating the masks $\mathcal{W}_0^5$ and $\mathcal{W}_{90}^5$ by $45°$, respectively.

$$\mathcal{W}_{45}^5 = \frac{2}{27}\begin{pmatrix} -\frac{2}{3} & -\frac{4}{9} & -\frac{2}{9} & -\frac{2}{9} & 0 \\ -\frac{4}{9} & -\frac{2}{3} & -\frac{4}{9} & 0 & \frac{2}{9} \\ -\frac{2}{9} & -\frac{4}{9} & 0 & \frac{4}{9} & \frac{2}{9} \\ -\frac{2}{9} & 0 & \frac{4}{9} & \frac{2}{3} & \frac{4}{9} \\ 0 & \frac{2}{9} & \frac{2}{9} & \frac{4}{9} & \frac{2}{3} \end{pmatrix}, \qquad \mathcal{W}_{135}^5 = \frac{2}{27}\begin{pmatrix} 0 & -\frac{2}{9} & -\frac{2}{9} & -\frac{4}{9} & -\frac{2}{3} \\ \frac{2}{9} & 0 & -\frac{4}{9} & -\frac{2}{3} & -\frac{4}{9} \\ \frac{2}{9} & \frac{4}{9} & 0 & -\frac{4}{9} & -\frac{2}{9} \\ \frac{4}{9} & \frac{2}{3} & \frac{4}{9} & 0 & -\frac{2}{9} \\ \frac{2}{3} & \frac{4}{9} & \frac{2}{9} & \frac{2}{9} & 0 \end{pmatrix}.$$

To obtain the directional derivatives of an image P in four main directions, we can create two masks, $\mathcal{W}_0^m$ and $\mathcal{W}_{90}^m$, with sizes $m \times m$ for any arbitrary value of h. We can then rotate these masks by $45°$ to obtain $\mathcal{W}_{45}^m$ and $\mathcal{W}_{135}^m$, respectively. By convolving these masks with the image P, we can obtain the directional derivatives in four main directions through the following convolutions:

$$\mathcal{P}_0^m = \bar{T}_m * \mathcal{W}_0^m, \qquad \mathcal{P}_{45}^m = \bar{T}_m * \mathcal{W}_{45}^m, \qquad \mathcal{P}_{90}^m = \bar{T}_m * \mathcal{W}_{90}^m, \qquad \mathcal{P}_{135}^m = \bar{T}_m * \mathcal{W}_{135}^m.$$

In fact, $\mathcal{P}_0^m(x_k, y_l) = c_{k,l}^{1,0}$ and $\mathcal{P}_{90}^m(x_k, y_l) = c_{k,l}^{0,1}$.

The optimized gradient matrix of image P is denoted as $E_m = [e_{ij}^m]$, where the elements e_{ij}^m are defined as:

$$e_{ij}^m = \max(|(\mathcal{P}_0^m)_{ij}| + |(\mathcal{P}_{90}^m)_{ij}|, |(\mathcal{P}_{45}^m)_{ij}| + |(\mathcal{P}_{135}^m)_{ij}|), \qquad i = 1, \cdots, N, \; j = 1, \cdots, M.$$

In the context of edge detection, masks with a small size can extract edge information that involves both strong and weak edges. However, these small masks are more sensitive to noise and tend to classify noisy pixels as edges. On the other hand, masks with a larger size can extract more general edges and are less affected by noise. To address these issues and achieve a more accurate detection of edge locations, we combine gradient matrices obtained from kernels with different sizes. This allows us to reduce the impact of noise and enhance the accuracy of edge detection. The steps involved in evaluating the image smoothed matrix and the optimized gradient matrix are summarized in Algorithm 1.

2.4. **Hierarchical fuzzy inference system in edge detection.** In this step, we utilize a fuzzy inference system to enhance image edge detection. We provide a concise overview of the fundamental principles of the fuzzy inference system. In order to accomplish greater precision and computational efficiency, we introduce the concept of the HFIS. The HFIS offers enhanced capability in handling the inherent uncertainty associated with edge positions in images, surpassing the capabilities of conventional fuzzy inference systems.

2.4.1. *Multi-input single-output fuzzy inference system.* The multi-input single-output (MISO) fuzzy inference system utilizes the ij-th elements of optimized gradient matrices E_m, where $m \in \{3, 5, 7, 9, \cdots\}$, as crisp inputs. To avoid confusion, we denote e_{ij}^m as e^m and fuzzify them accordingly. These fuzzified inputs are then passed through the inference engine, which utilizes a fuzzy rule base to derive the fuzzy output. Subsequently, the fuzzy output is defuzzified to generate a crisp output. These outputs enable the identification and determination of the image edges. The architecture of the multi-input single-output fuzzy inference system is illustrated in Figure 2. The fuzzy inference system establishes a relationship between fuzzy rules and generates conclusions for fuzzy inputs through IF-THEN rules comprised of antecedents and consequences as fuzzy sets. As an illustration, we can examine a fuzzy inference system in the following format:

Algorithm 1:

Steps of image smoothing and calculation of the optimized gradient matrix ;
Input : Color Image P
Output: Gradient matrix E_m
Read color image P
Utilize the following formula to compute the brightness information of an image:
$Y = 0.2999R + 0.587G + 0.114B.$
Normalize each component of the array $Y = [Y_{x_k y_l}]$ by utilizing the following formula:
$T(x_k, y_l) = \frac{Y(x_k,y_l)}{max(Y(:))}, \quad x_k = 1, \cdots, N, \ y_l = 1, \cdots, M.$
Choose an odd number $m > 1$ and generate a convolution mask $\mathcal{W}^m$.
Smooth the image T using the convolution mask $\mathcal{W}^m$ to obtain the smoothed image $\bar{T}_m$ by performing the following convolution operation: $\bar{T}_m = T * \mathcal{W}^m$.
Choose an odd integer $m > 1$ and generate four convolution masks: $\mathcal{W}_0^m, \mathcal{W}_{45}^m, \mathcal{W}_{90}^m, \mathcal{W}_{135}^m$.
Calculate the directional derivatives of the image P in the four main directions using the following formulas:
$\mathcal{P}_0^m = \bar{T}_m * \mathcal{W}_0^m, \quad \mathcal{P}_{45}^m = \bar{T}_m * \mathcal{W}_{45}^m, \quad \mathcal{P}_{90}^m = \bar{T}_m * \mathcal{W}_{90}^m, \quad \mathcal{P}_{135}^m = \bar{T}_m * \mathcal{W}_{135}^m.$
Calculate the optimized gradient matrix $E_m = [e_{ij}^m]$ of image P using the following formula:
$e_{ij}^m = \max(|(\mathcal{P}_0^m)_{ij}| + |(\mathcal{P}_{90}^m)_{ij}|, |(\mathcal{P}_{45}^m)_{ij}| + |(\mathcal{P}_{135}^m)_{ij}|) \quad i = 1, \cdots, N, \ j = 1, \cdots, M.$

$$\begin{array}{l} \text{Premise :IF } x \in A_i \text{ and } y \in B_i \text{ THEN } z \in C_i, \ i = 1, \cdots, n \\ \text{Fact : } x \in A' \text{ and } y \in B' \\ \hline \text{Conclusion } z \in C' \end{array}$$

In this structure, $A_i, A' \in \mathcal{F}(\mathcal{X})$, $B_i, B' \in \mathcal{F}(\mathcal{Y})$, $C_i \in \mathcal{F}(\mathcal{Z})$. Here, $\mathcal{F}(\mathcal{X})$ represents the collection of fuzzy subsets of $\mathcal{X}$. $C' \in \mathcal{F}(\mathcal{Z})$ can be interpreted using various methods such as Mamdani inference, Larsen inference, t-norm-based inference, etc. Throughout this paper, we employ t-norm-based inference system with a max-t-norm rule base which is defined as follows:

$$C'(z) = \bigvee_{x \in \mathcal{X}, y \in \mathcal{Y}} (A'(x) \wedge B'(y)) \star_t \mathcal{R}(x, y, z),$$

where $\mathcal{R} \in \mathcal{F}(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})$ is being used to interpret the fuzzy rule base in the premise and $\star_t$ is arbitrary t-norm.

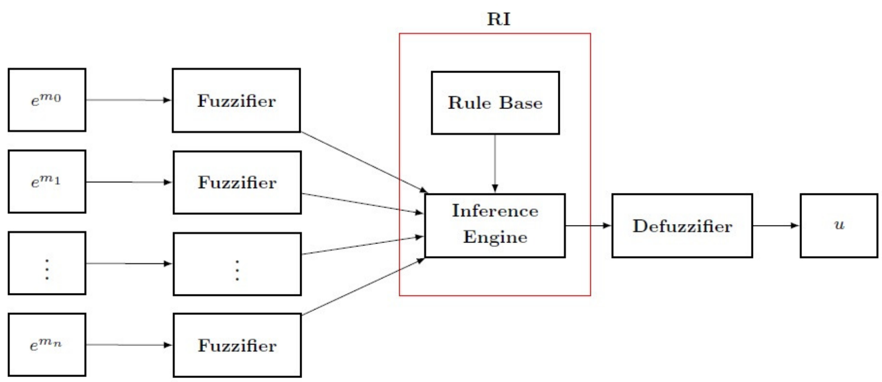


FIGURE 2. Structure of MISO fuzzy inference system.

2.4.2. *Hierarchical fuzzy inference system.* To enhance the efficiency of the MISO fuzzy inference system (Figure 2), we propose the utilization of the HFIS depicted in Figure 3. This hierarchical structure serves to simplify the fuzzy inference system, particularly in scenarios with higher dimensions. The proposed system consists of n layers, where j-th layer contains $n+1-j$ two-input single-output RIs. Each RI comprises an inference engine and a rule base, enabling effective processing and decision-making. In the first layer, we apply the fuzzified elements of optimized gradient matrices E_3 and E_m (for arbitrary $m \in \{5, 7, 9, \cdots\}$) as inputs to the RIs. The first layer serves as a foundation for subsequent processing. To streamline the system and eliminate unnecessary computations, we omit the fuzzifier and defuzzifier in the middle layers. This helps to improve efficiency by reducing the computational burden and simplifying the overall process. The inputs of the RI_i^j (representing the i-j-th RI) are determined by the outputs of the RI_i^{j-1} and the RI_{i+1}^{j-1} in the previous layer. This design allows for effective information propagation between layers, maximizing the flow and utilization of information throughout the system. Finally, the output of the last layer is defuzzified to determine the edges of the image.

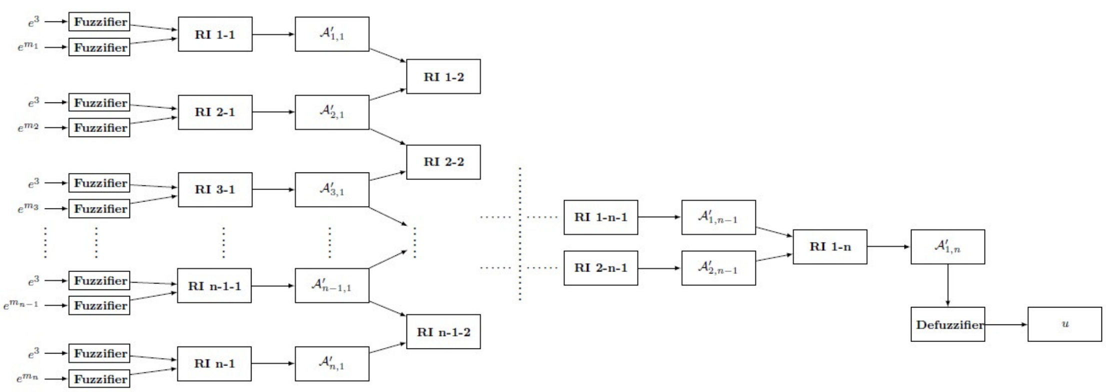


FIGURE 3. Structure of the HFIS.

2.4.3. *Fuzzification.* Based on the preceding discussion and Figure 3, it is evident that e^3 and e^m (where $m \in \{5, 7, 9, \cdots\}$) serve as the inputs of the RI_i^1 and need to be fuzzified. Fuzzification is the process of mapping crisp values into fuzzy sets. In this context, we employ a singleton fuzzifier approach, which is generally preferred in real-time systems due to its simplicity and popularity. Hence, we fuzzify e^m for $m \in \{3, 5, 7, 9, \cdots\}$ using the singular membership function $A'_m(t) = \chi_{\{e^m\}}(t)$.

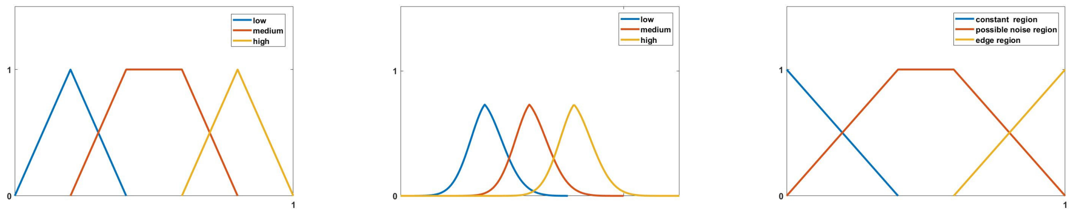
2.4.4. *Fuzzy Rule Base.* In the following, we define the fuzzy rule bases that will be utilized in the different layers of the HFIS.

First layer: The rule base for the fuzzy inference system in the first layer consists of antecedents $A^m_{\mathcal{L}}$, $A^m_{\mathcal{M}}$, and $A^m_{\mathcal{H}}$, which represent the membership functions for linguistic terms "low", "medium", and "high" respectively. These membership functions are defined for $m \in \{3, 5, 7, 9, \cdots\}$ and are illustrated in Figure 4(a).

$$
\begin{aligned}
A^m_{\mathcal{L}} &=< 0, \min(E_m), \min(E_m) + \alpha \max(E_m) >, \\
A^m_{\mathcal{M}} &=< \beta_1 \mathcal{E}^m - \alpha \max(E_m), \beta_1 \mathcal{E}^m, \beta_2 \mathcal{E}^m, \beta_2 \mathcal{E}^m + \alpha \max(E_m) >, \\
A^m_{\mathcal{H}} &=< (1-\alpha) \max(E_m), \max(E_m), 1 >,
\end{aligned}
$$

where $\mathcal{E}^m = \min(E_m) + \max(E_m)$ and $\alpha, \beta_1, \beta_2 > 0$.

Remark 2.4. We represent trapezoidal and triangular fuzzy numbers as $\mu_1 =< a, b, c, d >$ and $\mu_2 =< a, b, c >$, respectively.



(a) Antecedents of the first layer. (b) Antecedents of the middle layers. (c) Consequences of the layers.

FIGURE 4. Example of fuzzy values of three linguistic variables in the antecedents and consequences of the layers.

In the image classification process, the image is divided into three main regions: constant region, edge region, and possible noise region. In the constant region, all pixels have almost the same intensity. In the edge region, the pixels located on edges have similar intensity values in the edge directions, while their values differ significantly from those of the other surrounding pixels. The possible noise region is characterized by intensity values that differ from the neighboring pixels for both edge and non-edge pixels. The intensity values of the neighborhoods surrounding non-edge pixels are nearly the same, while the intensity values of the neighborhoods surrounding edge pixels in the edge directions are close to each other. The consequences of the rule base in the first layer of the fuzzy inference system are represented as B_1^m, B_2^m, and B_3^m, which model the membership functions for the linguistic terms "constant region", "edge region", and "possible noise region" respectively. These membership functions are defined in the following for $m \in \{5, 7, 9, \cdots\}$ and visualized in Figure 4(c).

$$B_1^m =< 0, 0, \min\{E_3, E_m\} >,$$
$$B_2^m =< 1 - \min\{E_3, E_m\}, 1, 1 >,$$
$$B_3^m =< 0, \nu_1 \max\{E_3, E_m\}, \nu_2 \max\{E_3, E_m\}, 1 >,$$

here, $\nu_1, \nu_2 > 0$.

By utilizing the antecedents and consequences provided earlier, we can establish the following rule base for the first layer of the fuzzy inference systems.

Rule 1: IF x_1 is $A^3_{\mathcal{L}}$ and x_2 is $A^m_{\mathcal{L}}$ THEN y is B_1^m.
Rule 2: IF x_1 is $A^3_{\mathcal{M}}$ and x_2 is $A^m_{\mathcal{M}}$ THEN y is B_2^m.
Rule 3: IF x_1 is $A^3_{\mathcal{H}}$ and x_2 is $A^m_{\mathcal{H}}$ THEN y is B_2^m.
Rule 4: IF x_1 is $A^3_{\mathcal{M}}$ and x_2 is $A^m_{\mathcal{L}}$ THEN y is B_3^m.
Rule 5: IF $x_1 \in A^3_{\mathcal{H}}$ and $x_2 \in A^m_{\mathcal{M}}$ THEN $y \in B_3^m$.
Rule 6: IF x_1 is $A^3_{\mathcal{H}}$ and x_2 is $A^m_{\mathcal{L}}$ THEN y is B_3^m.

To simplify the analysis, we assign numerical values to the linguistic terms $\mathcal{L}, \mathcal{M}, \mathcal{H}$ as $\mathcal{L} = 1$, $\mathcal{M} = 2$, and $\mathcal{H} = 3$. The following fuzzy relation is applied to interpret the fuzzy rule base in the first layer.

$$\begin{aligned}\mathcal{R}(x_1, x_2, y) = &\Big(\big(A_1^3(x_1) \wedge A_1^m(x_2)\big) \star_t B_1^m(y)\Big) \vee \Big(\big(A_2^3(x_1) \wedge A_2^m(x_2)\big) \star_t B_2^m(y)\Big) \\ &\vee \Big(\big(A_3^3(x_1) \wedge A_3^m(x_2)\big) \star_t B_2^m(y)\Big) \vee \Big(\vee_{i_1=2}^{3} \vee_{i_2=1}^{i_1-1} \big(A_{i_1}^3(x_1) \wedge A_{i_2}^m(x_2)\big) \star_t B_3^m(y)\Big),\end{aligned}$$

where $\star_t$ denotes arbitrary t-norm.

We classify the elements of the optimized gradient matrix E_m for $m \in \{3, 5, 7, 9, \dots\}$ into three categories: "low," "medium," and "high," based on the membership functions illustrated in Figure 4. To determine the region of each pixel in an image, we analyze the values of e^3 and e^m for $m \in \{5, 7, 9, \dots\}$. If these values correspond to the ij-th pixel being classified as "low," then we assign the pixel to the constant region. Conversely, if the values correspond to the

ij-th pixel being classified as "medium" or "high," then we assign the pixel to the edge region. We can leverage large-sized convolution masks to reduce the impact of noise. By examining the values of e^3 and e^m for $m \in \{5, 7, 9, \dots\}$, if these values do not fall into the same class, we classify the pixel as part of the possible noise region.

Middle layers: The antecedents of the rule base in the middle layers consist of three components: $\mathcal{A}_{\mathcal{L}}^{i,j}$, $\mathcal{A}_{\mathcal{M}}^{i,j}$, and $\mathcal{A}_{\mathcal{H}}^{i,j}$. These components represent membership functions that model the linguistic terms "low", "medium", and "high", respectively. The membership functions are defined as follows and can be visually depicted in Figure 4(b).

$$
\begin{aligned}
\mathcal{A}_{\mathcal{L}}^{i,j} &= \min(e^{-\frac{(x-(1-\lambda)\rho_{\mathcal{L}})^2}{2\sigma^2}}, e^{-\frac{(x-\rho_{\mathcal{L}})^2}{2\sigma^2}}),\\
\mathcal{A}_{\mathcal{M}}^{i,j} &= \min(e^{-\frac{(x-(1-\lambda)\rho_{\mathcal{M}})^2}{2\sigma^2}}, e^{-\frac{(x-\rho_{\mathcal{M}})^2}{2\sigma^2}}),\\
\mathcal{A}_{\mathcal{H}}^{i,j} &= \min(e^{-\frac{(x-(1-\lambda)\rho_{\mathcal{H}})^2}{2\sigma^2}}, e^{-\frac{(x-\rho_{\mathcal{H}})^2}{2\sigma^2}}),
\end{aligned}
$$

here

$$
\begin{aligned}
\rho_{\mathcal{L}} &= \min\{\min(E_3), \min(E_{m_i}), \cdots, \min(E_{m_{i+j}})\},\\
\rho_{\mathcal{M}} &= \frac{1}{2}\text{median}\{\mathcal{E}^3, \mathcal{E}^{m_i}, \cdots, \mathcal{E}^{m_{i+j}}\},\\
\rho_{\mathcal{H}} &= \max\{\max(E_3), \max(E_{m_i}), \cdots, \max(E_{m_{i+j}})\},
\end{aligned}
$$

and $\sigma = \frac{\rho_{\mathcal{H}}}{\gamma}$, $0 < \lambda < 1$, $\gamma > 0$.

The consequences of the rule base for the fuzzy inference system in the middle layers correspond to linguistic terms such as "constant region", "edge region", and "possible noise region". These terms are represented by the membership functions $\mathcal{B}_1^{i,j}$, $\mathcal{B}_2^{i,j}$, and $\mathcal{B}_3^{i,j}$ respectively. The membership functions are defined as follows and can be visualized in Figure 4(c).

$$
\begin{aligned}
\mathcal{B}_1^{i,j} &=< 0, 0, \rho_{\mathcal{L}} >,\\
\mathcal{B}_2^{i,j} &=< 1-\rho_{\mathcal{L}}, 1, 1 >,\\
\mathcal{B}_3^{i,j} &=< 0, \nu_1\rho_{\mathcal{H}}, \nu_2\rho_{\mathcal{H}}, 1 >,
\end{aligned}
$$

where $\nu_1, \nu_2 > 0$.

Based on the antecedents and consequences described earlier, the following rule base is defined in the i-th fuzzy inference engine within the j-th middle layer.

Rule 1: IF x_1 is $\mathcal{A}_{\mathcal{L}}^{i,j-1}$ and x_2 is $\mathcal{A}_{\mathcal{L}}^{i+1,j-1}$ THEN y is $\mathcal{B}_1^{i,j}$.
Rule 2: IF x_1 is $\mathcal{A}_{\mathcal{M}}^{i,j-1}$ and x_2 is $\mathcal{A}_{\mathcal{M}}^{i+1,j-1}$ THEN y is $\mathcal{B}_2^{i,j}$.
Rule 3: IF x_1 is $\mathcal{A}_{\mathcal{H}}^{i,j-1}$ and x_2 is $\mathcal{A}_{\mathcal{H}}^{i+1,j-1}$ THEN y is $\mathcal{B}_2^{i,j}$.
Rule 4: IF x_1 is $\mathcal{A}_{\mathcal{M}}^{i,j-1}$ and x_2 is $\mathcal{A}_{\mathcal{L}}^{i+1,j-1}$ THEN $y \in \mathcal{B}_3^{i,j}$.
Rule 5: IF x_1 is $\mathcal{A}_{\mathcal{H}}^{i,j-1}$ and x_2 is $\mathcal{A}_{\mathcal{M}}^{i+1,j-1}$ THEN y is $\mathcal{B}_3^{i,j}$.
Rule 6: IF x_1 is $\mathcal{A}_{\mathcal{H}}^{i,j-1}$ and x_2 is $\mathcal{A}_{\mathcal{L}}^{i+1,j-1}$ THEN y is $\mathcal{B}_3^{i,j}$.

The following fuzzy relation is utilized to interpret the fuzzy rule base mentioned above.

$$
\begin{aligned}
\mathcal{R}(x_1, x_2, y) = &\Big(\big(\mathcal{A}_1^{i,j-1}(x_1) \wedge \mathcal{A}_1^{i+1,j-1}(x_2)\big) \star_t \mathcal{B}_1^{i,j}(y)\Big)x \vee \Big(\big(\mathcal{A}_2^{i,j-1}(x_1) \wedge \mathcal{A}_2^{i+1,j-1}(x_2)\big) \star_t \mathcal{B}_2^{i,j}(y)\Big)\\
&\vee \Big(\big(\mathcal{A}_3^{i,j-1}(x_1) \wedge \mathcal{A}_3^{i+1,j-1}(x_2)\big) \star_t \mathcal{B}_2^{i,j}(y)\Big) \vee \Big(\vee_{i_1=2}^{3}\vee_{i_2=1}^{i_1-1}\big(\mathcal{A}_{i_1}^{i,j-1}(x_1) \wedge \mathcal{A}_{i_2}^{i+1,j-1}(x_2)\big) \star_t \mathcal{B}_3^{i,j}(y))\Big),
\end{aligned}
$$

where $\star_t$ denotes arbitrary t-norm.

The fuzzy rule bases defined in the layers of the HFIS have been proven experimentally to exhibit high coverage and accuracy. This has been achieved through rigorous experimentation on diverse datasets.

Table 1. Parameters of the HFIS edge detection method.

Antecedents of the first layer			Antecedents of the middle layers		Consequences of the layers	
α	β_1	β_2	λ	γ	ν_1	ν_2
0.6	0.8	1.5	0.1	2	0.5	1.5

2.4.5. *Mamdani inference system.* Due to the simplicity and advantageous computational properties of Mamdani inference, we employ the following Mamdani inference method in the i-th inference engine within the first layer.

$$\mathcal{A}'_{i,1}(y) = \bigvee_{x_1,x_2\in[0,1]} \left(A'_3(x_1) \wedge A'_{m_i}(x_2)\right) \star_t \mathcal{R}(x_1,x_2,y) = \mathcal{R}(e^3, e^{m_i}, y).$$

Furthermore, the following Mamdani inference method is applied to the i-th inference engine within the j-th middle layer.

$$\mathcal{A}'_{i,j}(y) = \bigvee_{x_1,x_2\in[0,1]} \left(\mathcal{A}'_{i,j-1}(x_1) \wedge \mathcal{A}'_{i+1,j-1}(x_2)\right) \star_t \mathcal{R}(x_1,x_2,y).$$

2.4.6. *Defuzzification.* In the last layer of the HFIS, the fuzzy output of the fuzzy inference engine is subjected to defuzzification. Specifically, we employ the center of gravity defuzzification method to handle this task. The center of gravity defuzzification is implemented as follows:

$$u = \mathrm{COG}(\mathcal{A}'_{1,n}) = \frac{\int_0^1 y.\mathcal{A}'_{1,n}(y)dy}{\int_0^1 \mathcal{A}'_{1,n}(y)dy}.$$

Remark 2.5. We have employed the aforementioned process to calculate the value u for each element within the image matrix. By executing this process, we derive $u_{ij} = u$ where $i = 1, \cdots, N$ and $j = 1, \cdots, M$. In the subsequent section, we will analyze the edge-detected image U, which is defined as $U = [u_{ij}]_{N\times M}$.

3. Evaluating the quality of the HFIS edge detection method

The performance of the HFIS edge detection method is analyzed in terms of accuracy and sensitivity to noise. To assess its effectiveness, we evaluate several criteria including ACC (Accuracy), SEN (Sensitivity), SPC (Specificity), PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and FOM (Figure of Merit). In order to explore the method's sensitivity to different types of noise, we specifically consider Gaussian noise and salt and pepper noise. The test images used in our analysis are sourced from the BSDS500 dataset, which can be accessed at https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. These images have dimensions of 321×481 and 481×321. Additionally, we utilize images from the CVG-UGR image dataset available at http://decsai.ugr.es/cvg/dbimagenes/. These images have dimensions of 256×256 and 512×512. During the experiment, we employ either the min-t-norm or algebraic product as the t-norm operator. The optimal parameters of the HFIS edge detection method have been determined through experimental analysis. The following parameters have been found to yield the best results.

3.1. **ACC, SEN, and SPC measures and experimental results.** To ensure the accuracy of the HFIS edge detection method, we assess its performance by comparing the resulting edge detected image to a ground truth image. To establish the ground truth, we start by using the Canny edge detection algorithm provided by MATLAB's built-in functions. In order to achieve this comparison, we utilize accuracy, sensitivity, and specificity measures as described below:

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \times 100\%,$$

$$\mathrm{SEN} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \times 100\%,$$

Table 2. The average values of ACC, SEN, and SPC for the HFIS method.

dataset	ACC	SEN	SPC
CVG-UGR	0.9226	0.9081	0.9313
BSDS500	0.9018	0.9184	0.9008

$$\mathrm{SPC} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \times 100\%.$$

The variables TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively. The high values for ACC suggest that there is a strong agreement between the detected edges and the actual edges in the ground truth. This means that the method is able to accurately identify the edges in the images. Furthermore, the high SEN values indicate that the method is able to capture a large proportion of the true edges present in the images. This means that the method is successful in identifying most of the actual edges in the images. The high SPC values suggest that there is a low rate of false edge detections. This means that the method is reliable in distinguishing between true edges and non-edges, and it does not erroneously identify non-edges as edges. To implement these criteria, we apply a binarization process to the output of the HFIS. The binarization is carried out using the following transformation:

$$\hat{\mathrm{u}} = \begin{cases} 1, & u > \mathcal{T}, \\ 0, & o.w. \end{cases}$$

where $\mathcal{T}$ represents the threshold value. The output of the HFIS edge detection method is converted into a binary image using this technique.

Here, we present the complete evaluation results of ACC, SEN, and SPC measures for both the CVG-UGR and BSDS500 datasets. These criteria offer a comprehensive evaluation of the performance of the HFIS method on both of these datasets. Table 2 indicates that all three values of ACC, SEN, and SPC are high. By analyzing the results in this table, it can be concluded that the HFIS edge detection method is effective in accurately identifying and delineating edges in images.

Now, our objective is to compare our technique with the methods introduced in [10, 14, 19]. We will accomplish this by evaluating the performance of the methods using two test images, namely House and Airplane. The House image has dimensions of 256×256, while the Airplane image has dimensions of 481×321. Figures 5(a) and 6(a) depict these images respectively. To establish a baseline for comparison, we have the ground truth images for House and Airplane, which are illustrated in Figures 5(b) and 6(b), respectively.

The fuzzy-based technique introduced in [19] integrates the F^1-transform into the Canny edge detection process. The combined technique involves applying the F^1-transform to the image, which helps derive the first partial derivatives in two directions. Following this, we continue with the remaining steps of the Canny algorithm, including non-maximum suppression and hysteresis thresholding. These steps help identify and refine the detected edges. We observe the results of this combined approach in Figures 5(c) and 6(c).

The techniques introduced in [10, 14] utilize fractional-order masks to derive directional derivatives in two directions: the horizontal and vertical. This results in computing the gradient matrix. We present a visualization of the binarized outcomes obtained from the fractional approach proposed in [14], depicted in Figures 5(d) and 6(d). Additionally, we showcase the binarized results achieved using the fractional approach introduced in [10], illustrated in Figures 5(e) and 6(e).

Furthermore, we apply the HFIS edge detection method and obtain binarized output images as shown in Figures 5(d) and 6(d). The visual evidence presented in Figures 5 and 6 supports the conclusion that the performance of the HFIS method significantly surpasses that of the other mentioned methods in achieving seamless and unbroken edges. To critically assess the effectiveness of the HFIS edge detection method, the fuzzy-based method and the fractional-based methods, we analyze three key evaluation metrics: ACC, SEN, and SPC. The ACC, SEN, and SPC values obtained in Table 3 demonstrate that the HFIS edge detection method outperforms the other mentioned methods, as it consistently achieves better continuous edges. The HFIS edge detection method is superior to the other methods

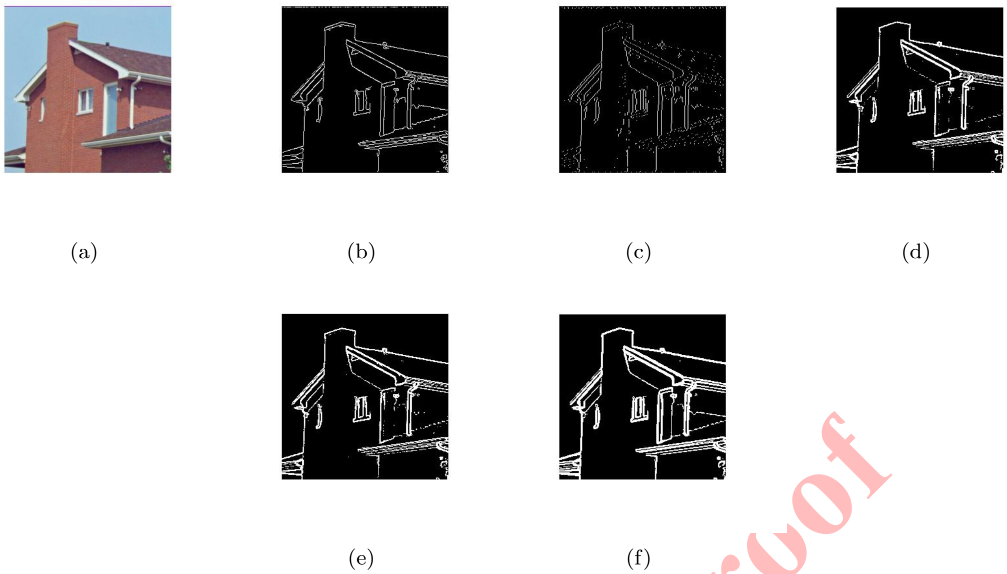


FIGURE 5. (a) Original image (b) Ground truth (c) Fuzzy-based method [19] (d) Fractional-based method [14] (e) Fractional-based method [10] (f) The HFIS edge detection method.

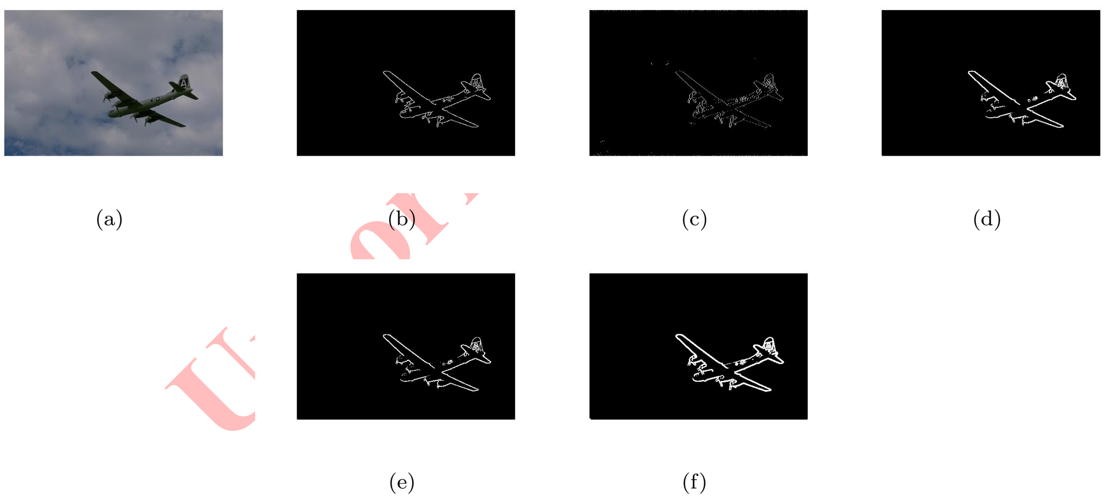


FIGURE 6. (a) Original image (b) Ground truth (c) Fuzzy-based method [19] (d) Fractional-based method [14] (e) Fractional-based method [10] (f) The HFIS edge detection method.

because it uses directional derivatives in four directions, leading to an optimized gradient matrix. This allows it to capture subtle variations in intensity more effectively. On the other hand, the hierarchical fuzzy inference system in our technique combines the information of various optimized gradient matrices, obtained through the use of F^1-transform with variable strides, within a hierarchical structure. This innovative approach ultimately enables accurate determination of edge locations.

Table 3. Comparison of ACC, SEN, and SPC values of the methods in [10, 14, 19] and the HFIS method.

	method [19]			method [14]			method [10]			HFIS method		
Images	ACC	SEN	SPC	ACC	SEN	SPC	ACC	SEN	SPC	ACC	SEN	SPC
House	0.967	0.430	0.991	0.935	0.755	0.945	0.939	0.817	0.946	0.901	0.918	0.900
Airplane	0.985	0.491	0.990	0.982	0.782	0.991	0.991	0.779	0.993	0.982	0.937	0.982

3.2. **PSNR and SSIM measures and experimental results.** We aim to evaluate the robustness of our algorithm against noise by adding two types of noise, Gaussian and salt and pepper, to the original image. To assess the performance of our method in detecting edges in noisy images, we utilize the peak signal to noise ratio (PSNR) and the structural similarity index measure (SSIM). PSNR compares the strength of strong and weak edges in noisy corrupted images. Higher PSNR values correspond to smaller differences between the edge-detected image of the original image and the noisy image, indicating better preservation of edge information in the presence of noise. PSNR is calculated using the following formula:

$$\mathrm{PSNR} = 10\log_{10}\frac{M \times N \times (\max(U))^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}(u_{ij}-\bar{u}_{ij})^2},$$

where $U = [u_{ij}]_{N\times M}$ is the edge-detected image of the original image, $\bar{U} = [\bar{u}_{ij}]_{N\times M}$ is the edge-detected image of the noisy image, and $\max(U) = \max\{u_{i,j} | i = 1, \cdots, N, j = 1, \cdots, M\}$.
Furthermore, SSIM provides a quantitative measure of how well the structural information is preserved between two images. Higher SSIM values indicate better preservation of structural information, while lower values indicate a greater distortion or loss of information. SSIM is expressed by the following equation:

$$\mathrm{SSIM} = \frac{(2\mu_1\mu_2 + C_1)(\sigma_{12} + C_2)}{(\mu_1^2 + \mu_1^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)}, \qquad C_1 = (0.01 \times \max(U))^2, C_2 = (0.03 \times \max(U))^2.$$

The variables μ_1, μ_2, σ_1^2, σ_2^2, σ_{12} denote the average, variance, and covariance of the edge-detected image of the original and noisy images.

In this step, we aim to demonstrate the increasing performance of the HFIS edge detection method as it advances through higher layers. To analyze the performance, we consider the test image Cottage, which has a size of 481×321 and is affected by Gaussian noise with zero mean and 0.01 variance (Figure 7(a)). We implement the highly effective HFIS edge detection method by utilizing three layers, which include the optimized gradient matrices E_3, E_5, E_7, and E_9. Our primary aim is to analyze and compare the performance and overall quality of the presented method across these three layers. To accomplish this, we conduct defuzzification on the outputs of the HFIS in every layer. Specifically, Figure 7(b) illustrates the output of the 2-th fuzzy inference system in the first layer (R_2^1), while Figure 7(c) presents the output of the 2-th fuzzy inference system in the second layer (R_2^2). Furthermore, Figure 7(c) shows the output of the last layer (R_1^3). Comparing the PSNR and SSIM values between the outputs R_3^1, R_2^2, and R_1^3 reveals a notable improvement in the performance of the HFIS edge detection method in higher layers. This improvement can be attributed to the innovative utilization of flexible rules and distinct membership functions within each layer. The reason for the increasing performance of the HFIS edge detection method as it advances through higher layers lies in the hierarchical structure of the system. This structure facilitates efficient propagation of relevant data (E_3, E_5, E_7, and E_9), allowing for maximized flow and utilization of information throughout the entire system. To assess the robustness of the HFIS edge detection method in differentiating edges from noise, we compare it with multiple edge detection algorithms discussed in previous studies [7, 10, 14, 19]. In order to comprehensively evaluate the algorithm's ability to handle noise, we utilize three test images: Woman's face, Butterfly, and Peppers, as illustrated in Figures 8(a), 9(a), and 10(a), respectively. The Woman's face image has dimensions of 321×481 pixels and is corrupted by Gaussian noise with a mean of zero and a variance of 0.3. The Butterfly image is a 256×256 pixel image affected by Gaussian noise with a mean of zero and a variance of 0.2. The Peppers image, measuring 512×512 pixels, is corrupted

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

FIGURE 7. (a) Test image Cottage of size 481 × 321 corrupted by Gaussian noise with zero mean, 0.01 variance (b) The output image of R_2^1 (PSNR=26.9720, SSIM=0.7379) (c) The output image of R_2^2 (PSNR=32.6635, SSIM=0.9024) (d) The output image of R_1^3 (PSNR=40.5497, SSIM=0.9682).

(a) (b) (c) (d) (e) (f)

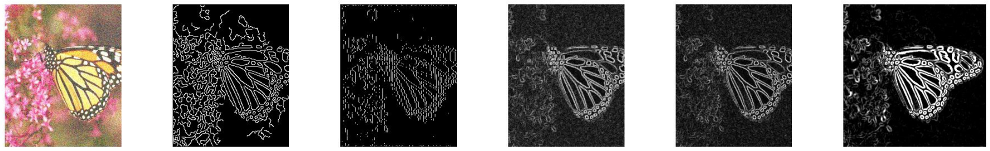
FIGURE 8. (a) Test image Woman's face of size 321 × 481 corrupted by Gaussian noise with zero mean, 0.3 variance (b) Canny algorithm (c) Fuzzy-based method [19] (d) Fractional-based method [14] (e) Fractional-based method [10] (f) The HFIS edge detection method.

TABLE 4. Comparison of PSNR and SSIM values of the methods in [7, 10, 14, 19] and the HFIS method.

	method [7]		method [19]		method [14]		method [10]		HFIS method	
Images	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Woman's face	9.4193	0.477	14.990	0.636	15.271	0.316	14.845	0.330	45.811	0.989
Butterfly	12.383	0.732	17.798	0.816	14.252	0.356	14.147	0.377	47.723	0.992
Peppers	8.3943	0.316	15.846	0.596	12.841	0.191	13.932	0.205	45.805	0.988

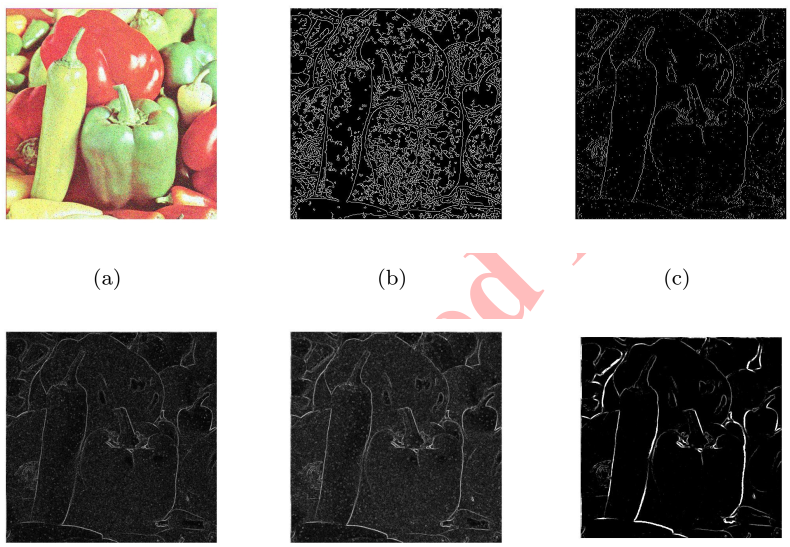
by mixed noise, consisting of Gaussian noise with a mean of zero and a variance of 0.2, as well as salt and pepper noise with a density of 0.01.

Figures 8(b), 9(b), and 10(b) present the edge-detected images obtained using the highly regarded and widely used Canny algorithm [7]. The edge-detected images obtained through the fuzzy-based technique (F^1-transform method) as described in [19] are displayed in Figures 8(c), 9(c), and 10(c). We have visually depicted the edge-detected images in Figures 8(d), 9(d), and 10(d) using a fractional-based technique described in [10]. Additionally, Figures 8(e), 9(e), and 10(f) present the results obtained by applying an alternative fractional-based technique as described in [14] to detect edges in the images. Our implementation of the HFIS edge detection method involves the integration of three layers. We utilize optimized gradient matrices, namely E_3, E_5, E_9, and E_{11}, to support the detection process. Lastly, the output from the HFIS edge detection method can be seen in Figures 8(f), 9(f), and **??**. The visual evidence presented in Figures 8, 9, and 10 unequivocally demonstrates that the effectiveness of the HFIS method in distinguishing edges



(a) (b) (c) (d) (e) (f)

FIGURE 9. (a) Test image Butterfly of size 256×256 corrupted by Gaussian noise with zero mean, 0.2 variance (b) Canny algorithm (c) Fuzzy-based method [19] (d) Fractional-based method [14] (e) Fractional-based method [10] (f) The HFIS edge detection method.



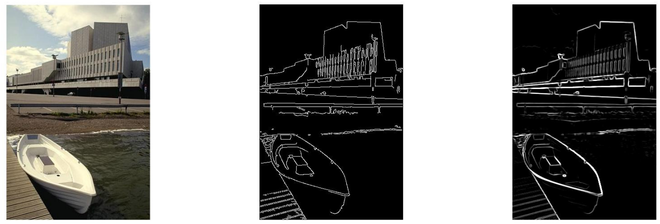
(a) (b) (c)

(d) (e) (f)

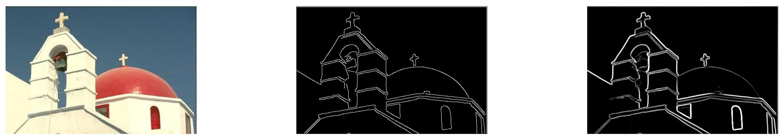
FIGURE 10. (a) Test image Peppers of size 512×512 corrupted by mixed noise containing Gaussian noise with zero mean, 0.2 variance and salt and pepper noise of density 0.01 (b) Canny algorithm (c) Fuzzy-based method [19] (d) Fractional-based method [14] (e) Fractional-based method [10] (f) The HFIS edge detection method.

from noise surpasses that of the other mentioned methods. In order to critically evaluate the performance of the HFIS edge detection method and methods mentioned in [7, 10, 14, 19], we have employed two key evaluation metrics: PSNR and SSIM. The results presented in Table 4 clearly demonstrate that the HFIS edge detection method outperforms other approaches such as Canny [7], fuzzy-based technique [19], and fractional-based techniques [10, 14] in terms of these evaluation metrics.

An especially notable benefit of the suggested approach is its extraordinary capability to precisely identify edge locations, all the while efficiently mitigating the impact of noise. This advantage is achieved through the implementation of a hierarchical structure with adaptable rules and diverse membership functions in our fuzzy inference system. In addition, our method utilizes the F^1-transform with different strides to construct optimized gradient matrices, increasing our methods tolerance to noise and enhancing its robustness.



(a) Original image. (b) Ground truth. (c) FOM=0.9016.



(d) Original image. (e) Ground truth. (f) FOM= 0.9158.

FIGURE 11. Results of the HFIS edge detection method for two images from BSDS500.

3.3. **FOM measure and experimental results.** Pratt's Figure of Merit (FOM) is a metric used to evaluate the performance of edge detection algorithms. It measures the accuracy and quality of the detected edges by comparing them with ground truth edges. The FOM is calculated using the following formula:

$$\text{FOM} = \frac{1}{\max(\mathcal{N}_{\text{ideal}}, \mathcal{N}_{\text{detected}})} \sum_{i=1}^{\mathcal{N}_{\text{detected}}} \frac{1}{1+\frac{d_i^2}{9}},$$

where $\mathcal{N}_{\text{ideal}}$ represents the number of edge pixels in the ground truth image, $\mathcal{N}_{\text{detected}}$ represents the number of edge pixels in the edge detected image, and d_i denotes the distance between an edge pixel in the edge detected image and its nearest edge pixel in the ground truth image.

The FOM ranges from 0 to 1, with 1 indicating a perfect match between the detected edges and ground truth edges. Higher values of FOM imply better performance in edge detection.

Initially, we evaluate the effectiveness of the HFIS method by employing the FOM criterion to analyze two test images depicted in Figures 11(a) and 11(d). The ground truths for these images are illustrated in Figures 11(b) and 11(e). The output from the HFIS edge detection method, as well as the corresponding FOM values, are demonstrated in Figures 11(c) and 11(f). The high values obtained for the FOM, along with the overall visual appearance of the images, indicate the success of the HFIS method. At this point, we present an examination of the FOM trends comparing the HFIS edge detection method to the fuzzy-based edge detection methods introduced in [19, 23]. We utilize a dataset consisting of 50 images sourced from the BSDS500 dataset along with their ground truth images created by skilled human professionals. The results of this comparison are presented in Figure 12. The findings from this analysis highlight the remarkable superiority and effectiveness of the HFIS edge detection method compared to existing techniques. In this stage, our focus is on assessing the robustness of our method in the presence of noise, which we measure using FOM criterion. To obtain a comprehensive analysis, we compare the average FOM values of the HFIS edge detection method with those of fuzzy-based techniques suggested in [16, 19]. To thoroughly assess the algorithm's capacity to handle noise, we employ four specific test images, namely Lena, Peppers, Car, and House. All of these images have dimensions of 256×256 pixels, as depicted in Figure 13. Moreover, these images have also been intentionally corrupted by the presence of Gaussian noise. The noise has a mean of zero and is characterized by distinct variances, namely 0.001 and 0.002. The results of this comparison are presented in Table 5. This analysis

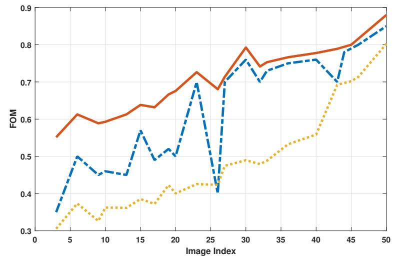
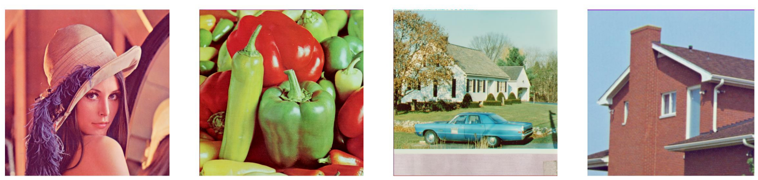


FIGURE 12. Performance comparison of HFIS edge detection method with fuzzy-based methods [19, 23] for BSDS500 images: HFIS method (Solid Line), method [23] (Dash Dot), method [19] (Dot).

TABLE 5. Comparison of the mean values of FOM of the methods in [16, 19] and the HFIS method.

	method [19]		method [16]		HFIS method	
Noise Levels	0.001	0.002	0.001	0.002	0.001	0.002
FOM	0.5089	0.4945	0.9382	0.9283	0.9803	0.9726

highlights the effectiveness and reliability of the HFIS edge detection method as a robust solution for edge detection in noisy images.



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

FIGURE 13. Test images Lena, Peppers, Car, and House of 256×256 Dimensions.

Finally, we present a performance comparison between the HFIS edge detection method and fractional-based techniques, as described in [10, 14, 26]. To evaluate the effectiveness of these methods, a test image of Lena is used, which has been corrupted with Gaussian noise at different noise levels. The results of this comparison are presented in Figure 14. This conclusion serves as quantitative evidence, supporting the superiority of the HFIS edge detection method in effectively mitigating the impact of noise on edge detection performance.

4. Conclusion

In this paper, we propose a novel technique for image edge detection using the F^1-transform and HFIS. By utilizing the F^1-transform with varying strides, we generate convolution masks of different sizes to smooth the image and calculate directional derivatives. The optimized gradient matrix is then evaluated by combining the four directional derivatives matrices. By adjusting the strides of fuzzy transforms, we able to control the level of smoothness in the image and obtain diverse optimized gradient matrices. We enhance the MISO fuzzy inference system by breaking it down into multiple two-input single-output fuzzy inference systems, each operating within the layers of the HFIS. This approach effectively reduces the computational cost by minimizing the number of rules and eliminating the need for fuzzifier and defuzzifier within the fuzzy inference systems located in the middle layers. We present the complete evaluation results of ACC, SEN, and SPC measures for both the CVG-UGR and BSDS500 datasets. The results

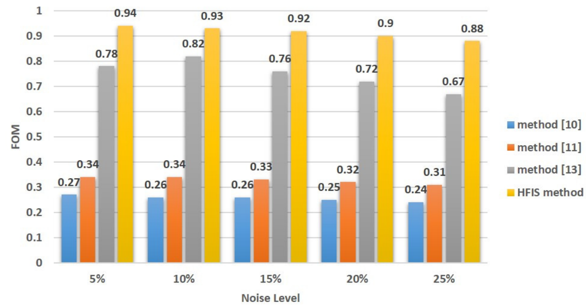


FIGURE 14. A comparative analysis of FOM trends in HFIS method and fractional-based techniques [10, 14, 26] for Lena's test image

show that our method achieves the high accuracy of 0.9018 for the BSDS500 dataset and 0.9226 for the CVG-UGR dataset. The ACC, SEN, and SPC values obtained from our technique for several images demonstrate that the HFIS edge detection method outperforms the fuzzy-based and fractional-based methods introduced in [10, 14, 19], as it consistently achieves better continuous edges. The performance evaluation of the HFIS edge detection method in terms of PSNR and SSIM criteria has been conducted on images corrupted by noise. The results have been compared with fuzzy-based and fractional-based techniques [10, 14, 16, 19, 26]. These criteria values demonstrate that the HFIS is significantly more robust against noise compared to other methods. Furthermore, we assess the performance of the HFIS edge detection method by comparing it to a fuzzy-based methods [16, 19, 23] and fractional-based methods [10, 14, 26], using the FOM criterion. The obtained results demonstrate that the HFIS edge detection method effectively and accurately estimates the precise location of image edges. The method's efficiency is crucial from two perspectives: utilizing the F^1-transform and employing HFIS. The utilization of F^1-transform with various strides in the method offers several benefits. These include the ability to control the smoothness of the image, obtaining continuous edges, minimizing the impact of noise, and acquiring different edge images. Moreover, the integration of HFIS in the method brings additional advantages, such as reducing computational costs and accurately detecting edge locations without being affected by noise interference.

Future Work: While the HFIS method performs well, several future directions exist. These include: 1) computational optimization via GPU parallelization for real-time video analysis; 2) automating parameter selection for specific image types using metaheuristic algorithms; 3) applying the method to domain-specific challenges like medical ultrasound or crack detection; and 4) exploring theoretical extensions such as higher-degree fuzzy transforms or combining it with deep learning to help the system learn and improve on its own (see [15]).

References

[1] R. Alikhani, M. Zeinali, F. Bahrami, S. Shahmorad, and I. Perfilieva, *Trigonometric F^m-transform and its approximative properties*, Soft Comput., *21* (2017), 3567–3577.

[2] R. Alikhani, I. Perfilieva, and M. Ganjeh-Alamdari, *Trigonometric F^{mn}-transform of multi-variable functions and its application to the partial differential equations and image processing*, Soft Computing, *26* (2022), 13301–13331.

[3] R. Alikhani and M. Mostafazadeh, *First order linear fuzzy differential equations with fuzzy variable coefficients*, Computational Methods for Differential Equations, *9* (2021), 1–21.

[4] R. Alikhani and M. Alamdari, *Fully fuzzy initial value problem of Caputo-Fabrizio fractional differential equations*, Computational Methods for Differential Equations, *11* (2023), 440–463.

[5] A. Bozorgmehr, M. K. Q. Jooq, M. H. Moaiyeri, K. Navi, and N. Bagherzadeh, *A novel digital fuzzy system for image edge detection based on wrap-gate carbon nanotube transistors*, Computers and Electrical Engineering, *87* (2020), 106811.

[6] Q. T. Bui, M. V. Pham, Q. H. Nguyen, L. X. Nguyen, and H. M. Pham, *Whale optimization algorithm and adaptive neuro-fuzzy inference system: a hybrid method for feature selection and land pattern classification*, International journal of remote sensing, *40* (2019), 5078–5093.

[7] J. Canny, *A computational approach to edge detection*, IEEE Transactions on pattern analysis and machine intelligence, *6* (1986), 679–698.

[8] Z. Cheraghali and M. Kazemi, *Ranking image segmentation methods using Data Envelopment Analysis*, Computing and distributed systems,*6* (2024), 53–58.

[9] F. Di Martino and S. Sessa, *Complete image fusion method based on fuzzy transforms*, Soft Comput., *23* (2019), 2113–2123.

[10] B. Ghanbari and A. Atangana, *Some new edge detecting techniques based on fractional derivatives with non-local and non-singular kernels*, Advances in Difference Equations, *1* (2020), 1–19.

[11] D. E. Ganesh, *Image segmentation using contemporary fuzzy logic*, International Journal of Computer Science and Engineering Technology, *8* (2017), 10(10.6084).

[12] C. I. Gonzalez, P. Melin, J. R. Castro, O. Castillo, and O. Mendoza, *Optimization of interval type-2 fuzzy systems for image edge detection*, Applied Soft Computing, *47* (2016), 631–643.

[13] S. Kumar and K. Alam, *PDE-based hyperbolic-parabolic model for image denoising with forward-backward diffusivity*, Computational Methods for Differential Equations, *9* (2021), 1100–1108.

[14] J. E. Lavín-Delgado, J. E. Solís-Pérez, J. F. Gómez-Aguilar, and R. F. Escobar-Jiménez, *A new fractional-order mask for image edge detection based on Caputo-Fabrizio fractional-order derivative without singular kernel*, Circuits, Systems, and Signal Processing, *39* (2020), 1419–1448.

[15] D. Li, X. Pan, Z. Fu, L. Chang, and G. Zhang, *Real-Time Accurate Deep Learning-Based Edge Detection for 3-D Pantograph Pose Status Inspection*, *71*(2022), 1–12.

[16] P. Melin, C. I. Gonzalez, J. R. Castro, O. Mendoza, and O. Castillo, *Edge-detection method for image processing based on generalized type-2 fuzzy logic*, IEEE Transactions on Fuzzy Systems, *22* (2014), 1515–1525.

[17] F. Orujov, R. Maskeliūnas, R. Damaševičius, and W. Wei, *Fuzzy based image edge detection algorithm for blood vessel detection in retinal images*, Applied Soft Computing, *94* (2020), 106452.

[18] I. Perfilieva, *Fuzzy transforms: theory and applications*, Fuzzy Sets Syst., *157* (2006), 993–1023.

[19] I. Perfilieva, P. Hodáková, and P. Hurtík, *Differentiation by the F-transform and application to edge detection*, Fuzzy Sets Syst., *288* (2016), 96–114.

[20] H. H. Razzaq, R. Ghazali, L. E. George, and M. Zulqarnain, *A novel framework for breast cancer scoring based on machine learning technique using immunohistochemistry images* , Computational Methods for Differential Equations, (2024), .

[21] M. S. Seidzadeh, H. R. Ghehsareh, S. K. Etesami, *An anomalous diffusion approach for speckle noise reduction in medical ultrasound images*, Computational Methods for Differential Equations, *10* (2022), 225–235.

[22] J. E. Solís-Pérez, J. F. Gómez-Aguilar, R. F Escobar-Jiménez, and J. Reyes-Reyes, *Blood vessel detection based on fractional Hessian matrix with non-singular Mittag-Leffler Gaussian kernel*, Biomedical Signal Processing and Control, *54* (2019), 101584.

[23] O. P. Verma and A. S. Parihar, *An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm*, IEEE Transactions on Fuzzy Systems, *25* (2016), 114–127.

[24] X. Wang, H. Gao, and Z. Jia, *Diffusion-based road defect detection model integrating edge information and efficient multi-scale convolution*, Applied Soft Computing, *178* (2025), 113332.

[25] J. Wu, Z. Yin, and Y. Xiong, *The fast multilevel fuzzy edge detection of blurry images*, IEEE signal processing letters, *14* (2007), 344–347.

[26] C. Xiu and X. Li, *Edge extraction based on memristor cell neural network with fractional order template*, IEEE Access, *7* (2019), 90750–90759.

[27] M. Zeinali, R. Alikhani, S. Shahmorad, F. Bahrami, and I. Perfilieva, *On the structural properties of F^m-transform with applications*, Fuzzy Sets and Systems, *342* (2018), 32–52.