International Journal of Intelligent Communication and Computer Science
Vol. 3, No. 3, 2025, Pages 16-31.

A Hybrid Rule-Guided Convolutional Neural Network Framework for Enhanced Image Edge Detection

Rakesh Ranjan

Department of Computer Science and Engineering, ABES Engineering College, Ghaziabad, U.P. INDIA

Email: iiirakeshranjan@gmail.com

Research Paper

Received: 17 Jan 2026, Revised: 1 March. 2026, Accepted: 4 Apr, 2026

Abstract:

This paper presents a novel edge detection framework based on Rule-Based Convolutional Neural Networks (RBCNNs) to improve the accuracy and reliability of edge extraction from digital images. Traditional edge detection techniques often struggle to identify subtle boundaries and complex structural patterns, particularly in challenging imaging environments. To overcome these limitations, the proposed approach integrates rule-based decision-making with the powerful feature learning capability of convolutional neural networks. By incorporating domain-specific rules into the learning process, the model benefits from both structured reasoning and deep feature representation. This hybrid framework not only enhances edge detection performance but also improves the interpretability of the decision-making process. Extensive experiments conducted on multiple benchmark datasets demonstrate the effectiveness and robustness of the proposed method across diverse imaging conditions. The results show significant improvements in detecting fine edges, complex boundaries, and intricate image details. Furthermore, the rule-based component provides greater transparency, enabling a better understanding of how edge decisions are made and facilitating model refinement when required. The proposed method outperforms several existing approaches, achieving a Figure of Merit (FoM) of 0.49 and an F-score of 0.82, highlighting its effectiveness for accurate and reliable image edge detection.

Keywords: CNN, Edge detection, Accuracy, Rules.

1. Introduction

Edge detection is a fundamental task in image processing and computer vision, serving as the basis for applications such as object recognition, image segmentation, feature extraction, and scene analysis [1]–[3]. Edges represent significant changes in image intensity and often correspond to the boundaries of objects or regions of interest. In binary images, an edge can be viewed as a boundary pixel separating foreground and background regions [1]. Accurate edge information is essential for understanding image structure and improving the performance of higher-level vision systems. According to Torre et al. [4], edges are formed by abrupt variations in brightness, texture, or colour, while Shui et al. [5] described them as important cues for distinguishing objects from their surroundings.

Traditional edge detection techniques, including the Canny [6] and Sobel [7] operators, identify edges by applying convolution masks to highlight intensity variations. Although these methods are computationally efficient and widely used, their performance is often affected by noise, texture complexity, illumination changes, and object occlusion. The selection of appropriate thresholds and convolution masks remains a significant challenge, frequently resulting in false edge detection or missing important boundary information [8], [9]. Even advanced approaches such as the Canny detector, which employs dual thresholds, may require additional post-processing operations to achieve satisfactory results under varying imaging conditions.

Robust edge detection is particularly important in modern applications where accurate boundary information directly influences subsequent analysis. In medical imaging, edge detection assists in identifying organs, tissues, tumours, and other anatomical structures, thereby supporting diagnosis and treatment planning [10], [11]. Similarly, in industrial inspection, object recognition, automation, and quality control systems, reliable edge extraction helps determine object shape, detect defects, and monitor structural integrity [12], [13]. These growing demands highlight the need for intelligent edge detection techniques that can effectively handle complex image characteristics while maintaining high accuracy and robustness across diverse applications.

1.1 Motivation

Edge detection is a fundamental component of computer vision and image processing, playing an important role in applications such as object recognition, image segmentation, and image enhancement. Traditional edge detection techniques, including Sobel, Canny, and Laplacian operators, rely on gradient-based calculations to identify intensity changes within an image. Although these methods are widely used, their performance often deteriorates in the presence of noise, illumination variations, and complex image structures. In addition, they usually require careful parameter tuning and may struggle to capture fine edge details in real-world scenarios. Recent advances in deep learning have led to the development of CNN-based edge detection methods, which can automatically learn meaningful features from image data and often achieve superior performance compared to conventional approaches. However, these methods typically depend on large training datasets and substantial computational resources. To address these limitations, combining rule-based edge extraction with CNN learning offers a promising alternative. Rule-based approaches provide explicit knowledge about edge patterns and are generally more resilient to noise and lighting variations, while CNNs excel at learning complex feature representations directly from data. By integrating these complementary strengths, a hybrid framework can achieve more accurate and robust edge detection while reducing the dependence on extensive training data and computational complexity.

1.2 Contributions

This study presents a novel edge detection framework that integrates convolutional neural networks (CNNs) with rule-based edge pattern recognition to improve detection accuracy and robustness. The proposed approach incorporates a set of edge-specific rules derived from fundamental edge structures, providing explicit guidance during the detection process and enhancing performance in the presence of noise, illumination changes, and complex image backgrounds. In addition, a customized CNN architecture is developed to learn discriminative edge features automatically from image data. By combining data-driven learning with structured rule-based knowledge, the framework effectively captures both local edge characteristics and complex visual patterns. The proposed method also reduces reliance on large training datasets and extensive computational resources, making it a practical and efficient solution for real-world edge detection applications across diverse imaging environments.

1.3 Organization of the paper

The structure of the remainder of this paper is organized as follows: Section 2 presents a comprehensive overview of related work. In Section 3, the proposed edge detection method is thoroughly discussed. Section 4 delves into the results obtained, and finally, Section 5 encapsulates the major conclusions drawn from this study.

2. Literature Survey

Traditional edge detection techniques based on convolution masks have played a significant role in the development of image processing systems. Early methods such as the Sobel and Prewitt operators primarily relied on pixel intensity gradients to identify edge locations [14]. While these approaches are simple and computationally efficient, they often generate false edges and produce relatively thick edge boundaries, which can obscure fine image details. To improve detection performance, researchers have explored alternative masking strategies, including hexagonal mask structures. In these approaches, conventional square masks are transformed into hexagonal patterns through interpolation, leading to improved edge representation and detection accuracy [15]. The integration of hexagonal masking with the Canny edge detector has also demonstrated enhanced performance compared to traditional implementations [16]. Furthermore, Canny-based edge detection has been successfully applied in content-based image retrieval systems [17].

Despite their widespread use, traditional mask-based methods have several limitations. Their performance is highly sensitive to image noise, illumination variations, and low-contrast regions, often resulting in inaccurate or incomplete edge extraction. In addition, fixed convolution kernels lack the flexibility required to handle complex textures and diverse image structures. Since these methods depend primarily on local gradient information, their effectiveness decreases when processing challenging real-world images. Consequently, more advanced approaches based on machine learning and deep learning have emerged to overcome these limitations and provide more accurate, adaptive, and robust edge detection capabilities.

2.1 Soft Computing-Based Image Edge Detection

Soft computing approaches have also been widely explored for image edge detection. Among these, Ant Colony Optimization (ACO) has been used to identify edge structures by mimicking the foraging behavior of ants, and its performance has been further enhanced through the integration of guided image filtering techniques to improve edge localization and reduce noise effects [18]–[20]. Researchers have also proposed modifications to the traditional Sobel operator by introducing eight-directional masks and entropy-based thresholding methods, leading to improved edge extraction capabilities [21]. Another notable approach combines image sharpening with Particle Swarm Optimization (PSO) to enhance edge visibility and optimize detection performance [22].

Although these methods have shown promising results, several challenges remain. ACO-based techniques often involve high computational costs and are sensitive to parameter selection, making them less suitable for real-time applications. Their performance may also degrade in images containing significant noise or highly complex textures. Similarly, the use of multi-directional Sobel masks and entropy-based thresholding can increase algorithmic complexity and implementation difficulty. PSO-based approaches, while effective in optimization, require careful tuning of parameters and may be influenced by initial conditions, which can affect detection consistency. These limitations highlight the need for more adaptive and efficient edge detection frameworks capable of delivering accurate results across a wide range of imaging conditions.

2.2 Fuzzy Logic-Based Image Edge Detection

Fuzzy set-based edge detection methods are founded on fuzzy set theory, where pixel intensities are represented using membership functions. To enhance edge detection performance, researchers have integrated guided image filtering with fuzzy logic [23, 24]. Kaur et al. proposed an edge detection approach based on sixteen fuzzy rules [25]. More recent studies have explored advanced fuzzy frameworks, particularly Type-2 fuzzy logic, to overcome the uncertainties and limitations associated with conventional edge detection techniques [26, 27]. Other developments include adaptive neuro-fuzzy systems for edge detection [28], as well as hybrid approaches that combine Ant Colony Optimization (ACO) with fuzzy logic to reduce false edge detection [29]. Additionally, Kalman filtering and artificial neural networks (ANNs) have been incorporated into edge detection frameworks to improve detection accuracy [30].

Despite their effectiveness, fuzzy set-based edge detection techniques present several challenges. The integration of guided image filtering and multiple fuzzy rules increases computational complexity, making the methods more resource-intensive. Although Type-2 fuzzy logic enhances the handling of uncertainty, it further complicates the system design and computational requirements. Similarly, adaptive neuro-fuzzy models and hybrid approaches involving ACO, Kalman filtering, or ANNs introduce additional processing overhead and parameter dependencies. These factors can limit their suitability for real-time applications and make performance highly sensitive to parameter tuning.

2.3 Machine Learning-Based Methods

Martin et al. proposed the Probabilistic Boundary (Pb) edge detection method, which utilizes texture feature descriptors and logistic regression to improve edge identification accuracy [31]. Building on this work, a multi-scale extension known as the Multi-scale Probabilistic Boundary (MsPb) method was introduced to enhance edge detection performance [32]. By analyzing image features at multiple scales, MsPb captures edge information more effectively across varying levels of detail, resulting in improved detection quality. Further advancements were made by Arbelaez et al., who extended the Pb framework to develop the Global Probabilistic Boundary (gPb) method [33]. This approach integrates multi-scale processing with spectral clustering, enabling more effective analysis of complex image structures by grouping similar edge features and improving the overall accuracy of edge detection. Despite their improved performance, Pb-based edge detection methods have several limitations. The original Pb framework relies on texture descriptors and logistic regression, which can be computationally demanding and sensitive to parameter selection. The more advanced MsPb and gPb methods further increase computational and memory requirements due to their use of multi-scale analysis and spectral clustering techniques. While these enhancements improve edge detection accuracy, they also introduce greater algorithmic complexity, making implementation more challenging. Consequently, the high computational cost and resource consumption of these methods may restrict their suitability for real-time applications and large-scale image processing tasks.

2.4 Deep Learning-Based Methods

The significance of supervised learning in image processing tasks such as edge detection continues increasing. Dollar et al. proposed an improved edge detection method based on probabilistic boosting tree classification by integrating probabilistic models [34]. Rahebi et al. leveraged artificial neural networks (ANNs) for edge detection, given the ability of the network to learn complex patterns in the data [35]. Lim et al. utilized a random forest classifier for sketch marker-based edge detection, showing that ensemble methods could handle well the variability of edge features [36]. In addition, edge refinement has been accomplished with cascaded convolutional neural networks (CNNs), which maintain effective performance in enhancing edge precision [37]. One major challenge concerning supervised learning methods arises in that they require large annotated datasets for training, considered truly time-consuming and costly to compile. Another difficulty with employing supervised learning techniques is their higher sensitivity towards the quality of the training data used. Poorly built or wrong datasets will lead to erroneous detection of edges by a model, thus limiting its applicability in real-life situations.

On the other hand, unsupervised learning provides an alternative way because it does not involve any manual labeling of edge features. With techniques such as sparse code gradients (SCG) [38] and pointwise mutual information architecture [39], edge contours can be identified without the need for annotated data. Yang et al. proposed a convolutional encoder-decoder network that automatically extracts object contours to enhance edge detection under various situations [40]. Likewise, Xia et al. offered another unsupervised semantic segmentation method geared toward edge detection, this time using an encoder-decoder framework to segment and recognize edge features automatically [41]. These unsupervised methods present an excellent avenue for edge detection since they require no labelled datasets, allowing for easier application to novel unlabelled data. In contrast, although unsupervised methods cannot match supervised methods for accuracy, they find none of these patterns correspond to true edge features.

2.5 State of the Art Methods

Recent studies have focused on improving edge detection accuracy in the presence of noise, complex backgrounds, and varying image conditions. Yin et al. proposed a Fusion Difference Convolution (FDC) method for edge detection that employs multiple convolution-based fusion strategies to achieve precise and robust edge extraction [42]. By effectively suppressing noise while preserving important edge details, the proposed approach demonstrates superior performance compared to conventional edge detection techniques.

Elharrouss et al. introduced a modified edge detection framework based on a Cascaded High-Resolution Convolutional Network (CHRNet) [43]. Their approach integrates multi-scale feature representations and enhances the resolution of edge maps, resulting in more accurate edge localization and improved fine-to-coarse edge representation. The method was specifically designed to address the challenges of high-resolution edge detection in complex visual environments. Liu et al. proposed a learning-based approach that incorporates gradient information to achieve crisp edge detection [44]. By combining traditional gradient-based techniques with learned feature representations, the method improves edge localization and produces sharper edge boundaries across a variety of imaging conditions. In a subsequent study, Liu et al. further enhanced edge detection performance by incorporating second-order derivative information [45]. Their findings demonstrated that higher-order image derivatives can effectively capture finer structural details, leading to more precise and well-defined edge maps.

All these studies point to the recent development in edge detection, where the marriage of higher-order information, learning techniques, and multi-scale techniques enhances edge detection performance considerably across diverse and challenging image types.

2.6 Research Gaps

In the last few decades, edge detection has grown immensely; however, there are still gaps in addressing issues like robustness to noise, complicated textures, and versatility to different light conditions. Scaling up to Canny and Sobel, the classic edge detection usually fails with images corrupted by noise or in the presence of complicated backgrounds, resulting either in missing of edges or discerning false-on edges. It's here! While deep-learning-based methods generally using CNNs would seem to work better under these terms, they need quite huge amounts of training data to be labelled, and they are expensive in terms of computation. Moreover, some techniques that are fast and adaptive at the same time for edge detection in real-time applications on embedded systems or mobile devices, which have narrowed-down computational resources, are still in demand. These gaps certainly indicate further research in developing edge detection algorithms that yield accurate yet computationally efficient results, primarily in dynamic and awkward environments.

2.7 Novelty of the Proposed Method

The edge detection methods based on CNNs tend to learn some complex features, but they get easily affected by noise which leads to false positives and false negatives. Thus, a rule-based CNN approach was proposed. This method exploits the complementary advantages of learning through the CNNs and the application of predefined rules to make it more robust to the effects of noise. Furthermore, adaptive thresholding is employed to differentiate between a true edge and a noise edge threshold and also computational complexity reduces due to rule-based structure which is the novelty of proposed work. This hybrid method exploits learned features while providing refinement from rule-based approaches, thereby offering better edge detection performance under noisy conditions.

3. Proposed Method

The present work leverages a hybrid framework that combines CNN-based and rule-based edge detection techniques, effectively integrating the strengths of both approaches. The CNN component automatically learns rich hierarchical features directly from raw image data, providing strong robustness to noise and enabling accurate edge detection across varying scales and levels of abstraction. This makes it particularly effective for complex and heterogeneous datasets. In parallel, the rule-based method contributes simplicity, computational efficiency, and interpretability, making it well-suited for real-time applications and scenarios with limited annotated data. By integrating these complementary strategies, the proposed approach achieves a well-balanced trade-off among accuracy, computational efficiency, and reliability, thereby enhancing performance across diverse edge detection tasks.

A CNN-based edge detection method is represented in Figure 1. The method comprises the input of thirty edge patterns (rules) to the CNN architecture. These edge patterns serve as instances of the complex structures that the network will need to identify in the input data. The CNN employs convolutional layers to sift through and extract features from the input edge patterns. With the convolution operations, the network learns hierarchical representations, identifying significant local patterns marking edges in the data. The extracted edge features are then used to form a feature vector which encodes the essential aspects of the input patterns. This feature vector could be said to give a rich representation of the detected edge information, from faint features to big ones, present in the input data. Then, the second stage involves classification using the sigmoid function: the sigmoid is constricted to produce values between 0 and 1 and works best for binary classification. Within the application of edge detection, the sigmoid classifier separates edge from non-edge pixels based on the input feature set. Exiting the sigmoid is the measure of probability; the closer the value to unity, the more likely it is to be classed as an edge pixel. On the other hand, the value of zero demonstrates more of a count toward the classification of being a non-edge pixel. Using a method of binary classification, it is able to identify pixels that form edges in the input data. In this particular edge detection method based on CNN shown in Figure 1, deep learning has successfully used to label complex patterns inside complex datasets. With a full reliance on convolutional layers to extract hierarchical features, followed by a classification based on sigmoid probability, we found that this method efficiently solves the edge detection and classification problem, which is indeed important for different image processing and computer vision applications.

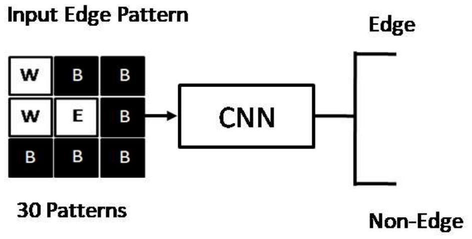
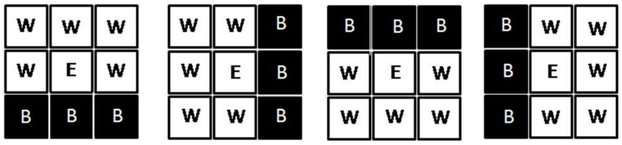


Figure 1: Schematic of the CNN based edge detection process

The formulation of the rules for a 3 × 3 mask is thoroughly defined in Figure 2, specifying the criteria to discern edge pixels in the context of the drawing. The mask is treated as a 3 × 3 array, the pixel states being denoted as follows: "W" for white pixels, "B" for black pixels, and "E" for edge pixels. For the classification of edge pixels according to the configuration of the surrounding pixels, 30 rules have systematically been devised [46]. This set of rules is essential to validate a pixel as either noise, non-edge, or possible edge.

The rule set specifies the different types of pixel arrangements to take into account for edge identification. A major part of the algorithm consists of the particular conditions concerning the identification of pixels that should not be taken into consideration as an edge. For example, a pixel surrounded by eight neighbouring pixels of the same colour, either all white or all black, is classified as noise. This condition brings forth the notion that uniformity in a pixel's neighbourhood suggests that it did not meaningfully participate in edge detection and can therefore be disregarded as being irrelevant in the edge identification context. Further, the argument follows that any situation whereby a single pixel change occurs within the neighbouring configuration will likewise be classified as a non-edge. This condition maintains that any important contrast or change between pixels is what should be qualified to represent true edge features. An important part of the rule set has to do with the identification of edge pixels when juxtaposed between two different colours. For instance, if at least two white pixels are surrounded by black ones or if at least two black ones are surrounded by white, such a situation could represent a possible edge. This rule highlights the importance of contrast and transition zones among one colour and another that are significant in the precise detection of edges. Contrast between adjacent colours represents a change in intensity or change in boundaries which act as a major indicator to edges. The rules therefore cover noise, non-edge, and possible-edge pixels in a thorough manner along with a detailed account of rules in 3x3 masks. The rule set elaborates a compromise between structured and efficient ways of identifying edges, enabling the system to disentangle a maelstrom of pixel data to come up with some meaningful features. The considerations are so framed that only pixels with intensive change in the neighbourhood are accepted as edges, thereby being more accurate. The rule-based approach sheds light on the principles on which edge detection is based, hence being very useful for applications, including but not limited to, medical image investigations, the field of computer vision, and object-recognition tasks.



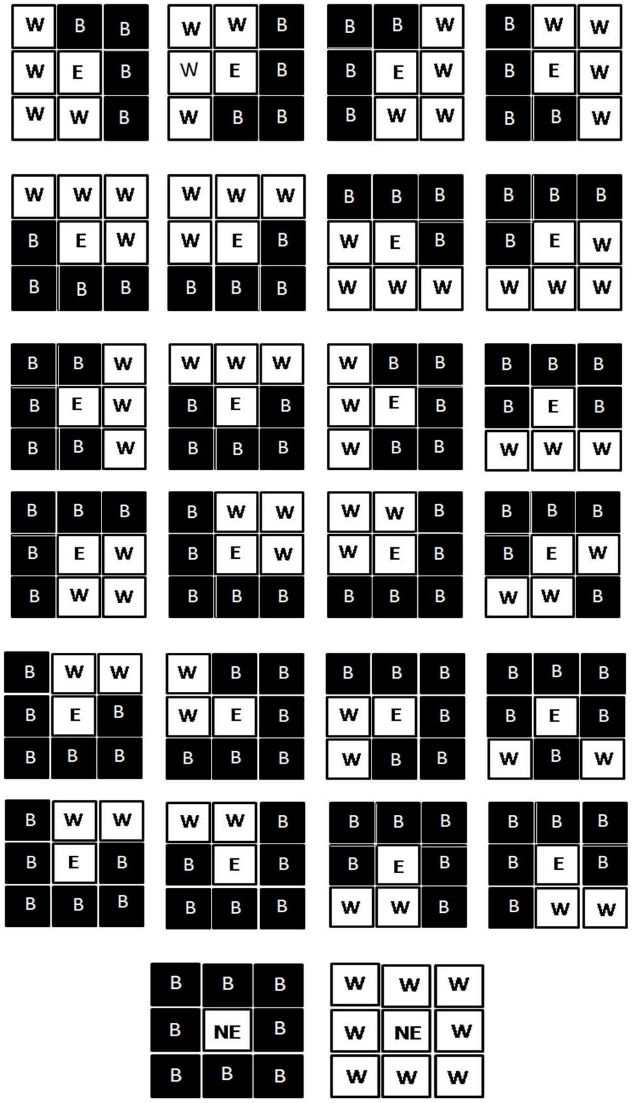


Figure 2: Schematic of the 30-rule base [39]

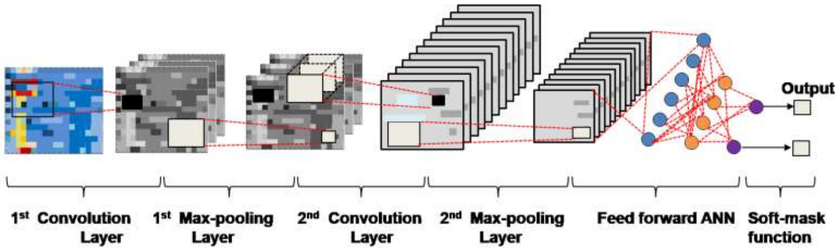


Figure 3: Schematic of detailed CNN model

To design a CNN architecture for edge detection, we focus on a simplified yet effective model using two convolutional layers and two max-pooling layers (Figure 3). The input to this network is an image that is grayscale. The first convolutional layer applies 32 filters with a kernel size of 3×3, using ReLU activation and 'same' padding to preserve the spatial dimensions of the input image. This layer captures the initial edge patterns and fine details. The output from this layer is then fed into a second convolutional layer, which also uses 32 filters with the same kernel size and activation function, further refining the detected features.

Following the convolutional layers, the first max-pooling layer, with a pool size of 2×2 and a stride of 2, reduces the spatial dimensions by half, retaining the most significant features detected by the convolutions. This process is repeated with a second set of convolutional and max-pooling layers. The second convolutional layer again uses 64 filters with a 3×3 kernel, ReLU activation, and 'same' padding, allowing the network to capture more complex patterns. The subsequent max-pooling layer further reduces the spatial dimensions, facilitating the network's focus on more abstract features.

Finally, the output from the second max-pooling layer is flattened and passed through a fully connected layer with 512 units, utilizing ReLU activation to aggregate the learned features. The final output layer employs a dense layer with a sigmoid activation function, producing a probability map that indicates the presence of edges at each pixel location in the input image. This architecture, with its combination of convolutional and max-pooling layers, effectively balances feature extraction and dimensionality reduction, making it well-suited for precise and accurate edge detection in diverse image datasets.

Non-linear Activation Function:

The non-linear activation function introduces non-linearity into the network. ReLU, represented by $f(x) = \max(0, x)$ is commonly used, where *x* is the input to the function.

Fully Connected Layers:

Fully connected layers produce the final output of the network, given by

$$O_i = \mathrm{Re}LU(\sum_j W_{ij} X_j + b) \qquad (1)$$

In this equation, *W* denotes the weights, *X* is the input, *b* represents the bias, O_i is the output at position *i*, and ReLU is the activation function.

Training with Backpropagation:

During training, CNNs use backpropagation and optimization algorithms to adjust weights. The weight update rule is given by $\Delta W = -\eta \frac{\partial E}{\partial W}$, where η is the learning rate, E is the error, and $\frac{\partial E}{\partial W}$ is the gradient of the error with respect to the weights. These mathematical expressions illustrate the core operations and concepts within Convolutional Neural Networks, highlighting their role in learning hierarchical representations from input data.

In this work, both CNN-based and rule-based edge detection methods are combined to leverage the strengths of each approach. The CNN component brings the ability to automatically learn complex features from raw image data, providing robustness to noise and high accuracy in detecting edges at different scales and abstraction levels. This adaptability is particularly useful for handling diverse and complex image data. On the other hand, the rule-based method offers simplicity, computational efficiency, and transparency, making it suitable for real-time applications and environments where labelled data is scarce. By combining both techniques, the proposed method achieves a balanced approach that maximizes accuracy, minimizes computational cost, and ensures reliable performance across different edge-detection tasks.

4. Results

In Figure 4, the loss versus epochs plot illustrates the training dynamics of a CNN applied to edge detection. At the outset of training, the loss is initially high, reflecting the model's random initialization and the disparity between its predictions and actual targets. As training progresses, the loss consistently diminishes, indicating that the model is learning to capture the relevant features in the input data.

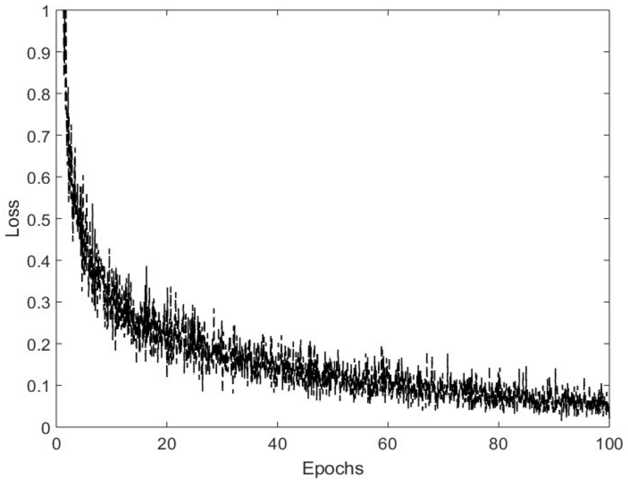


Figure 4: Loss vs Epochs

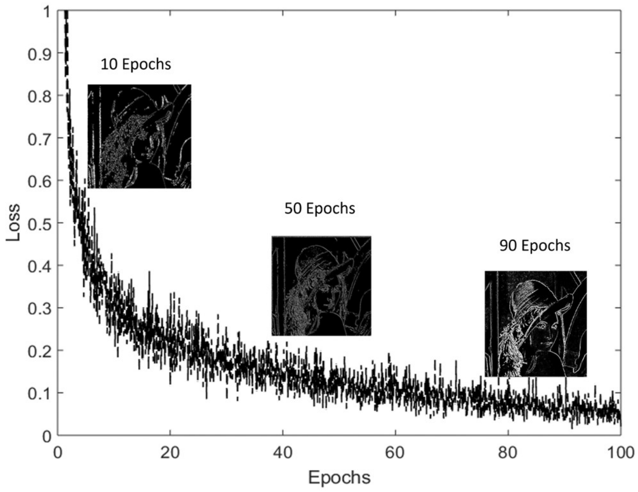


Figure 5: Detected edges at different epochs

Around 100 epochs, the loss converges to approximately 0.05, suggesting a highly accurate fit to the training dataset. This low loss value implies that the CNN has successfully discerned the intricate patterns associated with edge features. However, it is imperative to further assess the model's performance on validation or test datasets to gauge its ability to generalize to new, unseen data. Moreover, considerations regarding overfitting or underfitting should be taken into account, and potential hyperparameter tuning may be explored to optimize the model further. The overall shape and behaviour of the loss curve in Figure 5 provide insights into the training stability and effectiveness of the CNN for edge detection.

The quality of edge-detected images is assessed after 10, 50, and 90 epochs of training a CNN-based edge detection model and shown in Figure 5. As the number of epochs increases, the model has more opportunities to learn and fine-tune its parameters, leading to progressively better performance. After 10 epochs, the edge detection results are rudimentary, capturing only the most prominent edges while missing finer details and potentially misidentifying noise as edges. By 50 epochs, the model's ability to distinguish between true edges and noise improves, resulting in clearer and more accurate edge maps with better-defined contours and fewer false positives. At 90 epochs, the model has reached a higher level of refinement, capturing intricate edge details with high precision and robustness against noise. This progressive improvement demonstrates the model's enhanced learning and adaptation capabilities, leading to superior edge detection quality with increased training epochs.

4.1 Qualitative Results

In evaluating the effectiveness of our hybrid edge detection framework, we performed extensive qualitative assessments on a Lena image and Berkley segmentation dataset (BSD) images of size 512×512 [47]. Qualitative results provide a visual confirmation of the improvements our method achieves over traditional and standalone CNN-based edge detection methods. We compared the edge maps generated by our method against those produced by Sobel, Canny, and standalone CNN-based methods. Our method consistently produced sharper, more accurate edges with fewer false positives and negatives.

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

(d) ACO **(e) ACO+GF** **(f) Proposed**

Figure 6: Comparison of edge detection methods (Lena Image)

In Figure 6(a), a demonstration of Canny edge detection is presented, revealing that while the method accurately recognizes the majority of edges, there is a notable increase in erroneously accepted edges. Consequently, the Canny edge detection technique exhibits a considerable level of noise in its results. Figure 6(b) showcases the application of Sobel edge detection, where some genuine edges are correctly identified, but others are mistakenly rejected. Despite its advantages, fuzzy edge identification, as depicted in Figure 6(c), outperforms both Canny and Sobel approaches. However, this method still exhibits limitations, including the incorrect detection of the face boundary and the presence of noise in the hat area.

ACO edge detection, illustrated in Figure 6(d), indicates a mix of correct and mistaken identifications of actual edges. While some edges are accurately recognized, a considerable number are erroneously rejected. The results of Kumar et al.'s [16] edge detection method, combining ACO and guided filtering (Figure 6(e)), demonstrate a reduction in the number of falsely rejected real edges compared to the standalone ACO method. Nevertheless, there are still numerous instances of misidentifications. Figure 6(f) reveals the outcome of proposed CNN based approach, addressing the issue of broken edges observed in previous methods.

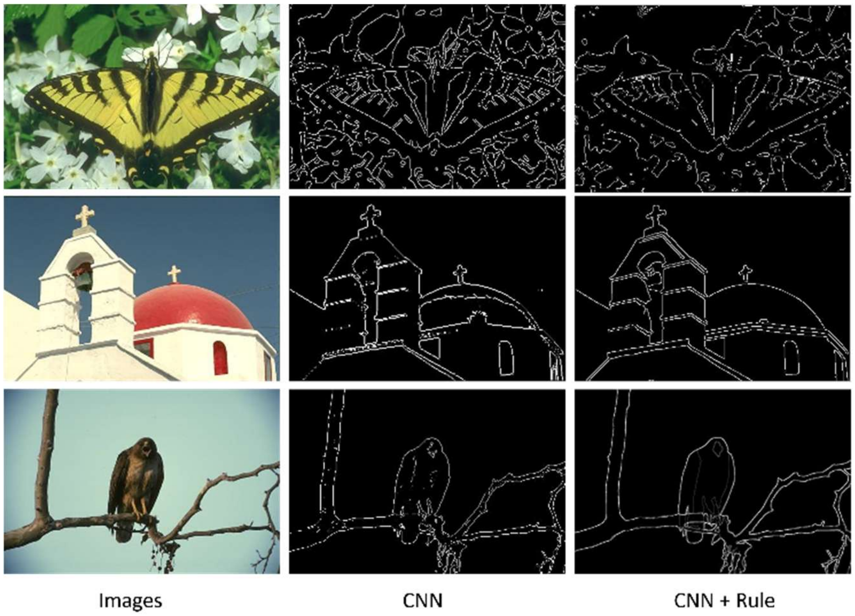


Figure 7: Comparison of edge detection with CNN and Proposed method (CNN+ Rules)

In Figure 7, the comparison of edge detection results between the standard CNN method and the proposed method (CNN+Rules) highlights significant differences in performance, particularly in noisy conditions. The standard CNN method, while effective at detecting edges, often struggles with noise, leading to false positives and less precise edge maps. The proposed method, which integrates predefined rules with the CNN, demonstrates a marked improvement. These rules help filter out noise-induced artifacts and enhance edge continuity, resulting in cleaner and more accurate edge detection. The edge maps produced by the proposed method show better-defined edges with fewer false detections, indicating its superior robustness and precision. This comparison underscores the efficacy of combining rule-based refinements with CNNs to address the limitations of noise susceptibility and improve overall edge detection quality.

4.2 Quantitative Results

For a rigorous quantitative evaluation, we utilized standard metrics such as Figure of Merit (FoM) and F-score to compare the performance of our hybrid method against baseline approaches. The FoM assesses the accuracy of edge detection by comparing detected edges to ground truth edges, with higher values indicating better performance. Our method achieved an average FoM significantly higher than traditional methods and standalone CNNs across multiple datasets, indicating superior edge detection accuracy. The F-score, which combines precision and recall into a single metric, was also used to measure the overall effectiveness of edge detection. Our approach consistently outperformed traditional and CNN-based methods, achieving higher F-scores that reflect both high precision (fewer false positives) and high recall (fewer missed edges). In summary, our hybrid edge detection framework demonstrated substantial improvements in both qualitative and quantitative evaluations, proving its effectiveness and robustness across various challenging conditions.

Figure of Merit: FoM is a metric that measures the overall performance of an edge detection algorithm. It considers both the ability to detect true edges (sensitivity) and the suppression of false edges (specificity). The FoM is defined as [48]:

$$FoM = \frac{2 \times TPR \times TNR}{(TPR + TNR)} \tag{2}$$

Where:
True Positive Rate (TPR) or Sensitivity is defined as

$$TPR = \frac{TP}{TP + FN} \quad (3)$$

True Negative Rate (TNR) or Specificity is defined as

$$TNR = \frac{TN}{TN + FP} \quad (4)$$

TP = True Positives (correctly detected edges), TN = True Negatives (correctly rejected non-edges), FP = False Positives (false alarms) and FN = False Negatives (missed edges)

F-score:
It is a harmonic mean of precision and recall, which are also useful metrics in evaluating edge detection algorithms. The Precession is defined as

$$P = \frac{TP}{TP + FP} \quad (5)$$

The Recall can be formulated as

$$R = \frac{TP}{TP + FN} \quad (6)$$

The F-score is defined as:

$$F = \frac{2PR}{P + R} \quad (7)$$

where, Precision measures the accuracy of positive predictions (edges detected) while Recall (Sensitivity) measures the ratio of actual positives (true edges) that are correctly identified.
FoM provides a holistic view by considering both sensitivity and specificity. F-score balances precision and recall to measure the performance of the edge detection.
The Matthews Correlation Coefficient (MCC) is a metric used to evaluate the quality of binary classifications. The formula for MCC is as follows:

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (8)$$

The results presented in the figures 8 and 9, offer a comprehensive evaluation of various edge detection methods, showcasing their performance metrics in terms of FoM and F-Score. Each method is associated with a specific reference and technique, providing valuable insights into their effectiveness.
The Canny edge detection method, referenced in [6] and employing masking, exhibits a FoM of 0.3 and an F-Score of 0.49. Despite its longstanding popularity, Canny demonstrates moderate performance in capturing relevant edges, as indicated by its F-Score. Sobel edge detection, referenced in [7] and also utilizing masking, shows a higher FoM of 0.42, but its F-Score of 0.40 suggests challenges in precision and recall. The limitations of Sobel's masking approach are reflected in its comparative metrics. Lim et al.'s method [36], utilizing Sketch Token, achieves a noteworthy F-Score of 0.73, showcasing a substantial capability in accurately identifying edges. However, the FoM value is not provided, making a complete assessment challenging. Kumar et al.'s Fuzzy edge detection method, referenced in [24], demonstrates a FoM of 0.4 and an F-Score of 0.64.

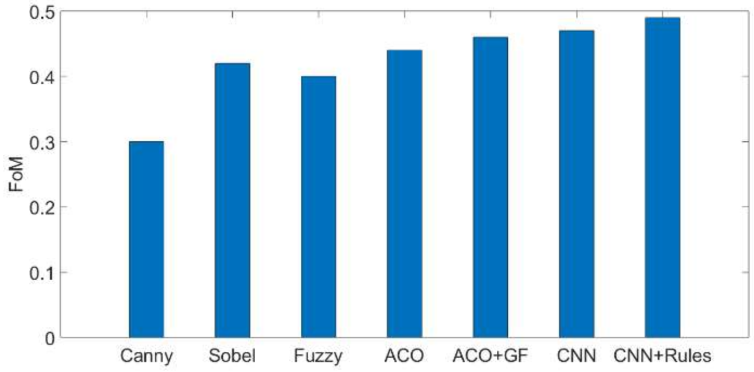


Figure 8: Comparison of edge detection methods in terms of FoM

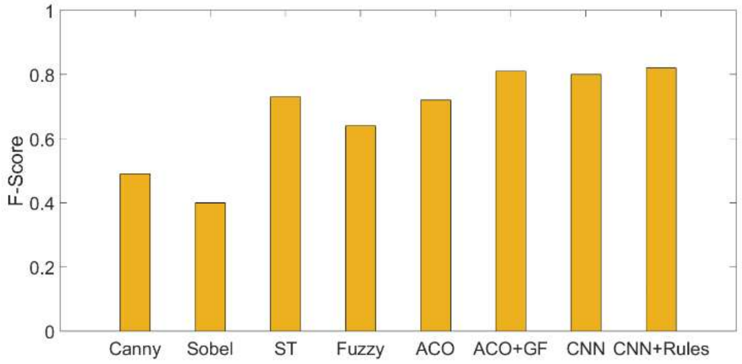


Figure 9: Comparison of edge detection methods in terms of F-Score

This approach leverages fuzzy logic to enhance edge detection, resulting in a balanced performance across the FoM and F-Score metrics. The ACO edge detection method, introduced by Kumar et al. [19], achieves a FoM of 0.44 and an impressive F-Score of 0.72. This suggests that the Ant Colony Optimization (ACO) approach is effective in capturing edges, exhibiting a high level of precision and recall. Building upon the ACO method, Kumar et al.'s ACO + Guided Filtering approach, referenced in [20], further enhances performance with a higher FoM of 0.46 and an even more substantial F-Score of 0.81. This indicates the effectiveness of guided filtering in refining edge detection results. Finally, the proposed method, employing a CNN, outperforms the referenced methods with a FoM of 0.49 and an impressive F-Score of 0.82. The CNN approach showcases superior performance in both precision and recall, affirming its efficacy in accurately detecting edges in the given context. In summary, the results provide a comparative overview of various edge detection methods, highlighting the strengths and weaknesses of each. The proposed CNN method emerges as particularly robust, surpassing other techniques in capturing edges with high precision and recall.

As shown in Figure 10, the MCC values for various edge detection methods are presented, illustrating their comparative performance. The MCC values for the different methods—Canny (0.21), Sobel (0.29), ST (0.51), fuzzy (0.48), ACO (0.53), ACO + GF (0.57), CNN (0.51), and CNN + rules (0.71)—demonstrate the progression from traditional methods to more advanced deep learning-based and hybrid approaches. The graph visually highlights the superior performance of hybrid techniques like CNN + rules, which achieves the highest MCC value of 0.71, followed by methods like ACO + GF and ACO. In contrast, traditional methods like Canny and Sobel show relatively lower MCC values, indicating their limitations in accurately detecting edges compared to more sophisticated methods. This visual representation further emphasizes the enhanced accuracy and robustness of advanced and hybrid edge detection techniques, as they incorporate both optimization and learning-based strategies for improved edge recognition.

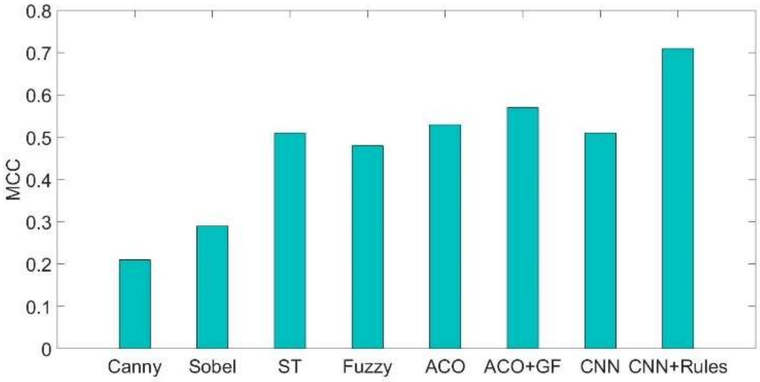


Figure 10. Comparison of edge detection methods in terms of MCC

6. Comparison with state-of-the-art methods

The comparison of various methods for edge detection presented in recent papers highlights the performance of different techniques in terms of their F-Score is compared in Table 1.

Table 1: Comparison of the state-of-the-art methods (F-Score)

Paper	Method	F-Score
Yin et al. [42]	Fusion Difference Convolution	0.819
Elharrouss et al. [43]	Cascaded High-Resolution Convolutional Network	0.816
Liu et al. [44]	Gradient Information for Crisp Edge Detection	0.805
Liu et al. [45]	Second-Order Derivative Information for Crisp Edge Detection	0.813
Proposed	Rule Based CNN	0.82

Yin et al. [42] proposed the Fusion Difference Convolution method, which achieved an F-Score of 0.819, demonstrating its effectiveness in combining multiple convolutional operations for edge detection. Elharrouss et al. [43] introduced a Cascaded High-Resolution Convolutional Network, yielding an F-Score of 0.816, showcasing its ability to detect fine details at high resolutions through successive layers of convolutional processing. Liu et al. [44] applied Gradient Information for Crisp Edge Detection, achieving an F-Score of 0.805, emphasizing the use of gradient information to improve edge detection sharpness. Similarly, Liu et al. [45] explored Second-Order Derivative Information for Crisp Edge Detection, slightly improving the F-Score to 0.813 by incorporating second-order derivative features. The proposed method, a Rule-Based CNN, achieved an F-Score of 0.82, providing a promising solution by combining the precision of rule-based systems with the adaptability of convolutional neural networks. This combination allows the proposed method to achieve a competitive F-Score while maintaining an effective balance between traditional and deep learning techniques for enhanced edge detection performance.

7. Conclusion

In conclusion, this paper presents a pioneering approach to edge detection by synergizing Rule-Based CNNs, thereby elevating precision and accuracy in edge identification. Conventional methods for edge detection often grapple with the nuanced patterns and intricate details inherent in images. In response to these challenges, our proposed methodology integrates rule-based strategies into the CNN framework, creating a dynamic and adaptive solution. The introduced model leverages the interpretability of rule-based systems while harnessing the potent feature extraction capabilities of CNNs. Through rigorous experiments and evaluations, we establish the superior performance of our approach in edge detection compared to traditional methods. The fusion of rule-based decision-making and CNNs not only enhances edge identification but also provides a transparent and interpretable framework for comprehending the complexities of edge detection in intricate image datasets. This research contributes significantly to the advancement of computer vision by introducing a promising paradigm for achieving precision edge detection in diverse applications.

References

1. Chi Z, Li H, Lu H, Y. Ming-Hsuan. Dual deep network for visual tracking. IEEE Trans Image Process 2017; 26:2005–2015.
2. Leal-Taixé L, Canton-Ferrer C, S. Konrad. Learning by tracking: Siamese CNN for robust target association. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 33-40.
3. Ojha S, Sakhare S. Image processing techniques for object tracking in video surveillance-A survey. In: 2015 International Conference on Pervasive Computing (ICPC). pp. 1-6.
4. Torre V, Poggio TA. On edge detection. IEEE Trans Pattern Anal Mach Intell 1986; 2:147–163.
5. Shui PL, Zhang WC. Noise-robust edge detector combining isotropic and anisotropic Gaussian kernels. Pattern Recogn 2012; 45:806–820.
6. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986; 8:679–698.
7. Gao W, Zhang X, Yang L, Liu H. An improved Sobel edge detection. In: 2010 3rd International Conference on Computer Science and Information Technology. pp. 67-71.
8. Sun, R., Lei, T., Chen, Q., Wang, Z., Du, X., Zhao, W. and Nandi, A.K. Survey of image edge detection. *Frontiers in Signal Processing*, 2022; 2, p.826967.
9. Rashno A, F. Sadegh. Image restoration by projection onto convex sets with particle swarm parameter optimization. Int J Eng 2023; 36:398–407.
10. Khaliluzzaman Md, Kobra K, Liaqat S. Comparative analysis on real-time hand gesture and sign language recognition using convexity defects and YOLOv3. Sigma J Eng Nat Sci 2024; 42:99-115.
11. Çiloğlu FU. Detection of COVID-19 infection by using Raman spectroscopy of serum samples and machine learning. Sigma J Eng Nat Sci 2024; 42:1892–1898.
12. Demırcıoglu P, Bogrekcı I, Sucuoglu S, Guven E. The relationship between the fusion temperature and dimensional accuracy of 3D printed parts. Sigma J Eng Nat Sci 2020; 38:21-28.
13. Rahou M. Optimization and modeling of the workpiece position. Sigma J Eng Nat Sci 2020; 38:47-59.
14. Lakshmi S, Sankaranarayanan V. A study of edge detection techniques for segmentation computing approaches. IJCA Special Issue on Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications CASCT 2010;35-40.
15. Fadaei S, R. Abdolreza. A framework for hexagonal image processing using hexagonal pixel-perfect approximations in subpixel resolution. IEEE Trans Image Process 2021; 30:4555-4570.
16. Firouzi M, F. Sadegh, R. Abdolreza. A new framework for Canny edge detector in hexagonal lattice. Int J Eng 2022; 35:1588-1598.
17. Fadaei S. New dominant color descriptor features based on weighting of more informative pixels using suitable masks for content-based image retrieval. Int J Eng 2022; 35:1457–1467.
18. Baterina AV, Oppus C. Image edge detection using ant colony optimization. WSEAS Trans Signal Process 2010; 6:58–67.
19. Kumar A, Raheja S. Edge detection using guided image filtering and enhanced ant colony optimization. Procedia Comput Sci 2020; 173:8–17.
20. Kumar A, Raheja S. Edge detection using guided image filtering and ant colony optimization. In: 2021 Recent Innovations in Computing: Proceedings of ICRIC 2020. pp. 319-330.
21. Ravivarma G, Gavaskar K, Malathi D, Asha KG, Ashok B, Aarthi S. Implementation of Sobel operator-based image edge detection on FPGA. Mater Today Proc 2021; 45:2401–2407.
22. Verma A, Dhanda N, Yadav V. Binary particle swarm optimization-based edge detection under weighted image sharpening filter. Int J Inf Technol 2023; 15:289–299.

23. Kumar A, Raheja S. Edge detection in digital images using guided L0 smoothen filter and fuzzy logic. Wirel Pers Commun 2021; 121:2989–3007.
24. Raheja S, Kumar A. Edge detection based on type-1 fuzzy logic and guided smoothening. Evol Syst 2021; 12:447–462.
25. Kaur EK, Mutenja V, Gill IS. Fuzzy logic-based image edge detection algorithm in MATLAB. Int J Comput Appl 2010; 1:55–58.
26. Aborisade DO. Novel fuzzy logic-based edge detection technique. Int J Adv Sci Technol 2011; 29:75–82.
27. Begol M, Maghooli K. Improving digital image edge detection by fuzzy systems. World Acad Sci Eng Technol 2011; 81:76–79.
28. Zhang L, Xiao M, Ma J, Song H. Edge detection by adaptive neuro-fuzzy inference system. In: 2009 2nd International Congress on Image and Signal Processing. pp. 1-4.
29. Dorrani Z, Farsi H, Mohamadzadeh S. Image edge detection with fuzzy ant colony optimization algorithm. Int J Eng 2020; 33:2464–2470.
30. Siddharth D, Saini DKJ, Singh P. An efficient approach for edge detection technique using Kalman filter with artificial neural network. Int J Eng 2021; 34:2604–2610.
31. Martin DR, Fowlkes CC, Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans Pattern Anal Mach Intell 2004; 26:530–549.
32. Ren X. Multi-scale improves boundary detection in natural images. In: 2008 10th European Conference on Computer Vision, Marseille, France. pp. 533–545.
33. Arbelaez P, Maire M, Fowlkes C, Malik J. Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 2010; 33:898–916.
34. Dollar P, Tu Z, B. Serge. Supervised learning of edges and object boundaries. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1964–1971.
35. Rahebi J, Tajik HR. Biomedical image edge detection using an ant colony optimization based on artificial neural networks. Int J Eng Sci Technol 2011; 3:8211–8218.
36. Lim JJ, Zitnick CL, Dollár P. Sketch tokens: A learned mid-level representation for contour and object detection. In: 2013 IEEE conference on computer vision and pattern recognition. pp. 3158–3165.
37. Elharrouss O, Hmamouche Y, Kamal Idrissi A, El Khamlichi B, El Fallah-Seghrouchni A. Refined edge detection with cascaded and high-resolution convolutional network. Pattern Recogn 2023; 138:109361.
38. Ren X, Bo L. Discriminatively trained sparse code gradients for contour detection. Adv Neural Inf Process Syst 2012;25.
39. Isola P, Z. Daniel, K. Dilip, A. Edward H. Crisp boundary detection using pointwise mutual information. In: 2014 13th European Conference, Zurich, Switzerland. pp. 799–814. Springer Int Publ.
40. Yang J, P. Brian, C. Scott, L. Honglak, Yang MH. Object contour detection with a fully convolutional encoder-decoder network. In: 2016 IEEE conference on computer vision and pattern recognition. pp. 193–202.
41. Xia X, Kulis B. W-net: A deep model for fully unsupervised image segmentation. arXiv preprint 2017. arXiv:1711.08506.
42. Yin Z, Wang Z, Fan C, Wang X, Qiu T. Edge detection via fusion difference convolution. Sensors 2023; 23:6883.
43. Elharrouss O, Hmamouche Y, Kamal Idrissi A, El Khamlichi B, El Fallah-Seghrouchni A. Refined edge detection with cascaded and high-resolution convolutional network. Pattern Recogn 2023; 138:109361.
44. Liu C, Zhang W, Liu Y, Li Y, Li W, Fan Y, Zhang L. Learning to utilize gradient information for crisp edge detection. arXiv preprint 2024. arXiv:2406.05779.
45. Liu C, Zhang W, Liu Y, Fan Y, Li M, Li W. Learning to utilize image second-order derivative information for crisp edge detection. CoRR 2024.
46. Ranjan R, Avasthi V. A hybrid edge detection mechanism based on edge preserving filtration and type-1 fuzzy logic. Int J Inf Technol 2022; 14:2991–3000.
47. Xie S, Tu Z. Holistically-nested edge detection. In: 2015 IEEE International Conference on Computer Vision. pp. 1395–1403.
48. Ranjan R, Avasthi V. Edge detection using guided Sobel image filtering. Wirel Pers Commun 2023; 132:651–677.