The Edge Detection Technology of Gray-Scale Image Based on Dynamic Fuzzy Control

Qingpeng Ran
Department of Mathematics and Health Care, Yangtze University College of Arts and Sciences
Jingzhou 434020, China
E-mail: ran0409@126.com

Keywords: fuzzy set, fuzzy control, dynamic fuzzy, grayscale image, edge detection

Received: February 18, 2025

Edge detection is a key step in image processing, which is significant for image analysis and understanding. Traditional edge detection algorithms have limitations when dealing with grayscale images with blurred boundaries or low contrast features. To optimize the edge detection quality in grayscale images, relevant detection algorithms are designed based on fuzzy control methods in fuzzy mathematics. Then, a dynamic fuzzy control method is proposed to further optimize the algorithm. The study compares the detection effects of various commonly used image edge detection methods, including Canny, Roberts, Sobel, Fuzzy Control algorithm, and the research method before optimization. The results showed that the number of edge points detected by the research method increased by 32.78% compared with before optimization. The average detection time was decreased by 4.76%. The peak signal-to-noise ratio at maximum noise level was 4.43dB, and the structural similarity index was 0.0367, which was 0.35 dB and 0.0016 higher compared with the unoptimized method. The highest true positive rate and true negative rate were 47.79% and 98.87%, respectively, and the highest sensitivity was 7.75% higher compared with the method before optimization. The proportion of total hardware resources in the system was 23.71%, and the optimized detection method reduced the proportion of total hardware resources by 8.61% compared with before optimization. All relevant evaluation indexes were better than other similar algorithms. When the noise level increased from 0.5% to 1%, the fluctuation amplitude of peak signal-to-noise ratio and structural similarity index of the research method was lower than 0.12dB and 8×10^{-3}, respectively. The true negative rate value was stable from 93.26% to 98.87%. The research method can efficiently and accurately process edge images with low system resource utilization, improving the adaptability and robustness of the detection method.

Povzetek: Članek uvaja metodo zaznavanja robov sivinskih slik, ki temelji na dinamičnem mehkem krmiljenju. Z inovativnim prilagajanjem funkcij pripadnosti in pravil glede na lokalne značilnosti slike metoda izboljša detekcijo robov v pogojih šuma in nizkega kontrasta.

1 Introduction

Image edge is the boundary between objects in the image and the background or different objects, which is an important visual feature. Edge detection identifies areas with sharp changes in grayscale values by analyzing the grayscale values of the image. Through edge detection, key information in the image can be extracted, providing a foundation for subsequent image analysis and understanding [1]. Edge detection is largely used in image processing and computer vision, serving as a bridge between image processing technology and practical applications. It exerts a crucial role in automated monitoring, medical image analysis, and robot navigation [2]. Grayscale images refer to images with only one sampled color per pixel, which still has multi-level color depth between black and white. Although traditional edge detection algorithms perform well in certain application scenarios, they often struggle to obtain satisfactory results when processing grayscale images with fuzzy boundaries or low contrast features [3]. The fuzzy boundary is the area where the standard gradient is less than 5. The low contrast feature is where the normalized gray difference of the adjacent area is less than 0.15, or the local window signal-to-noise ratio is less than 10dB. In recent years, fuzzy mathematics has been introduced into image processing as an effective tool for dealing with uncertainty problems to improve the performance of existing algorithms. In fuzzy mathematics, elements are allowed to belong to a set with certain memberships, rather than strict binary logic. Fuzzy mathematics provides a new method for solving problems such as grayscale transitions, irregular shapes, and partial occlusion in images [4].

Existing methods still have significant limitations when dealing with complex scene of grayscale images. How to realize the high-precision and low-energy edge detection in the complex noise environment has become a key challenge restricting the development of intelligent image processing technology. A dynamic fuzzy control framework is proposed to address the above issues, which adjusts membership functions and fuzzy rules in real-time based on local image features, breaking the limitations of traditional methods that rely on prior parameters. The combination of adaptive filtering and fuzzy logic enhances edge continuity while suppressing noise, solving the balance between denoising and edge preservation in traditional methods. This study aims to construct a universal and robust edge detection solution by combining the performance advantages of subjective visual evaluation and objective metric quantification algorithms, and compare the edge localization accuracy of mainstream algorithms on the BSDS500 dataset. The measured algorithm consumes time and resource occupancy, and evaluates the adaptability of the method to the complex environment through cross-scenario testing.

2 Related works

There are various research methods currently available for image edge detection, which have different detection effects on different images.

Versaci M et al., in order to improve the accuracy of image edge detection, is proposed based on fuzzy entropy and fuzzy divergence to minimize the edge detection method, and USES the Chaira and Ray fuzzy edge detection technology. Compared with the traditional Canny algorithm, the proposed method had higher subjective visual scores at the same noise level, with Peak Signal-to-Noise Ratio (PSNR) increased by ≥0.3dB and True Positive Rate (TPR) increased by ≥5% [5]. Yongbin Y U et al. used genetic algorithm to optimize the image edge detection method. The performance was superior to the latest swarm intelligence models [6]. To improve the recognition and feature extraction of small targets in satellite images, Bhatti U A et al. used edge detection methods based on Clifford algebra and its subalgebras to process color images separately through quaternion Fourier transform. The results indicated that this method could provide more detailed information in edge detection [7]. Li S et al. proposed a two-level precise localization method based on edge and grayscale feature fusion to accurately detect workpiece deformation. The method employed normalized cross-correlation matching and truncated shape matching. The results showed that the new method was superior to other methods in detecting robustness and positioning accuracy. The average detection accuracy reached 81.7% and the positioning error was as low as 2.44 pixels [8]. Yang W et al. proposed a selective and adaptive training method to optimize the training performance of edge detection, and introduced a similarity preserving self-distillation mechanism. Compared with the method proposed in 2022, the F-score value of this method was increased by 0.5% and 7.6%, respectively [9].

Image edge detection based on fuzzy mathematics can handle uncertain and inaccurate information in images. Kumawat A et al. combined feature image registration and improved Canny fuzzy logic to optimize the edge detection accuracy in image processing. This method outperformed in seven image quality assessment parameters [10]. Raheja S et al. adopted fuzzy logic to improve the accuracy of edge detection, and used sharpening guided filters to control edge quality. This method could significantly improve the quality of detected edges [11]. To improve the accuracy of early detection of diabetes retinopathy, Setiadi D R et al. introduced Gabor filter, Gaussian filter, enhancement pre-processing method, and fuzzy c-means clustering segmentation algorithm for mathematical correction accuracy. The results showed that the introduced MMFCM improved the segmentation and analysis of retinal images [12]. Kumar A et al. used a Guided L0 smoothing filter based on fuzzy logic to improve edge detection in image segmentation. Compared with traditional methods, this method had an F-score of 0.848 on Berkley segmentation database and USC-SIPI [13]. Ranjan R et al. combined edge preserving guided filtering and fuzzy logic to address the impact of image quality on edge detection. This method could improve the accuracy of edge detection by precisely adjusting the learning parameters [14]. The performance indicators of the existing methods are shown in Table 1.

In conclusion, traditional fuzzy methods, such as [5], [10] and [11], rely on fixed membership functions and fuzzy rules, and need to be manually adjusted to adapt to different image features, resulting in poor versatility. Traditional algorithms, such as Canny, Roberts, etc., are sensitive to noise and experience a sharp drop in PSNR under high noise conditions. The Clifford algebra method in reference [7] and the intelligent group used in reference [6] have high hardware resource consumption and poor real-time performance. References [13] and [14] rely on expert experience to set the parameters, and the model generalization is limited. To address the issues with the above methods, an adaptive mechanism is established by automatically adjusting the membership function parameters and fuzzy rules based on real-time image features and expected values. The adaptive median filtering method dynamically adjusts the filtering window size while maintaining edge details. The effective implementation of the above methods is explored to solve the defects of traditional methods in dynamic adaptability, noise robustness, and resource efficiency.

Table 1: Comparison of the performance indicators of existing methods.

Study	Key Techniques	Supplemental Metrics	Limitations
Versaci M et al. (2021)	Fuzzy divergence + fuzzy entropy minimization	PSNR=4.1dB, SSIM=0.035, and TPR=38% (noise=0.5%)	Manual parameter tuning and poor dynamic adaptability.

Yongbin YU et al. (2021)	Genetic algorithm optimization	PSNR=3.8dB, SSIM=0.030, and processing time=0.45ms	High computational complexity (typical of swarm intelligence).
Bhatti U A et al. (2021)	Clifford algebra+quaternion Fourier transform	PSNR=4.0dB, and hardware resource utilization=35%	Inefficient for grayscale images (optimized for multi-spectral).
Huang W et al. (2023)	Standard operators (e.g., Canny) under turbulence	PSNR=4.2dB (low noise), and SSIM=0.025 (high noise)	Edge fragmentation in low-contrast images.
Yang W et al. (2023)	Self-distillation training mechanism	PSNR=4.1dB, SSIM=0.033, and F1-score=0.89 (requires 100k+ training samples)	Data dependency (poor small-sample performance).
Kumawat A et al. (2022)	Canny-fuzzy logic+feature registration	PSNR=4.15dB, SSIM=0.034, and hardware resource utilization=30%	Static fuzzy rules (no dynamic updates).
Raheja S et al. (2021)	Fuzzy logic+sharpening-guided filter	PSNR=4.2dB (low noise) → 3.5dB (noise=0.5%), SSIM=0.028 → 0.015	Noise sensitivity (no explicit anti-noise strategy).
Setiadi D R et al. (2024)	Graded fuzzy edge detection	PSNR=3.9dB, and SSIM=0.028 (steganography scenarios)	Limited generality (optimized for steganography).
Kumar A et al. (2021)	Guided L0 smoothing+fuzzy logic	PSNR=4.1dB, SSIM=0.035, and F1-score =0.85 (Real-time=1.2ms)	High latency (unoptimized computational efficiency).
Ranjan R et al. (2022)	Edge-preserving filter + fuzzy logic	PSNR=4.0dB, SSIM=0.032, and hardware resource utilization=28%	Parameter tuning relies on empirical expertise.

3 Methods and materials

3.1 Edge detection algorithm based on fuzzy control

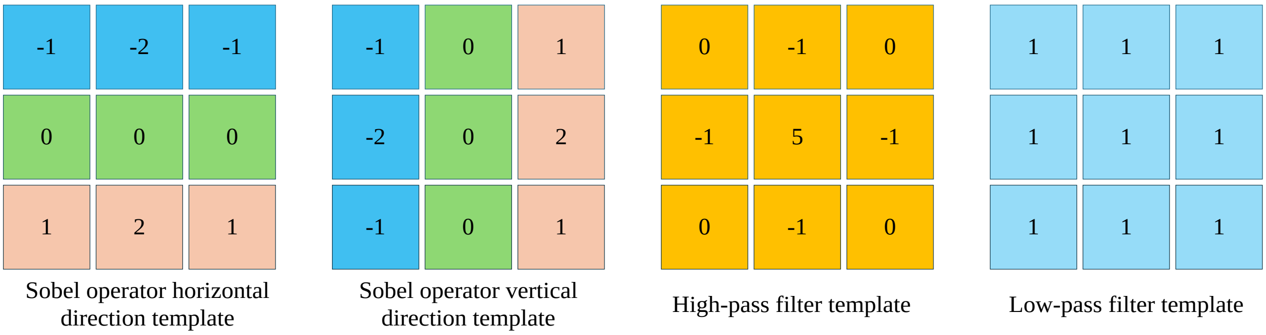
Fuzzy control is based on fuzzy mathematics, which makes decisions by imitating the experience and intuition of human experts. It is suitable for control systems with nonlinear, uncertain, or complex dynamic characteristics. In image edge detection, fuzzy control technology utilizes fuzzy logic to handle the uncertainty problem in the edge detection process. It processes and expresses gradient and fuzzy edges through fuzzy sets and fuzzy inference mechanisms. It determines the position of edge regions in a flexible and continuous manner, without relying on traditional hard threshold judgment [15]. Before performing edge detection, the noise is removed from the input image and a gradient image is generated as the input variable. The Sobel operator is used to obtain gradient images both horizontally and vertically, while convolving the images with both high-pass and low-pass filters. The convolution operation is shown in equation (1).

$$\begin{cases} I_x = \frac{1}{N} G_x * I \\ I_y = \frac{1}{N} G_y * I \\ I_h = G_h * I \\ I_l = G_l * I \end{cases} \tag{1}$$

In equation (1), G represents the gradient image calculated by the Sobel operator. I_x and I_y signify the gradient images in the horizontal and vertical directions. I_h and I_l represent high-pass and low-pass filtered images, respectively. N is the adaptive parameter, dynamically adjusted according to the window size. For example, for N 3×3 filter window, $N = 9$. The role of $1/N$ is to normalize the filter coefficient to ensure that the brightness of the filtered image remains stable and avoid numerical overflow or deviation caused by different window sizes. I is the original image. The Sobel operator and filter template used in the study are shown in Figure 1.

To construct a fuzzy set of each pixel and mathematically model ambiguous pixels in the image, the study describes the fuzzy set through membership functions to determine the degree to which pixels belong to the fuzzy set. The Gaussian membership function can change its shape and width by adjusting the mean and standard deviation parameters, making it suitable for different image processing needs, effectively smoothing the image, and reducing noise [16]. Equation (2) displays the Gaussian membership function.

$$f(x,\sigma,c) = e^{\frac{(x-c)^2}{2\sigma^2}} \qquad (2)$$



-1	-2	-1
0	0	0
1	2	1

Sobel operator horizontal direction template

-1	0	1
-2	0	2
-1	0	1

Sobel operator vertical direction template

0	-1	0
-1	5	-1
0	-1	0

High-pass filter template

1	1	1
1	1	1
1	1	1

Low-pass filter template

Figure 1: Sobel operator and filter template.

In equation (2), x is the normalized gray value of the input pixel. h is the center position of the membership function, which is the average gray value of the local window of the image. σ is the standard deviation. The width of the membership function is controlled. The initial value is set at 0.2, which is dynamically adjusted according to the global gradient variance of the image. e signifies the base of the natural logarithm. The membership function determines the degree to which each pixel belongs to the set, enabling the fuzzy set to be described and quantified. Fuzzy sets convert the edge information of pixels into fuzzy values through membership functions, thereby achieving edge localization and extraction [17]. The definition of a fuzzy set is shown in equation (3).

$$A = \left\{(\mu_A(x)), x \in X\right\} \qquad (3)$$

In equation (3), A is the fuzzy set, which is used to describe the edge information of pixels in the image. $\mu_A(x)$ signifies the membership function. X is the specified discourse domain, that is, the original gray value is mapped to the fuzzy discourse domain through x. $\mu_A(x)$ is the designated domain of discourse. After obtaining the fuzzy set, relevant fuzzy control rules need to be established. Fuzzy control rules define the relationship between input conditions and output results through fuzzy sets, helping the control system to reason based on fuzzy logic. The description of fuzzy reasoning is shown in equation (4).

$$if \;\; x \;\; is \;\; A, then \;\; y \;\; is \;\; B \qquad (4)$$

In equation (4), A represents the input condition. B represents the output result. The fuzzy rules defined in grayscale image edge detection are shown in Table 2.

The variables in fuzzy rules are fuzzy language variables. The state description of fuzzy language variables is shown in equation (5).

$$\begin{aligned} &T(x) = \{-\text{high}, \text{-medium}, \text{-low}, \\ &\text{none}, +\text{none}, +\text{low}, +\text{medium}, +\text{high}\} \end{aligned} \qquad (5)$$

In equation (5), $T(x)$ is the set of language variables. $-\text{high}$, -medium, -low, none, $+\text{none}$, $+\text{low}$, $+\text{medium}$, and $+\text{high}$ represent different states of input and output variables, used to describe the size of the language variables, indicating the degree to which they belong to a fuzzy set. To strengthen or weaken the tone of fuzzy language, the study uses tone operators to change the strength of fuzzy statements, so that fuzzy logic can be closer to human language and thinking patterns [18]. The membership function processed by tone operators is shown in equation (6).

$$\mu_{H_\lambda(A)}(u) = (\mu_A(u))^\lambda \qquad (6)$$

In equation (6), A is a fuzzy set. u is the domain of fuzzy sets. $\mu_A(u)$ signifies the membership function of the fuzzy set A. H_λ signifies the tone operator. $H_\lambda(A)$ is another fuzzy set on the domain after $H_\lambda(A)$ processing. λ represents the change in tone. Fuzzy reasoning is the process of deriving results based on the conditions in fuzzy rules by mimicking humans when faced with fuzzy information. The Mamdani method is used for fuzzy reasoning in the study [19]. This method is shown in equation (7).

$$\mu_{R_M}(x,y) = \mu_A(x) \wedge \mu_B(y) \qquad (7)$$

In equation (7), R_M represents the relationship between fuzzy sets. $\mu_{R_M}(x,y)$ represents the membership function of R_M. $\mu_A(x)$ and $\mu_B(y)$ signify the membership functions of fuzzy sets A and B on elements x and y, respectively. After the image is blurred in the early stage, it needs to be deblurred, that is, to find the accurate output value within the output range. The study uses the centroid method for deblurring, which determines the output of fuzzy inference by calculating the centroid position of the image area in the fuzzy set [20]. The calculation process of the centroid method is shown in equation (8).

$$u = \frac{\int u \cdot \mu_N(x)dx}{\int \mu_N(x)dx} \qquad (8)$$

In equation (8), u is the numerical output obtained from the fuzzy set N. $\mu_N(x)$ is the membership function value of N at pixel x. When the output transformation stage of a fuzzy set is discretized, the centroid method is shown in equation (9).

$$u=\frac{\sum_{i=1}^{m}u_i\cdot\mu_N\left(x_i\right)}{\sum_{i=1}^{m}\mu_N\left(x_i\right)} \tag{9}$$

In equation (9), x_i signifies the discrete point value in the theoretic domain. m represents the quantization series after discretizing the continuous gray values, which are typically quantified by 8-bit in image processing, i. e., $m=256$. The membership function value of the selected N is dynamically determined by the Gaussian type membership function, which can be taken in the closed interval [0, 1]. Its parameters c and σ are adjusted according to the image gradient distribution and mathematical expectation. A larger gradient indicates that the function value is closer to 1, and the probability of pixels is higher. When the gradient is below the dynamic threshold, the function value approaches 0, indicating noise or non-edge areas. The fuzzy control process in grayscale image edge detection is shown in Figure 2.

Table 2: Fuzzy rule set.

Serial number	**Input variable**				**Output variable**
	I_x	I_y	I_h	I_l	**Edge**
1	Medium	None	Low	None	High
2	None	Medium	Low	None	High
3	Low	None	Low	None	Low
4	Low	Low	None	None	Low
5	Medium	Medium	None	None	High
6	High	High	None	None	High
7	None	High	None	Medium	High
8	None	Medium	None	Low	Low
9	High	None	None	Medium	High
10	High	None	None	Low	Medium

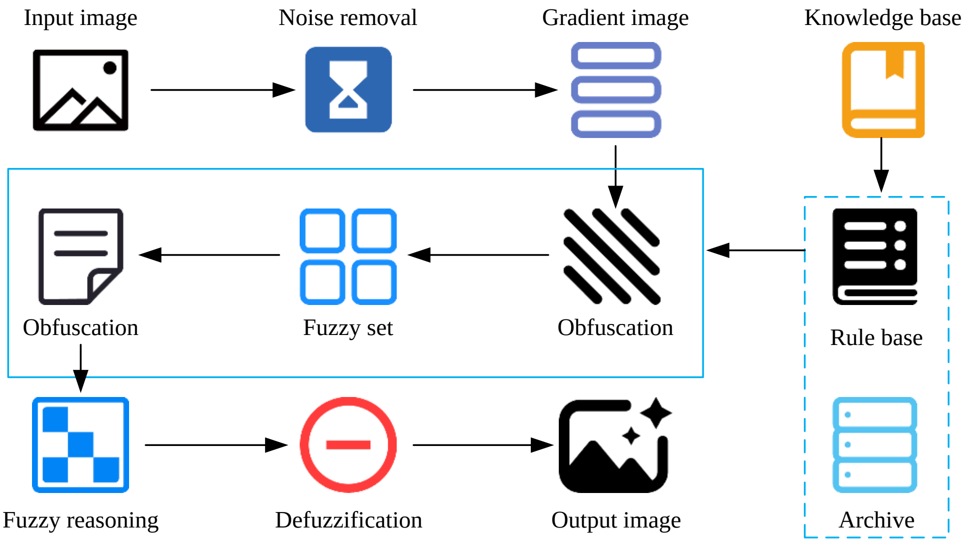


Figure 2: Fuzzy control flow in edge detection of grayscale image.

In Figure 2, in the fuzzy control process, the preprocessed input image needs to be blurred first. Relevant features in the images include horizontal gradient, vertical gradient, local gray scale variation, noise intensity, gradient direction variance, and normalized gray values, which are converted into fuzzy sets through membership functions. The membership degree of each pixel in the image is determined based on the rule library, that is, whether the pixel belongs to the edge. The results of fuzzy reasoning are transformed into clear decisions, that is, deblurring. Finally, the output image is obtained.

3.2 Edge detection algorithm based on dynamic fuzzy control

The parameter settings of the membership function and fuzzy rules in fuzzy control methods cannot be adaptively adjusted with changes in the image. Therefore, to ensure the quality of the output image, the parameters of the membership function and fuzzy rules are reset, which makes the detection process cumbersome and reduces the accuracy of the algorithm. To adaptively adjust the relevant parameters and rules based on the real-time image characteristics and environmental changes, and optimize the adaptability and accuracy, a grayscale image edge detection based on dynamic fuzzy control is built. The dynamic fuzzy control process is shown in Figure 3.

As shown in Figure 3, in dynamic fuzzy control, noise is first removed from the input image information. After filtering, gradient images and fuzzy sets are generated. According to fuzzy rules and fuzzy theory, edge control is applied to fuzzy sets. The parameters of the membership function and fuzzy rules are adaptively adjusted based on the edge information of the current image to change the membership function of subsequent input images. Finally, the fuzzy set is subjected to deblurring processing.

In dynamic fuzzy control, an adaptive median filter is taken to remove noise, which can dynamically adjust the window size according to the noise level, edge intensity, and pixel distribution. In image processing, the noise intensity may vary in different regions [21]. Adaptive median filtering dynamically adjusts the size of the filter window according to the noise level in the local region. If the local noise is higher, the filter may expand the window to remove the noise more effectively. If the noise is lower, the window can be narrowed to reduce the interference to the details of the image. Adaptive median filtering will detect the edge strength in local regions. If the local area contains strong edges, the filter will minimize the window size to avoid blurred edges. If the local area has no obvious edges, the window can be appropriately increased to remove noise. Pixel distribution includes statistical properties such as the mean and variance of pixel values. If the pixel value distribution of the local area is relatively uniform, it means that the area may be the smooth area in the image, and the window can be appropriately increased to remove noise. If the pixel value distribution is not uniform, it may indicate that the area contains complex texture or edges. Therefore, expanding the window should be avoided. The workflow of the adaptive median filter is shown in Figure 4.

From Figure 4, when using a filter to remove noise, first, the size of the filtering window is determined. Then, the minimum grayscale value $Z_{\min}$, the maximum grayscale value $Z_{\max}$, and the grayscale median value Z_{med} under the current window are calculated. The sizes of the three values are compared to determine whether Z_{med} is within $Z_{\max}$ and $Z_{\min}$. If the judgment is true, the original pixel value Z_{xy} is further determined to be within the same range. If it is within the same range, the pixel is considered not to be a noise point. Otherwise, the noise point should be replaced with Z_{med} . If the judgment is not valid, the window size is added. When the window size is larger than the maximum size $F_{\max}$, Z_{med} is used to replace the current pixel point. To address the issue of manually setting parameters in static fuzzy methods during image changes, this study takes dynamic fuzzy control to collect edge information from images. Firstly, the number of occurrences of image grayscale values within the minimum and maximum grayscale value ranges is calculated, and the first expected value of the image grayscale value is obtained [22]. The first expected value is shown in equation (10).

$$e_1 = \frac{\sum_{v_{\min}}^{v_{\max}} v \times I(v)}{\sum_{v_{\min}}^{v_{\max}} I(v)} \tag{10}$$

In equation (10), e_1 is the first expected value, which represents the minimum and maximum grayscale values. $I(v)$ signifies the number of times the grayscale value of the image appears within $v_{\min}$ and $v_{\max}$. Afterwards, the second, third, and fourth expected values are calculated separately from the first expected value. Due to the different linguistic variables in the input images, the parameters in the corresponding Gaussian membership function also change accordingly. When the fuzzy language is ' *low*" , the pixel value is $c = 0$, which is put into the Gaussian membership function. The expression about σ can be obtained, as shown in equation (11).

$$\begin{cases} c = 0 \\ \sigma = \dfrac{Threshold_{(low)}}{2\sqrt{3\ln^{10}}} \end{cases} \tag{11}$$

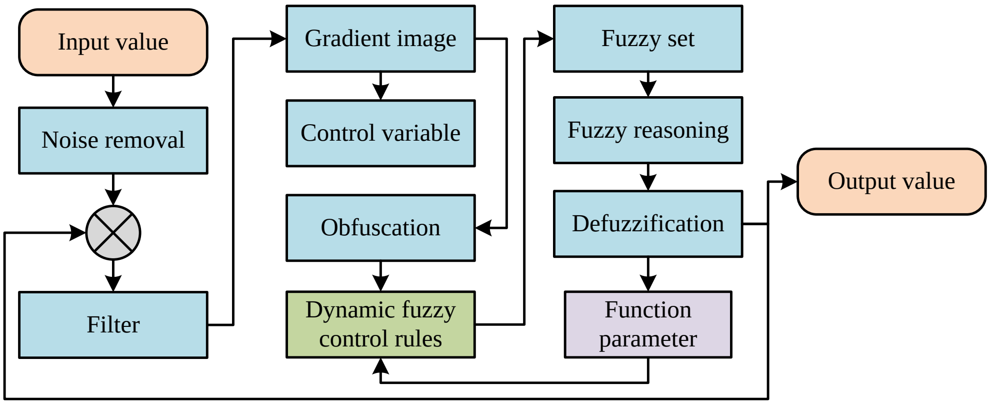


Figure 3: Flow chart of dynamic fuzzy control.

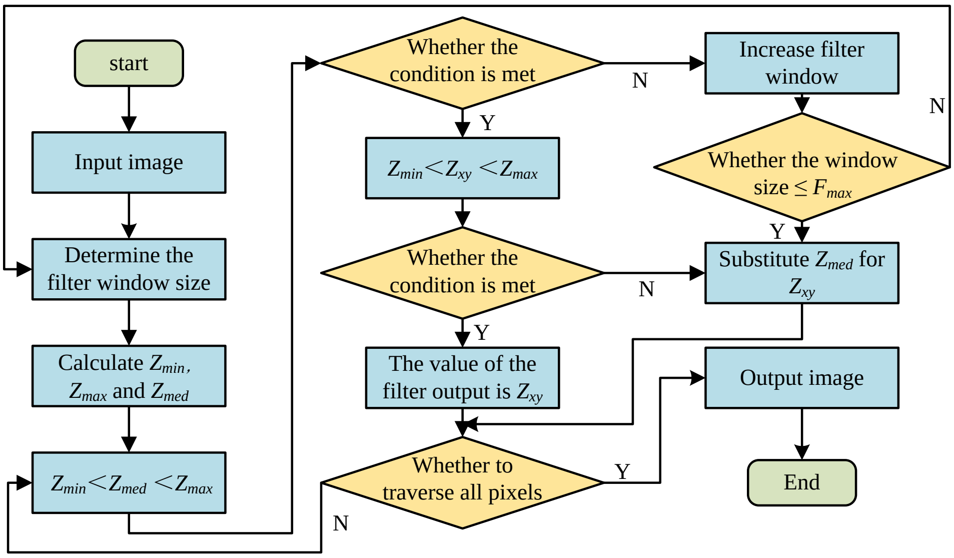

start
Input image
Determine the filter window size
Calculate Z_{min}, Z_{max} and Z_{med}
$Z_{min} < Z_{med} < Z_{max}$
Whether the condition is met
N
Y
Increase filter window
N
Whether the window size ≤ F_{max}
Y
$Z_{min} < Z_{xy} < Z_{max}$
Whether the condition is met
N
Y
Substitute Z_{med} for Z_{xy}
The value of the filter output is Z_{xy}
Whether to traverse all pixels
Y
N
Output image
End

Figure 4: Flow chart of adaptive median filter.

In equation (11), c represents the pixel value. $Threshold_{(low)}$ represents the threshold when fuzzy language is “low” . Grayscale images typically store each pixel value in 8-bit (1 byte) in a range from 0 (pure black) to 255 (pure white). The setting of the input value range [0, 255] stems from a normalized representation of the 8-bit grayscale image as an input range of the Gaussian membership function. When the fuzzy language is “medium” and “high” , the pixel value is expressed as half of the maximum pixel value (127.5) and the maximum pixel value (225), respectively. When the fuzzy language is “low” , the pixel value is $c = 0$. It is used in the Gaussian membership function to obtain the expression about σ . When the fuzzy language is “medium” and “high” , the pixel value c is changed to 127.5 and 225, respectively, and then calculated according to the same equation to obtain the parameters under various fuzzy language variables [23]. The threshold $Threshold$ in the above equation is shown in equation (12).

$$\begin{cases} Threshold_{(low)} = \begin{cases} e_1 & if \quad Mean_{Local} < e_1 \\ Mean_{Local} & otherwise \end{cases} \\ Threshold_{(medium)} = Mean_{Local} \\ Threshold_{(high)} = \begin{cases} e_4 & if \quad Mean_{Local} > e_4 \\ Mean_{Local} & otherwise \end{cases} \end{cases} \tag{12}$$

In equation (12), e_4 is the fourth expected value. $Threshold_{(medium)}$ and $Threshold_{(high)}$ are the thresholds for “medium” and “high” in the fuzzy language, respectively. $Mean_{Local}$ is the average grayscale value, as shown in equation (13).

$$Mean_{Local} = \frac{\sum_{x=-1}^{x=1} \sum_{y=-1}^{y=1} (x+i, y+j)}{9} \tag{13}$$

In equation (13), x and y signify the center point coordinates of the filter window. i and j are the offsets of the coordinates. To deal with the constantly changing fuzzy logic in dynamic fuzzy control, the Takagi-Sugeno fuzzy reasoning method [24] is adopted. The reasoning process of this method is shown in equation (14).

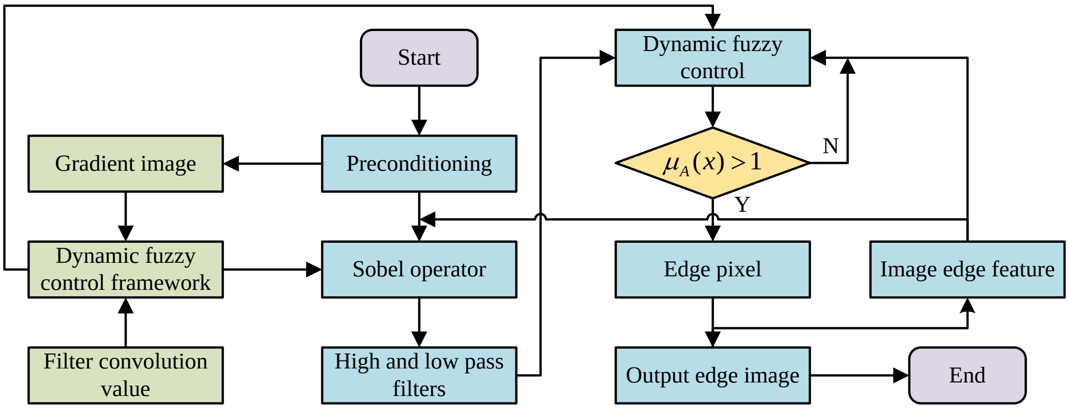


Figure 5: Flow chart of grayscale image edge detection algorithm based on dynamic fuzzy.

$$R^l: If \quad x_1 \text{ is } F_1^l \text{ and} \cdots x_n \text{ is } F_n^l \text{ Then } x(t)=A_l x(t)+B_l u(t) \qquad (14)$$

In equation (14), R^l is the approximate inference rule l in the Takagi-Sugeno method. x_1 and x_n are input variables, including the image gradient amplitude and mathematical expected value. F_1^l and F_n^l are fuzzy sets corresponding to the input variables in rule l. $x(t)$ is the state variable, representing the current state of the dynamic fuzzy control system, including image edge features, membership function parameters, and thresholds. A_l is the coefficient matrix multiplied by the input variable. B_l is the coefficient matrix multiplied by the control input variable. $u(t)$ is the control signal used to guide the edge detection process and adjust the detection strategy. After normalizing the inference fuzzy set, product inference and single point fuzzy processing are used in dynamic fuzzy control. The final state is calculated through the weighted average of the fuzzy set [25], as displayed in equation (15).

$$x(t)=A(\mu(t))x(t)+B(\mu(t))u(t) \quad (15)$$

In this algorithm, firstly, the grayscale image needs to be converted into a two-dimensional arithmetic matrix form, and an adaptive median filter is used to remove noise from the image. The Sobel operator and filter are used to obtain the corresponding gradient image. The relevant parameters of the Gaussian membership function are obtained from the gradient amplitude of the gradient image, the local window gradient mean, and the global gradient variance. The centroid method is used for deblurring fuzzy sets to obtain accurate edge image information. The Sobel operator and the window size in the filter are dynamically adjusted according to the edge information of the gray image. The relevant parameters of the Gaussian membership function are adjusted according to the expected value of the maximum gray value of the edge image. When encountering similar images in future detection, relevant parameters can be automatically set. The entire detection algorithm process is shown in Figure 5.

The fuzzy rule base is designed based on expert experience. The input variables are horizontal gradient G_x, vertical gradient G_y, local gray change, and noise intensity, and the output variable is edge probability. The value range corresponding to the fuzzy language variable in the rule is defined, as shown in equation (16).

$$\begin{cases} \text{Gradient variable:} \\ \begin{cases} Low:[0,0.3\cdot G_{\max}] \\ Medium:[0.3\cdot G_{max},0.7\cdot G_{max}] \\ High:[0.7\cdot G_{\max},G_{\max}][0.7\cdot G_{max},G_{max}] \end{cases} \\ \text{Gray change:} \\ \begin{cases} Low:[0,\ 0.2] \\ Medium:[0.2,\ 0.6] \\ High:[0.6,\ 1.0] \end{cases} \end{cases} \qquad (16)$$

In equation (16), $G_{\max}$ is the maximum gradient value of the image, normalized to [0,1]. The initial value of the standard deviation σ of the Gaussian function is 0.2, which is dynamically updated according to the global gradient variance V_g of the image. The updating process is shown in equation (17).

$$\sigma = 0.2 + 0.1\cdot\tanh(V_g/0.05) \qquad (17)$$

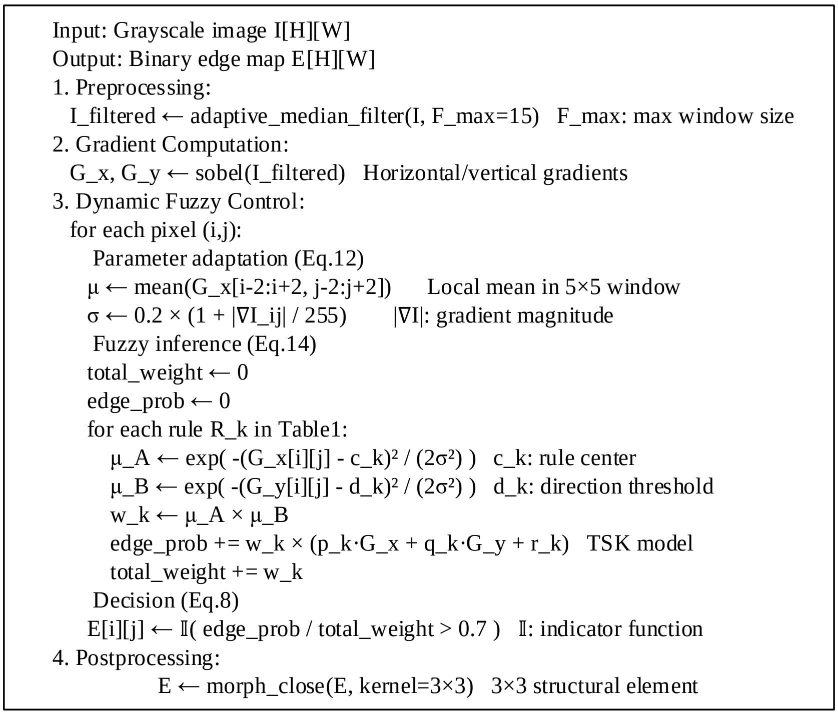
Real-time thresholds are generated by Takagi-Sugeno fuzzy inference, whose weight coefficients are automatically calculated based on the fourth-order expected values. In the real-time processing, the membership function parameters are updated once per frame, and the fuzzy rule weights are updated once every 5 frames. The update delay is controlled within 0.5ms to ensure that the algorithm meets the real-time performance requirements. In the image gray transition problem, equation (2) is used to quantify the fuzziness of the gray gradient region. For the low-contrast region with gray difference $\Delta I < 0.15$ between adjacent pixels, the fuzzy set is dynamically divided by adjusting the standard deviation of the membership function, and the continuous gray value is mapped into three types of fuzzy states, namely weak edge, medium edge, and strong edge, to avoid the edge fracture caused by traditional hard threshold segmentation. In the irregular shape processing problem, fuzzy rule weights are generated dynamically through Takagi-Sugeno fuzzy inference (Equation 14), gradient direction distribution of local window (equation 1), and gradient direction variance V_θ within the window. When the gradient direction variance of the irregular shape is $V_\theta > 0.1$, the weight of the gradient size rule is reduced, and the decision priority of the local gray change (equation 3) is raised, so as to avoid the edge missing detection caused by rule rigidity.

The image occlusion area processing based on fuzzy mathematics is as follows:

For the occluded region in the image, the fuzzy set $\mu_{\text{occlusion}}(I)$ of occluded object and background is modeled by the dual-modal membership function, as shown in equation (18).

$$\mu_{\text{occlusion}}(I) = \max\left(\mu_{\text{low}}(I), \mu_{\text{high}}(I)\right) \quad (18)$$

In equation (18), $\mu_{\text{low}}(I)$ and $\mu_{\text{high}}(I)$ are membership functions for low grayscale occlusion and high grayscale background, which retain potential edge information through maximization operation. In the real-time processing, the membership function parameters are updated every frame, and the fuzzy rule weights are updated every 5 frames. The update delay is controlled within 0.5ms to ensure that the algorithm meets the real-time performance requirements. The pseudo-code of the dynamic fuzzy control algorithm is shown in Figure 6.



```
Input: Grayscale image I[H][W]
Output: Binary edge map E[H][W]
1. Preprocessing:
  I_filtered ← adaptive_median_filter(I, F_max=15)   F_max: max window size
2. Gradient Computation:
  G_x, G_y ← sobel(I_filtered)   Horizontal/vertical gradients
3. Dynamic Fuzzy Control:
  for each pixel (i,j):
     Parameter adaptation (Eq.12)
    μ ← mean(G_x[i-2:i+2, j-2:j+2])      Local mean in 5×5 window
    σ ← 0.2 × (1 + |∇I_ij| / 255)        |∇I|: gradient magnitude
     Fuzzy inference (Eq.14)
    total_weight ← 0
    edge_prob ← 0
    for each rule R_k in Table1:
      μ_A ← exp( -(G_x[i][j] - c_k)² / (2σ²) )   c_k: rule center
      μ_B ← exp( -(G_y[i][j] - d_k)² / (2σ²) )   d_k: direction threshold
      w_k ← μ_A × μ_B
      edge_prob += w_k × (p_k·G_x + q_k·G_y + r_k)   TSK model
      total_weight += w_k
     Decision (Eq.8)
    E[i][j] ← 𝕀( edge_prob / total_weight > 0.7 )   𝕀: indicator function
4. Postprocessing:
           E ← morph_close(E, kernel=3×3)   3×3 structural element
```

Figure 6: The pseudo-code of the dynamic fuzzy control algorithm.

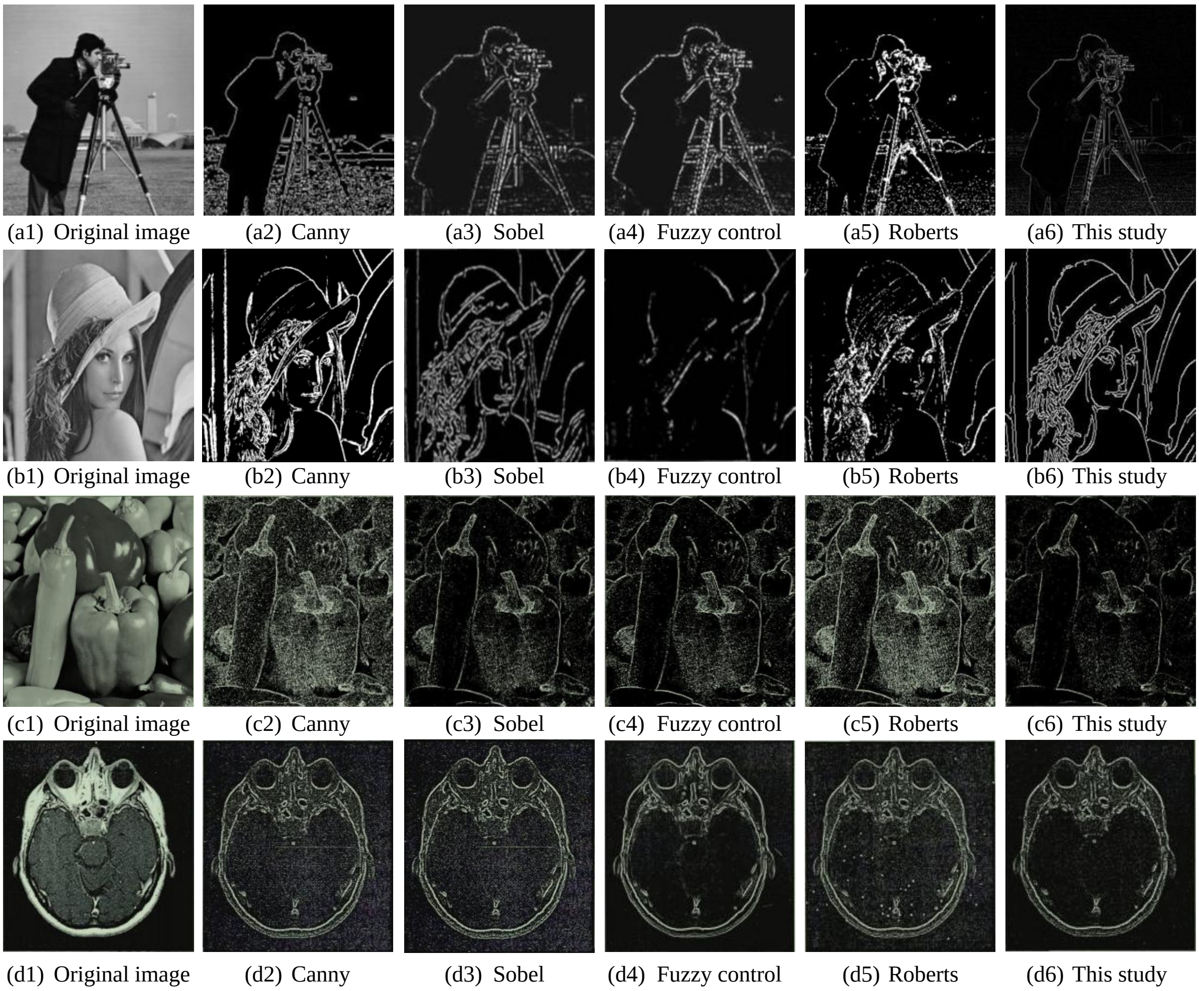
4 Results

4.1 Comparison of subjective analysis and detection efficiency

To verify the proposed algorithm, comparative experiments are conducted with the improved fuzzy control algorithm and commonly used edge detection algorithms to assess the performance. The fuzzy control algorithm and Canny algorithm are from [8] and [11], the edge detection method based on Dense Residual Network (DRN) is from reference [26], and the Sobel algorithm and Robert algorithm are from reference [27]. All algorithms are tested under the same conditions. The central processing unit of the desktop computer used in the experiment is Intel Core i5-10700K, the operating system is Windows 10 Professional 64-bit, and the software environment is MATLAB R2023. The experimental dataset is BSDS500, which contains 500 grayscale images covering various scenes, including complex textures, low contrast, and noise interference. The training, validation, and test sets have 200,100, and 200 images, respectively. The input images are also normalized to the [0, 1]. The detailed experimental environment parameters are shown in Table 3. The maximum window size of the adaptive median filter is 15, the initial value of the standard deviation of the Gaussian membership function is 0.2, the upper limit of the parameter update delay is 0.5ms, the fuzzy rule update frequency is 5 frames per time, and the edge determination threshold is 0.7. The fuzzy control algorithm proposed in the study and the improved dynamic fuzzy control algorithm are used to perform edge detection on the two grayscale images listed. Three commonly used edge detection algorithms are compared to subjectively analyze the generated edge images. In the BSDS500 dataset, the edge detection images obtained by various methods are shown in Figure 7.

Table 3: Parameter information of model training test platform.

Hardware facility		Software facility	
Device CPU	Intel Core i5-10700K	Operating system	Windows 10 Professional 64-bit
Device graphics card	NVIDIA GeForce RTX 3060 12GB	Programming language	Python 3.6
Storage device	SSD 1TB NVMe	Programming environment	MATLAB R2023a
Network interface	Gigabit Ethernet	Simulation and testing tools	Jupyter Notebook
Internal memory	32GB DDR4	Image processing toolkit	MATLAB Image Processing Toolbox

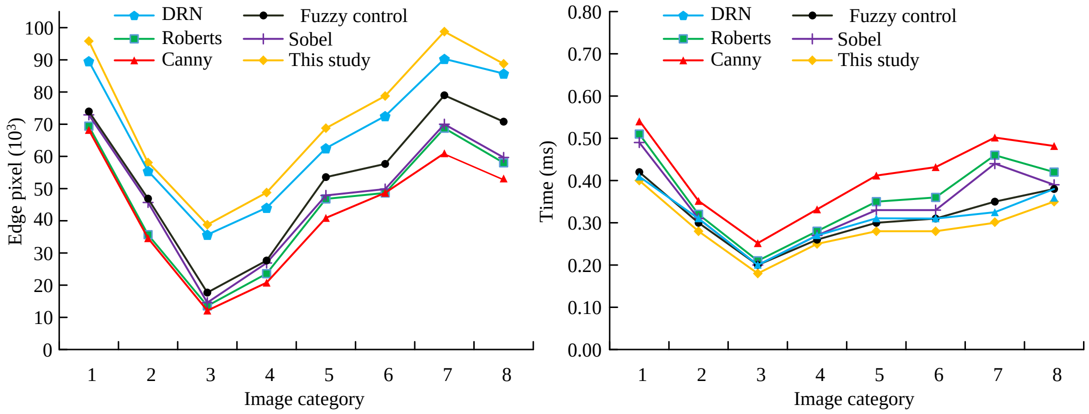

(a1) Original image (a2) Canny (a3) Sobel (a4) Fuzzy control (a5) Roberts (a6) This study

(b1) Original image (b2) Canny (b3) Sobel (b4) Fuzzy control (b5) Roberts (b6) This study

(c1) Original image (c2) Canny (c3) Sobel (c4) Fuzzy control (c5) Roberts (c6) This study

(d1) Original image (d2) Canny (d3) Sobel (d4) Fuzzy control (d5) Roberts (d6) This study

Figure 7: Different methods for detecting edge images based on the original image.



(a)The number of edge pixels extracted by each method

(b) The extraction time of image edge pixels of each method

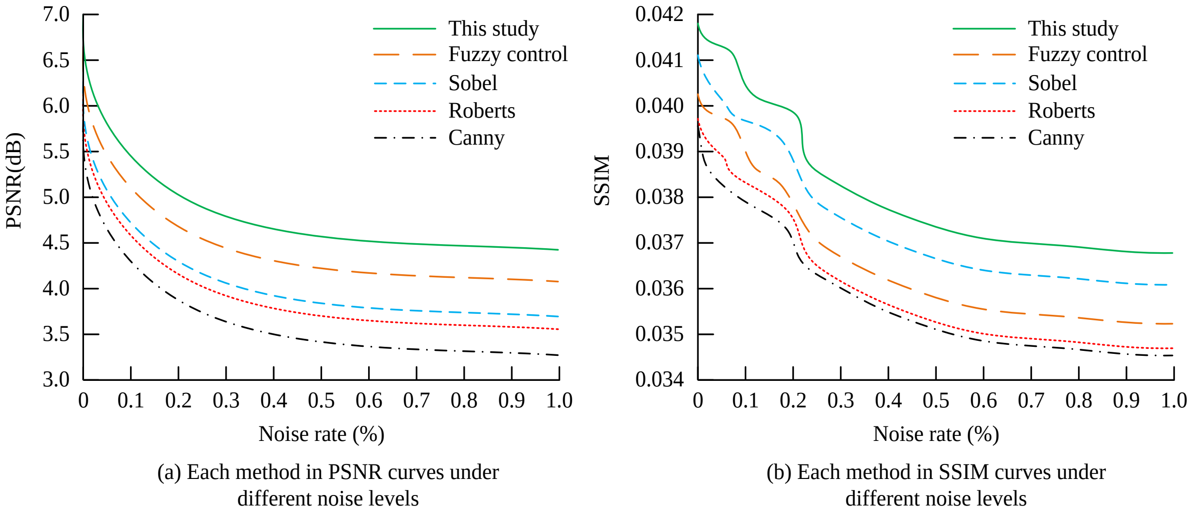
Figure 8: Each method for detecting the number of edge points and detection time.

In Figure 7, from images c2, b5 and a4, the detected images obtained by Canny, Roberts and Sobel algorithms appeared, and the edge image quality extracted by these three algorithms was significantly lower than that of images d2, d5 and d4. From the image b4, the edge detection method based on static blur control exhibited edge information loss without more pseudo edges and isolated points. Therefore, compared with other methods, the optimized algorithm can effectively detect the real edges of the image. From images a6 and b6, the lines were extracted with clear and smooth edges, which were close to the original image, and the noise was reduced to a large extent. To further compare the detection performance of various methods, this study compares the above methods and the latest edge detection method based on DRN. The number of edge points extracted during the edge detection of 8 types of grayscale image and the time required for the detection process. The statistical results are shown in Figure 8.

Figure 8 (a) shows the number of edge points extracted by each method, and Figure 8 (b) shows the time required for the detection process. In Figure 8 (a), the difference in the number of edge points extracted by Canny, Roberts, and Sobel algorithms in various types of images was relatively small. The optimized method can extract more edge points from all categories of images. Compared with before optimization, the number of detected edge points was increased by 32.78%. Compared with DRN, the research method extracted more number of edge points in different categories of images, and the average number of edge points increased by 7.52×10^3. As shown in Figure 8 (b), the detection time of the research method was the lowest, with an average time of 0.30ms, which was 4.76% shorter than before optimization, 27.04% shorter than Canny algorithm, 17.53% shorter than Roberts algorithm, and 13.04% shorter than Sobel algorithm. This research method required a shorter time to extract images of different categories, reducing the average detection time by 0.18ms. From this, the method proposed in the study has a relatively accurate detection effect and high computational efficiency.

4.2 Comparison of edge detection capability and resource utilization rate

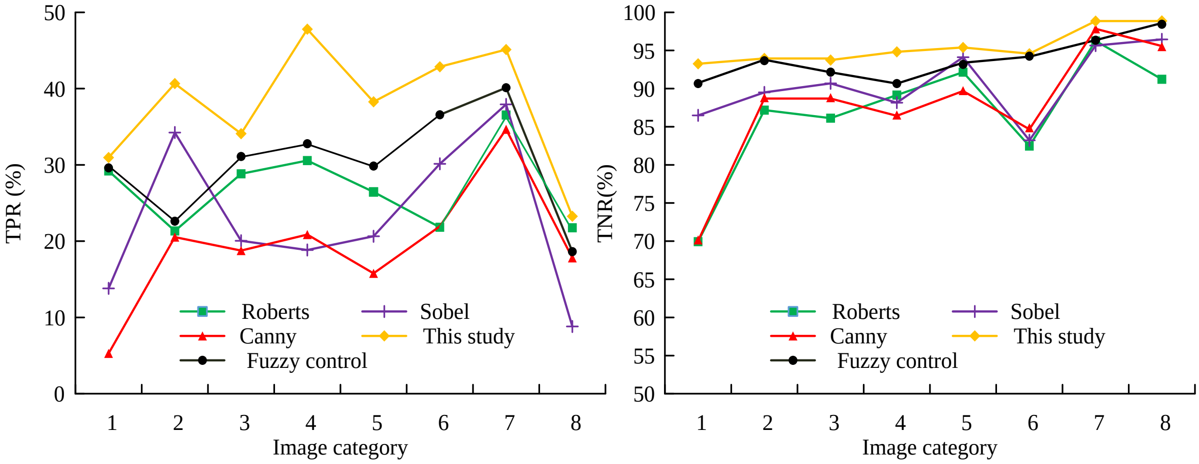
To further objectively analyze the performance, the Structural Similarity Index (SSIM) and PSNR between the original grayscale image and the edge image are compared. PSNR is an objective metric for measuring the difference between two images. A high value indicates that the error between the reconstructed image and the original image is small, that is, indicating better image quality. SSIM is a quality evaluation metric applied to assess the similarity between two images, focusing on the structural information and better reflecting the perception of the human visual system. The PSNR and SSIM changes of each method are displayed in Figure 9. Figure 9 (a) shows the PSNR change curve of each method under different noise levels, and Figure 9 (b) shows the SSIM change curve of each method under different noise levels.



(a) Each method in PSNR curves under different noise levels

(b) Each method in SSIM curves under different noise levels

Figure 9: PSNR and SSIM change curve with noise level.



(a) TPR of image edge detection by each algorithm

(b) TNR of image edge detection by each algorithm

Figure 10: TPR and TNR for image edge detection by each algorithm.

In Figure 9 (a), the PSNR of various methods rapidly decreased before the noise level reached 0.3%, and then showed a gradually slow decreasing trend. The curve of the Canny algorithm was located at the bottom, demonstrating that there was an obvious difference and severe distortion between the edge image and the original image. With the increase of noise level, the PSNR of the research method was always higher than other methods, and the PSNR at the maximum noise level was 4.43dB, indicating that this method had lower distortion compared to other methods and was more similar to the original image. In Figure 9 (b), the proposed method had a higher SSIM compared to other methods. However, in high noise environment (Noise rate=1%), PSNR and SSIM decreased to 4.43 dB and 0.0367, respectively. This is because the maximum window size of the adaptive median filter cannot cover the cluster distribution area of high-intensity pulse noise, resulting in residual noise. Meanwhile, the membership function of dynamic fuzzy rule is frequently adjusted by noise interference, causing the misjudgment rate of low gradient state in the rule base, thereby disrupting edge continuity. In general, there is high similarity between the edge image and the original image. The study evaluates the binary classification ability of various algorithms through sensitivity (TPR) and specificity (True Negative Rate, TNR). The average values of TPR and TNR when detecting different images by each method are shown in Figure 10.

Figure 10 (a) shows the TPR statistics of the detection results of different categories of images by each method, and Figure 10 (b) shows the TNR statistics of the detection results of different categories of images by each method. In Figure 10 (a), various types of images had different features, resulting in significant variations in TPR for each method. The research method achieved high TPR in edge detection of various categories of images, and could accurately classify foreground pixels in the image. The highest TPR was 47.79% when processing image 4. In image category 8, the TPR of low contrast blurred images (medical X-rays) decreased to 22.63%, mainly due to insufficient response of dynamic blur rules to weak gradients, resulting in edge continuity fracture. From Figure 10 (b), the research method had the highest TNR for edge detection of various types of images, and was more accurate in determining non-edge points in the images. The TNR was the highest when processing image 8, which was 98.87%. In image category 1, the TNR of high-frequency texture interference images (forest remote sensing images) was relatively low, at 93.26%, due to the limitation of the adaptive median filtering window size, which completely suppressed dense texture noise. Further experiments showed that when the local gradient standard deviation of images was below 5, the dynamic parameter adjustment delay was about 0.5ms, resulting in the membership function being unable to adapt to the changes in local features over time. In general, the research method has good binary classification ability in edge detection. The hardware resource utilization of various algorithms in the image edge extraction is shown in Table 4. The hardware resources in Table 4 are Lookup Table Random Access Memory (LUTRAM), Flip-Flop (FF), Lookup Table (LUT), and Block Random Access Memory (BRAM).

Table 4: The consumption of hardware resources of each part of the system by different algorithms.

Hardware resource			**LUTRAM**	**FF**	**LUT**	**BRAM**
Available resource			6000	35500	17800	60
Detection method	Canny	Utilized resources	617	4256	3614	5.8
		Resource Utilization (%)	10.28%	11.99%	20.30%	9.67%
	Roberts	Utilized resources	482	3566	3083	4.3
		Resource Utilization (%)	8.03%	10.05%	17.32%	7.17%
	Sobel	Utilized resources	450	3083	2540	4.1
		Resource Utilization (%)	7.50%	8.68%	14.27%	6.83%
	Fuzzy control	Utilized resources	395	2862	2137	3.4
		Resource Utilization (%)	6.58%	8.06%	12.01%	5.67%
	This study	Utilized resources	236	2137	1500	3.2
		Resource utilization (%)	3.93%	6.02%	8.43%	5.33%

According to Table 4, the total hardware resource utilization rate of the fuzzy control method was 32.32%. The research method showed that the resource utilization rates of lookup table memory, storage, lookup table, and block random access memory were 3.93%, 6.02%, 8.43%, and 5.33%, respectively. The total hardware resources of the system accounted for 23.71%, which was 8.61% lower than before optimization. On the Xilinx Zynq-7020 FPGA platform, the algorithm increased from 28 fps to 35 fps, meeting the real-time video processing requirements (30 fps). Power consumption test showed that the energy efficiency ratio of the dynamic fuzzy control module was 3.2 GOPS/W, which was 77.81% higher than the traditional Canny algorithm (1.8 GOPS/W). It is suitable for UAV navigation and other low-power scenarios. In the NVIDIA Jetson Nano embedded device, the memory footprint decreased from 412MB before optimization to 327MB, supporting the simultaneous execution of other image analysis tasks such as object detection. From this, the research method can complete image processing with low hardware resource consumption.

5 Discussion

The dynamic fuzzy control edge detection method presented in this paper shows significant advantages in performance index, adaptability, and resource efficiency. Experimental data showed that on the BSDS500 dataset, the PSNR of the proposed method under the maximum noise (0.5%) reached 4.43dB, which was significantly better than the sharp-guided filtering method proposed by Raheja S (reference [11], PSNR=3.50dB) by 26.57%. The SSIM was 0.0367, which was 7.94% higher than that of the Canny fuzzy logic method proposed by Kumawat A (reference [10], SSIM=0.0340), verifying its advantage in preserving image structure information. In addition, the TPR of the proposed method reached 47.79%, 25.76% higher than 38.00% in reference [5]. The TNR maintained at 98.87%, which was superior to Guided L0 smoothing filtering method designed by Kumar A (reference [13], TNR=98.50%) by 0.37%. These improvements benefit from the adaptive mechanism of dynamic fuzzy rules. The membership function parameters are adjusted in real-time by Takagi-Sugeno fuzzy reasoning, which solves the edge breakage caused by fixed rules in traditional fuzzy methods (e.g., reference [10]). The adaptive median filter is combined to dynamically adjust the window size and retain the edge details in the denoising, avoiding the core defect of the sensitivity of the sharpening filter to noise in reference [11]. Compared with the baseline, the number of edge points detected by the research method was increased by 32.78%. This is due to the synergistic effect of dynamic fuzzy rules and noise perception parameter optimization strategies: adaptive modeling based on fourth-order expected values using grey distribution, and precise compression of computing resources through centroid deblurring.

However, in a rapidly changing image sequence, the update frequency of dynamic parameters may not fully match the changing speed of the scene, resulting in transient performance fluctuations. In addition, although the resource usage of the proposed method is low, its dynamic inference process still relies on floating point operation, which may face the computing power bottleneck in low-end hardware. Nevertheless, this method provides an efficient and reliable edge detection solution for real-time image processing through the collaborative design of dynamic fuzzy control and noise robustness optimization.

6 Conclusion

To achieve accurate edge localization of grayscale images, a grayscale image edge detection algorithm based on fuzzy control was designed. A dynamic fuzzy control was designed to address the fuzzy rule and parameter not being able to be adaptively adjusted. The edge images generated by different methods, the number of edge points extracted during the detection process, and the detection time were compared. The relevant evaluation indicators and system resource utilization of edge images were compared. The results indicated that the edge image obtained by the research method was relatively close to the original image, with clear and smooth lines. The average number of edge points detected by the research method was 78329, which was 32.78% higher than the fuzzy control method before optimization. The average detection time was 0.30ms, which was 4.76% shorter than before optimization, 27.04% shorter than Canny algorithm, 17.53% shorter than Robert’s algorithm, and 13.04% shorter than Sobel algorithm. The PSNR at maximum noise level was 4.43dB, and the SSIM was 0.0367. The maximum values of TNR and TPR were 47.79% and 98.87%, respectively, both higher than the other four methods. The research method can accurately process edge images with low system resource utilization, improving the adaptability and robustness of the edge detection method. The fuzzy rules in the research method are formulated based on the experience of relevant experts. Over reliance on complex fuzzy rules in the edge detection process also increases the complexity of the algorithm. Future research will further optimize the fuzzy rules and design more concise and effective algorithms.

References

[1] P. G. Kamto, E. Oksum, T. P. Luan, and J. Kamguia, “Contribution of advanced edge detection filters for the structural mapping of the Douala Sedimentary Basin along the Gulf of Guinea,” Vietnam Journal of Earth Sciences, vol. 45, no. 3, pp. 287-302, 2023. https://doi.org/10.15625/2615-9783/18410

[2] M. Firouzi, S. Fadaei, and A. Rashno, “A new framework for canny edge detector in hexagonal lattice,” International Journal of Engineering, vol. 35, no. 8, pp. 1588-1598, 2022. https://doi.org/10.5829/IJE.2022.35.08B.15

[3] A. Al-Amaren, M. O. Ahmad, and M. N. Swamy, “A low-complexity residual deep neural network for image edge detection,” Applied Intelligence, vol. 53, no. 9, pp. 11282-11299, 2023. https://doi.org/10.1007/s10489-022-04062-6

[4] T. Gao and J. Liu, “Application of improved random forest algorithm and fuzzy mathematics in physical fitness of athletes,” Journal of Intelligent & Fuzzy Systems, vol. 40, no. 2, pp. 2041-2053, 2021. https://doi.org/10.3233/JIFS-189206

[5] M. Versaci and F. C. Morabito, “Image edge detection: A new approach based on fuzzy entropy and fuzzy divergence,” International Journal of Fuzzy Systems, vol. 23, no. 4, pp. 918-936, 2021. https://doi.org/10.1007/s40815-020-01030-5

[6] Y. B. Yu, C. Y. Yang, Q. Deng, T. Nyima, S. Liang, and C. Zhou, “Memristive network-based genetic algorithm and its application to image edge detection,” Journal of Systems Engineering and Electronics, vol. 32, no. 5, pp. 1062-1070, 2021. https://doi.org/10.23919/JSEE.2021.000091

[7] U. A. Bhatti, M. Zhou, Q. S. Huo, S. Ali, A. Hussain, Y. Yan, Z. Yu, L. Yuan, and S. A. Nawaz, “Advanced color edge detection using clifford algebra in satellite images,” IEEE Photonics Journal, vol. 13, no. 2, pp. 1-20, 2021. https://doi.org/10.1109/JPHOT.2021.3059703

[8] S. Li, F. Zhu, and Q. Wu, “Precise positioning method of deformed workpiece by fusing edge and grayscale features,” Journal of Image and Graphics, vol. 29, no. 1, pp. 192-204, 2024. https://doi.org/10.11834/jig.221183

[9] W. Yang, W. Wu, X. D. Chen, X. Tao, and X. Mao, “How to use extra training data for better edge detection?,” Applied Intelligence, vol. 53, no. 17, pp. 20499-20513, 2023. https://doi.org/10.1007/s10489-023-04587-4

[10] A. Kumawat and S. Panda, “A robust edge detection algorithm based on feature-based image registration (FBIR) using improved canny with fuzzy logic (ICWFL),” The Visual Computer, vol. 38, no. 11, pp. 3681-3702, 2022. https://doi.org/10.1007/s00371-021-02196-1

[11] S. Raheja, and A. Kumar, “Edge detection based on type-1 fuzzy logic and guided smoothening,” Evolving Systems, vol. 12, no. 2, pp. 447-462, 2021. https://doi.org/10.1007/s12530-019-09304-6

[12] A. Raj, N. M. Noor, R. Mohemad, N. A. C. Mat, and S. Hussain, “An intelligent mathematically modified

fuzzy C-means clustering technique for fundus image segmentation for diabetic retinopathy identification," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 7s, pp. 603-612, 2024. https://doi.org/10.1016/j.psep.2022.11.005

[13] A. Kumar and S. Raheja, "Edge detection in digital images using guided L0 smoothen filter and fuzzy logic," Wireless Personal Communications, vol. 121, no. 4, pp. 2989-3007, 2021. https://doi.org/10.1007/s11277-021-08860-y

[14] R. Ranjan and V. Avasthi, "A hybrid edge detection mechanism based on edge preserving filtration and type-1 fuzzy logic," International Journal of Information Technology, vol. 14, no. 6, pp. 2991-3000, 2022. https://doi.org/10.1007/s41870-022-01059-9

[15] Q. Wang, J. Cao, and H. Liu, "Adaptive fuzzy control of nonlinear systems with predefined time and accuracy," IEEE Transactions on Fuzzy Systems, vol. 30, no. 12, pp. 5152-5165, 2022. https://doi.org/10.1109/TFUZZ.2022.3169852

[16] P. Xu, B. Liu, X. Hu, T. Ouyang, and N. Chen, "State-of-charge estimation for lithium-ion batteries based on fuzzy information granulation and asymmetric Gaussian membership function," IEEE Transactions on Industrial Electronics, vol. 69, no. 7, pp. 6635-6644, 2021. https://doi.org/10.1109/TIE.2021.3097613

[17] L. Tang, J. Xie, and D. Wu, "An interval type-2 fuzzy edge detection and matrix coding approach for color image adaptive steganography," Multimedia Tools and Applications, vol. 81, no. 27, pp. 39145-39167, 2022. https://doi.org/10.1007/s11042-022-13127-0

[18] Z. Yuan, C. Jin, and Z. Chen, "Research on language analysis of English translation system based on fuzzy algorithm," Journal of Intelligent & Fuzzy Systems, vol. 40, no. 4, pp. 6039-6047, 2021. https://doi.org/110.3233/JIFS-189443

[19] R. S. Almeida, F. V. da Silva, and S. S. Vianna, "Combining the bow-tie method and fuzzy logic using Mamdani inference model," Process Safety and Environmental Protection, vol. 169, pp. 159-168, 2023. https://doi.org/10.1016/j.psep.2022.11.005

[20] M. A. Hatefi, "An improved rank order centroid method (IROC) for criteria weight estimation: an application in the engine/vehicle selection problem," Informatica, vol. 34, no. 2, pp. 249-270, 2023. https://doi.org/10.15388/23-INFOR507

[21] M. Murugesan, K. Kaliannan, S. Balraj, K. Singaram, T. Kaliannan, and J. R. Albert, "A hybrid deep learning model for effective segmentation and classification of lung nodules from CT images," Journal of Intelligent & Fuzzy Systems, vol. 42, no. 3, pp. 2667-2679, 2022. https://doi.org/10.3233/JIFS-212189

[22] Y. Li, Z. Li, Y. Shen, and Z. Guo, "Infrared small target detection via center-surround gray difference measure with local image block analysis," IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no.1, pp. 63-81, 2022. https://doi.org/10.1109/TAES.2022.3189336

[23] L. Wang, C. Zhao, H. Xue, and J. Lu, "The expert synthesis and integration research method based on hesitant fuzzy language," Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice, vol. 41, no. 8, pp. 2157-2168, 2021. https://doi.org/10.12011/SETP2019-2276

[24] X. Xie, C. Wei, Z. Gu, and K. Shi, "Relaxed resilient fuzzy stabilization of discrete-time Takagi-Sugeno systems via a higher order time-variant balanced matrix method," IEEE Transactions on Fuzzy Systems, vol. 30, no. 11, pp. 5044-5050, 2022. https://doi.org/10.1109/TFUZZ.2022.3145809

[25] W. Jiang, K. Q. Zhou, A. Sarkheyli-Hägele, and A. M. Zain, "Modeling, reasoning, and application of fuzzy Petri net model: a survey," Artificial Intelligence Review, vol. 55, no. 8, pp. 6567-6605, 2022. https://doi.org/10.1007/s10462-022-10161-0

[26] F. Abedi, "Dense residual network for image edge detection," Multimedia Tools and Applications, vol. 83, no. 42, pp. 90227-90242, 2024. https://doi.org/10.1007/s11042-024-19264-y

[27] H. A. T. Abdullah, R. Z. Mahmood, S. M. A. Zber, R. A. Mohammed, A. H. ALRifaie, M. R. Ahmed, and A. W. Talab, "FPGA-based three edge detection algorithms (Sobel, Prewitt and Roberts) implementation for image processing," PRZEGLAD ELEKTROTECHNICZNY, vol. 100, no. 2, pp. 29-33, 2024. https://doi.org/10.15199/48.2024.02.05