Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Directional Projection of the Hessian: A Robust Method for Edge Detection

Mohamed LAJILI

m.lajili@audensiel.fr

Audensiel https://orcid.org/0000-0002-9508-3587

Fella Hayat Laroui

Aix Marseille University

Research Article

Keywords: Edge Detection, Local Features, Second-order derivatives, Curvature analysis, Directional Moments, Noisy Image, Directional filtering.

Posted Date: March 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9115139/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Directional Projection of the Hessian: A Robust Method for Edge Detection

Mohamed Lajili and Fella Hayat Laroui

***Abstract*—Edge detection remains a fundamental component in many computer vision and image processing applications, including image segmentation, object recognition, and scene understanding. Despite extensive research, achieving both high precision and robustness in the presence of noise and structural ambiguity remains challenging. This paper introduces a novel edge detection method based on Directional Hessian Projection (DHP), which exploits second-order differential structure by projecting the local Hessian matrix onto multiple orientations. The proposed approach constructs a directional edge map by averaging the absolute directional curvatures obtained from these projections. This mathematically grounded formulation enhances the detection of curvature-rich and fine-scale structures while improving robustness to noise and filtering artifacts. Extensive experiments on synthetic and natural images, including the BSDS500 benchmark and the MultiCue dataset, show that DHP consistently outperforms several classical edge detectors such as Sobel, Canny, Marr–Hildreth, and wavelet-based methods. Quantitatively, DHP achieves lower MSE, higher PSNR and accuracy, and an average precision (AP) of 0.77 on BSDS500. On the MultiCue dataset, it reaches an ODS of 0.901 and an OIS of 0.93, with precision of 0.85 and recall of 0.96. Compared with deep learning approaches, DHP does not require large annotated datasets while narrowing the performance gap with learning-based methods.**

***Index Terms*—Edge Detection, Local Features, Second-order derivatives, Curvature analysis, Directional Moments, Noisy Image, Directional filtering.**

I. Introduction

Edges play a crucial role in visual perception, marking transitions in intensity, texture, and structure, from light to shadow, or from object to background. In computer vision and image processing, edge detection is a fundamental preprocessing step that reveals the structural backbone of an image, simplifying visual data while preserving essential information [1], [2], [3]. Whether the goal is object recognition [4], image segmentation [5], or feature extraction [6], reliably and accurately detecting edges remains one of the most critical challenges in visual processing pipelines. Over the past decades, a diverse range of approaches has been proposed, from early gradient-based methods to sophisticated learning-driven frameworks, all aiming to improve edge localization under increasingly complex imaging conditions [7], [8].

Classical Edge Detection. Early edge detection techniques were primarily based on local gradient and intensity analysis. These methods focused on identifying abrupt variations in brightness, typically indicative of physical object boundaries.

The most prominent traditional method remains the Canny detector [9], which revolutionized edge detection by integrating Gaussian smoothing, gradient estimation, non-maximum suppression, and hysteresis thresholding into a multi-step process. It remains a benchmark due to its ability to produce thin, connected, and accurate edges. Nevertheless, these classical filters are highly sensitive to noise, parameter-dependent, and tend to fail in highly textured or low-contrast environments.

Among the most well-known are the Sobel operator [10], which remains widely used due to its simplicity and efficiency, and the Laplacian of Gaussian (LoG) operator [11]. While the Sobel operator estimates the first-order derivatives of intensity using discrete convolution masks and highlighting changes in both vertical and horizontal directions, LoG operator adds a layer of robustness by combining a smoothing filter with a second-order derivative, thus improving edge detection in noisy contexts.

In general, traditional edge detection methods often struggle with noise, low contrast, and complex textures. Their fixed rules make them less effective at capturing meaningful edges across varying shapes and scales.

To overcome these limitations, more advanced techniques based on mathematical models have been proposed [12], [13], [14]. Variational models, in particular, formulate edge detection as an energy minimization problem. For example, the Mumford-Shah model [15], [16] finds edges by balancing fidelity to the image with smoothness. Other techniques, like the topological gradient method [17], [18], detect edges by analyzing local intensity changes. These methods are more robust in the presence of noise, but are usually slower and more computationally resources.

From Manual Features to Learning-Based Approaches. To overcome the limitations of traditional methods, researchers turned to learning-based approaches that rely on supervised training to predict edge locations. These methods fall into two broad categories: classical learning-based models and deep learning-based models[19].

Early learning-based edge detectors used low- and mid-level handcrafted features, such as brightness, color, texture, and gradient orientation, extracted from local image regions. These features were then combined to predict the presence of edges at each pixel location. A good example is the Probability of Boundary (Pb) model[20], which formulated edge detection as a probabilistic inference problem by modeling local brightness and texture cues independently and combining them using a logistic regression function. The gPb model [21] extended this

Mohamed Lajili is with Lab'Innov, Audensiel, 93 Rue Nationale, 92100 Boulogne-Billancourt, France (e-mail: m.lajili@audensiel.fr).
Fella Hayat Laroui is with Aix Marseille university, 163 Av. de Luminy - 13009 Marseille Cedex 7, France (e-mail: fella-hayat.LAROUI@etu.univ-amu.fr) .

approach by incorporating global cues and spectral clustering, significantly improving edge continuity and segmentation quality.

In parallel, patch-based models like Sketch Tokens [22] focused on learning structured appearance patterns from local image patches, enabling better suppression of noisy edges and improvement of boundary coherence. While these techniques proved superior to traditional filters, their reliance on handcrafted representations made them less adaptable to diverse and unseen image domains.

The emergence of deep learning has fundamentally reshaped the field. Convolutional Neural Networks (CNNs) [23], [24], [25] learn hierarchical features directly from raw image data and have demonstrated remarkable success in recognizing edges, shapes, and high-level contours across varied datasets.

Architectures like Holistically-Nested Edge Detection (HED) [26] introduced the idea of deep supervision, enforcing edge predictions at multiple layers to refine localization. Similarly, Richer Convolutional Features (RCF) [27] aggregated outputs across all convolutional stages, improving both edge crispness and robustness to scale changes. More recent models like BDCN [28] and DexiNed [29] push these innovations further by incorporating hierarchical representations and recursive refinement.

Despite their impressive performance, deep models come with important caveats. They require large annotated datasets, involve computationally expensive training, and often lack interpretability. Their internal decisions are difficult to explain or trust, especially in high-stakes applications such as medical imaging.

In this paper, we propose a new method for edge detection that uses a local directional Hessian to better understand and detect the structure of an image, which we call the Directional Hessian Projection (DHP) method. Unlike traditional edge detection methods such as Canny, which perform global analysis, our approach emphasizes local geometric context. DHP is a classical approach to edge detection that analyzes second-order information within images. Traditional edge detection techniques mainly rely on gradients to find where intensity changes occur, and there are mainly two challenges related to this. First, being a first-order derivative, the gradient fails in detecting discontinuities of second order, i.e., discontinuities of the Laplacian/Hessian. However, this information may be valuable in some cases and could be a very good indicative information for various applications. Second, gradient detection is also very sensitive to noise that is always present in the images and false edges that are caused by noise.

DHP addresses these challenges by examining the Hessian matrix, which contains second-order derivatives and describes the curvature of the image intensity surface at each point. By analyzing this curvature information, DHP gains a deeper understanding of how intensity changes evolve across different directions. This curvature-based perspective allows it to detect not only strong, well-defined edges but also finer, weaker structural details that may be missed by gradient methods.

Another advantage of DHP, compared to deep learning–based methods, is that it does not rely on any supervised learning or large training datasets. Instead, it uses purely mathematical calculations based on geometric principles, making it computationally efficient and easy to implement. Its ability to combine accuracy, robustness, and simplicity makes DHP suitable for a wide range of practical applications, particularly in situations where edges are not clearly visible or consistent performance is required across diverse image types.

This paper is organized as follows. Section 2 presents the theoretical foundation and algorithm of our proposed edge detection method. This includes a detailed discussion of the method itself, followed by a mathematical analysis of the proposed model. We provide a rigorous mathematical analysis of the proposed model. We demonstrate how the projected directional Hessian corresponds to the second-order directional derivative and serves as a principled measure of image curvature in a given orientation. In addition, we illustrate the effectiveness of our proposed method through visual comparisons on clean and noisy images. These examples highlight the ability of the DHP framework to detect both strong and subtle edges while maintaining robustness to noise. Together, these components establish the DHP model as a mathematically grounded and practically effective edge detection framework. Section 3 discusses the results; specifically, we compare our proposed method with several other methods, both in the presence and absence of noise. Finally, Section 4 concludes the paper with a summary of our findings and potential future directions.

II. Directional Hessian Projection (DHP): Proposed Edge Detection Framework

Based on the limitations of traditional edge detection methods and the insights discussed earlier, we propose our complete edge detection method, which is specifically designed to extract structural features from images in a way that is both robust to noise and sensitive to directional variations. This section presents the theoretical formulation and practical implementation of the proposed DHP method for edge detection. DHP is designed to identify edges by analyzing second-order intensity variations, providing detailed information about how brightness changes evolve locally in different directions.

DHP computes the Hessian matrix within a neighborhood around each pixel to capture local curvature information. The method then projects this matrix onto a set of uniformly sampled unit directions to obtain directional curvature responses, which are subsequently aggregated to produce an edge map that is sensitive to structural variations across all orientations. By combining second-order derivative analysis with directional projections, DHP enables the detection of both strong and subtle edges with high precision.

In what follows, we detail the mathematical formulation, geometric interpretation, and implementation strategy of this method, highlighting its unique ability to combine *curvature-based sensitivity* with *directional robustness* for superior edge detection performance.

Overview. The DHP method consists of three main steps: (i) computing the local mean Hessian matrix at each pixel over a predefined neighborhood; (ii) projecting this Hessian

onto multiple unit directions sampled uniformly over $[0, \pi[$; and (iii) aggregating the absolute directional responses to form the final edge intensity map. This process enables the detection of edges with enhanced geometric robustness and multi-directional sensitivity.

A. Mathematical Formulation

Definition II.1 (local mean Hessian)**.** Let Ω be a bounded domain in $\mathbb{R}^2$. Mathematically, an image defined on Ω is a map

$$I : \Omega \to \mathbb{R},$$
$$(x, y) \mapsto I(x, y).$$

We assume that the image I is "sufficiently" smooth and twice-differentiable, so that its Hessian matrix at each pixel (x, y) is defined as:

$$H(x, y) = \begin{bmatrix} I_{xx}(x, y) & I_{xy}(x, y) \\ I_{xy}(x, y) & I_{yy}(x, y) \end{bmatrix},$$

where I_{xx}, I_{xy}, and I_{yy} denote the second-order partial derivatives of the image intensity. These derivatives capture the curvature and provide richer structural information compared to first-order derivatives.

To ensure robustness against noise and to capture structural variations within a neighborhood, the **local mean Hessian** over a square region $R(x, y)$ of size $w \times w$ centered at (x, y), is given by:

$$H_R(x, y) = \frac{1}{|R|} \sum_{(i,j)\in R(x,y)} \begin{bmatrix} I_{xx}(i, j) & I_{xy}(i, j) \\ I_{xy}(i, j) & I_{yy}(i, j) \end{bmatrix}. \quad (1)$$

Once the local Hessian matrix is computed, we project it onto a set of unit directions to extract directional curvature information.

Definition II.2 (Projected Directional Hessian)**.** Let $\{v_k\}_{k=1}^K$ be a set of K uniformly distributed unit vectors in $[0, \pi[$, defined by:

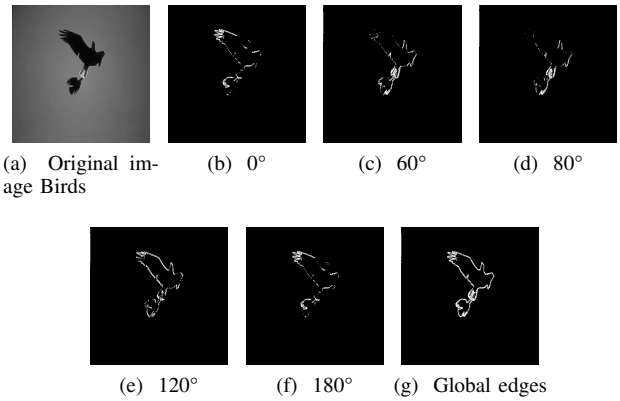
$$v_k = \begin{bmatrix} \cos\theta_k \\ \sin\theta_k \end{bmatrix}, \quad \theta_k = \frac{(k-1)\pi}{K}, \quad k = 1, \ldots, K.$$

Given the local mean Hessian matrix $H_R(x, y)$, defined in (1) at each pixel (x, y), the *projected directional Hessian* in the direction v_k is defined as:

$$H_{v_k}(x, y) := v_k^\top H_R(x, y) v_k.$$

This term represents the second directional derivative of the image in the direction v_k, i.e., the local curvature projected along v_k. The projected directional Hessian captures the second derivative along each direction, providing a measure of local curvature variation useful for edge detection.

To better illustrate the ability of the proposed DHP method to capture edges across multiple orientations, Fig. 1 shows the directional edge maps obtained by projecting the local Hessian onto five uniformly sampled directions. This visualization demonstrates that different structures are enhanced depending on the projection angle, confirming the directional sensitivity and robustness of the approach.



(a) Original image Birds (b) 0° (c) 60° (d) 80° (e) 120° (f) 180° (g) Global edges

Fig. 1: Illustration of Edge Detection in Different Directions Using The proposed DHP Method: (a) Input image, (b–f) Directional edges at 0°, 60°, 80°, 120°, 180°, respectively, highlighting directional sensitivity. (g) Edge detection by projected directional Hessian.

B. Mathematical Analysis of the Proposed Model

Proposition II.1 (Projected Directional Hessian as a Second-Order Derivative)**.** Let $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ be a twice-differentiable image, and let $v \in \mathbb{R}^2$ be a unit direction vector defined by:

$$v = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

Then, the scalar quantity

$$M_v(x, y) := v^\top H(x, y) v$$

represents the second-order derivative of the image intensity in the direction v at the point (x, y), where $H(x, y)$ denotes the Hessian matrix of I at that point. This quantity, referred to as the *projected directional Hessian*, provides a geometrically interpretable measure of local curvature along a direction v, making it a natural indicator for edge or ridge presence in that orientation.

Proof. Consider the directional curve $f(s) := I(x + sv_1, y + sv_2)$, which samples the image intensity along the line passing through (x, y) in the direction v, where $s \in \mathbb{R}$ is the position parameter.

Applying the chain rule yields:

$$f'(s) = \nabla I(x + sv_1, y + sv_2) \cdot v$$

Differentiating once more:

$$f''(s) = \frac{d}{ds} \left(\nabla I(x + sv_1, y + sv_2) \cdot v\right) = v^\top \nabla^2 I(x+sv_1, y+sv_2)\, v$$

Evaluating at $s = 0$, we obtain:

$$f''(0) = v^\top H(x, y) v$$

which confirms that $M_v(x, y)$ is the directional second derivative at (x, y). □

Proposition II.2 (Directional Hessian Projection for Edge Detection)**.** Let $I : \mathbb{R}^2 \to \mathbb{R}$ be a twice-differentiable image

and $H(x, y)$ its Hessian matrix at pixel (x, y). Define a set of K uniformly sampled unit directions over $[0, \pi[$:

$$v_k = \begin{bmatrix} \cos\theta_k \\ \sin\theta_k \end{bmatrix}, \quad \theta_k = \frac{(k-1)\pi}{K}, \quad k = 1, \ldots, K.$$

For each direction v_k, we define the *projected directional Hessian* as:

$$H_{v_k}(x, y) := v_k^\top H(x, y) v_k.$$

which corresponds to the second directional derivative of the image intensity along v_k, and thus captures the local curvature in that direction.

To quantify the edge response, we consider the absolute value:

$$\text{Hessian}_k(x, y) := |H_{v_k}(x, y)|.$$

This ensures invariance to the sign of the curvature, making the method equally sensitive to both convex and concave structures. Specifically:

- $H_{v_k} > 0$: convex variation (ridge-like),
- $H_{v_k} < 0$: concave variation (valley-like),
- $H_{v_k} = 0$: no second-order variation in direction v_k.

Finally, the directional edge strength at pixel (x, y) is obtained by aggregating the directional responses over all directions:

$$\text{EdgeStrength}(x, y) := \frac{1}{K} \sum_{k=1}^{K} |v_k^\top H(x, y) v_k|.$$

This aggregated measure provides a robust estimate of local curvature magnitude, invariant to rotation and contrast inversion, and is well suited to detect edges and curved structures with enhanced stability under noise.

C. Visual Examples with and without Noise

To better understand the behavior of the proposed DHP, we present a set of illustrative examples on both clean and noisy images. These visualizations aim to highlight the robustness of the DHP method to noise, as well as its ability to preserve fine structures.

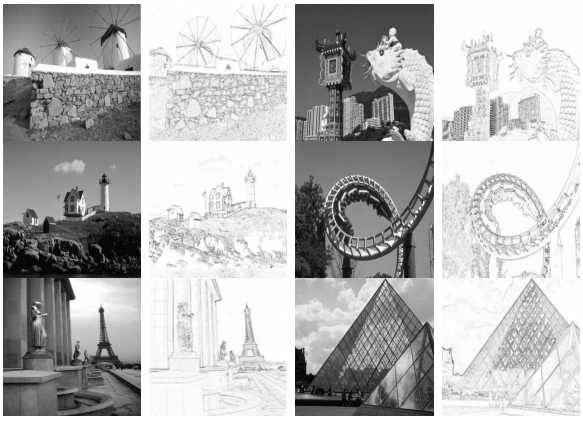
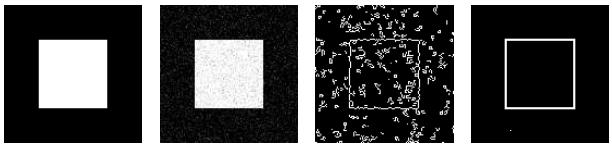


Fig. 2: Illustrative results of edge detection using the proposed DHP method.

As illustrated in Fig. 2, the proposed DHP demonstrates its ability to handle weak edges and second-order discontinuities, providing satisfactory edge detection results on several BSDS500 images [30]. These observations confirm the robustness and relevance of our approach.

Moreover, to evaluate the robustness of the proposed Directional Hessian Projection (DHP) method under noisy conditions, a synthetic image experiment was disturbed with additive Gaussian noise. Fig. 3 presents (a) the original clean image, (b) the noisy image with $\sigma = 0.1$, (c) the edge map produced by the classical Canny method, and (d) the result obtained with DHP combined with Otsu thresholding.



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

Fig. 3: Illustrative results of Robustness of DHP to Gaussian noise: (a) Original synthetic image. (b) Noisy image with $\sigma = 0.1$. (c) Edge map obtained using Canny. (d) Edge map obtained using the proposed DHP method.

Despite incorporating a Gaussian smoothing step to reduce noise, the classical Canny method produces fragmented and cluttered edge maps under additive Gaussian noise, with many spurious detections across the image. In contrast, the proposed DHP method does not include any explicit denoising step, yet it generates clean, continuous edges that accurately outline the square structure despite the noise. This robustness arises from DHP's use of second-order directional derivatives, which capture local curvature information less affected by random intensity fluctuations, combined with local averaging that further stabilizes the response. These results demonstrate that DHP achieves reliable and precise edge detection even under challenging noisy conditions, making it suitable for applications where both accuracy and noise resilience are critical.

To further assess the noise robustness of the proposed DHP method, an extended comparative analysis was conducted against classical edge detectors under increasing levels of Gaussian noise. In this experiment, DHP method was compared to the widely used Sobel and Canny methods. Fig. 4 shows how the mean edge value detected by each method changes with increasing Gaussian noise. At low noise levels, Canny produces the lowest edge values among the three methods. This is because Canny applies non-maximum suppression in its algorithm, which thins edges and retains only local maxima along gradient directions. As a result, even in noise-free images, Canny detects fewer edges compared to Sobel and DHP, which do not include this suppression step. In contrast, Sobel has a higher mean edge value at low noise. As the noise level increases, Sobel's mean edge value rises sharply. This is expected because Sobel treats noise-induced intensity fluctuations as edges, making it highly sensitive to noise. Canny's mean edge value also increases with noise but

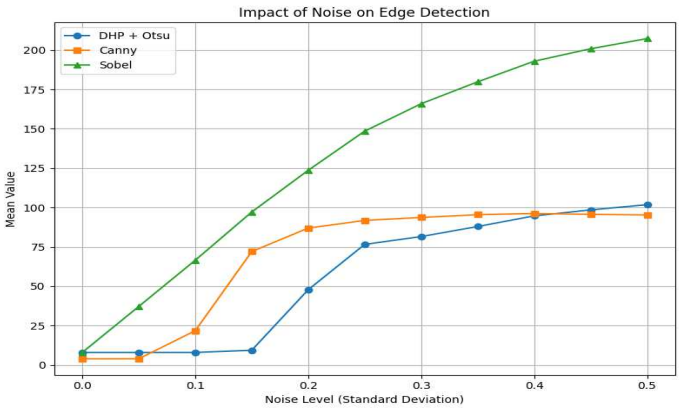


Fig. 4: The impact of Gaussian noise on the performance of three edge detection algorithms, Sobel, Canny, and the proposed DHP.

at a slower rate, as its initial Gaussian smoothing step reduces the effect of small noise variations, and its non-maximum suppression further limits false detections. This explains why Canny remains below Sobel across all noise levels in the figure. In addition, DHP shows a different and more stable behavior. Its mean edge value remains almost constant up to a noise level of around 0.15. Between noise levels of 0.15 and 0.4, DHP detects fewer edges than Canny. This indicates that DHP can suppress noise-induced false edges more effectively, while still preserving true structural edges. However, at very high noise levels (above 0.4), DHP's mean edge value starts to increase, surpassing that of Canny. This can be explained by the fact that, unlike Canny, DHP does not use explicit Gaussian filtering. Instead, it reduces noise effects by analyzing second-order directional derivatives and averaging curvature information over multiple directions. At extreme noise levels, this integration of small variations across directions leads to the detection of more edges, including some noise-induced structures.

Overall, this analysis shows that while Sobel is highly sensitive to noise and Canny provides moderate robustness due to smoothing and suppression, the proposed DHP method maintains strong stability in low to moderate noise conditions. Its ability to capture curvature information makes it less sensitive to random noise while accurately detecting both strong and subtle edges, which is essential for practical applications that require reliable edge detection under varying noise levels.

As expected, Figs. 2, 3, and 4 show that the proposed method effectively addresses the problem of weak edges and second-order discontinuities. It also confirms the robustness of our approach to noise, yielding satisfactory edge detection compared to the Canny and Sobel methods.

D. Discretizations and algorithm

To implement the proposed Directional Hessian Projection (DHP) method in a discrete image setting, we approximate the second-order derivatives using finite differences. This allows for efficient computation of the local Hessian matrix and directional curvature projections. Let $I(x, y)$ be a grayscale image defined over a $2D$ grid. The second-order partial derivatives are approximated by central finite differences as follows:

$$\frac{\partial^2 I}{\partial x^2} \approx I(x+1, y) - 2I(x, y) + I(x-1, y), \tag{2}$$

$$\frac{\partial^2 I}{\partial y^2} \approx I(x, y+1) - 2I(x, y) + I(x, y-1), \tag{3}$$

$$\frac{\partial^2 I}{\partial x \partial y} \approx \frac{1}{4}\left[I(x+1, y+1) - I(x+1, y-1) - I(x-1, y+1) + I(x-1, y-1)\right]. \tag{4}$$

These values form the discrete Hessian matrix:

$$H(x, y) := \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}.$$

The algorithm is described below:

Algorithm 1 Directional Hessian Projection (DHP) Edge Detection

Input: Grayscale image $I(x, y)$.
Output: Edge map $E(x, y)$.

1: Compute the second-order derivatives I_{xx}, I_{yy}, and I_{xy}, as defined in equations (2), (3), and (4), respectively.
2: **C**alculate Local Mean Hessian as defined in equation (1)
3: **Project onto Unit Directions**
4: **for** each v_k, $k = 1, \ldots, K$ **do**

$$v_k = \begin{bmatrix} \cos\theta_k \\ \sin\theta_k \end{bmatrix}, \quad \theta_k = \frac{(k-1)\pi}{K}$$

$$H_{v_k}(x, y) = v_k^\top H_R(x, y) v_k$$

5: **end for**
6: **Aggregate Directional Hessian**

$$H(x, y) = \frac{1}{K} \sum_{k=1}^{K} |H_{v_k}(x, y)|$$

7: **Apply Thresholding**

$$E(x, y) = \{(x, y) \in \Omega \mid H(x, y) > T\}$$

where T is an adaptive or Otsu threshold.
8: **return** $E(x, y)$

III. Results and Discussion

In the following, we display a comprehensive evaluation of the proposed DHP method. We start with the experimental setup, including the datasets, implementation details, and evaluation metrics. We then assess the robustness of DHP to noise and compare its performance against classical and state-of-the-art edge detection methods.

All experiments were conducted using Python 3.10 under Windows 11. Moreover, the datasets that used for our evaluation :

- **Standard images:** To illustrate qualitative performance and visual interpretability, we included classical images such as *White square*,*Windmill*, *Lena*, and *Oranges*[31] *Red tulip*, *Swan*[32],and *Butterfly*[33].

- **BSDS500:** A widely used benchmark dataset for edge detection and segmentation[30], containing 500 natural images annotated with ground truth contours by multiple human subjects. Our experiments used the standard test subset of 200 images for quantitative evaluation.
- **Multicue:** originally designed to study human boundary perception. Unlike BSDS500's static images, Multicue consists of 100 short stereo video sequences recorded in complex real-world environments. The final frame of each left-view sequence is annotated with both object boundaries and low-level edges. It provides 80 training and 20 testing samples, with results averaged over three runs.

The DHP method was implemented by computing second-order derivatives to construct the local Hessian matrix. The number of directions K was set to 10, uniformly sampled over $[0, \pi[$, and the neighborhood window size w was set to 3, balancing computational speed and local curvature estimation accuracy.

We applied Otsu's adaptive thresholding for post-processing to binarize the aggregated edge map.

A. Evaluation Metrics

To comprehensively evaluate the performance of the proposed DHP method against ground truth annotations, we computed six standard metrics commonly used in edge detection evaluation, following the definitions from Li and Xu (2025) [34]:

- **Mean Squared Error (MSE):**

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2,$$

  where M and N are image dimensions, I is the ground truth, and K is the predicted edge map.
- **Peak Signal-to-Noise Ratio (PSNR):**

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right),$$

  where MAX is the maximum possible pixel value (1.0 for normalized images).
- **Accuracy:**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

  where TP is true positives, TN true negatives, FP false positives, and FN false negatives. This metric measures the proportion of correctly classified pixels.
- **Average Precision (AP):**

$$AP = \frac{TP}{TP + FP},$$

  where TP and FP are as defined above. This metric evaluates the precision of edge detection results.
- **Recall:** Measures how many actual positive instances are correctly identified:

$$Recall = \frac{TP}{TP + FN},$$

  where FN is the number of false negatives.
- **F1-Score:** Represents the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

  It ranges from 0 to 1, with higher values indicating better balance between precision and recall.
- **Optimal Dataset Scale (ODS)** and **Optimal Image Scale (OIS):** These F-scores are commonly used in edge detection to evaluate performance across thresholds.
  - **ODS:** Measures the best F-score over the entire dataset using a single optimal threshold:

$$ODS = \max_{\text{threshold}} \left(\frac{2 \times Precision \times Recall}{Precision + Recall} \right).$$

    It reflects the model's global performance consistency.
  - **OIS:** Computes the best F-score for each image individually, then averages the results:

$$OIS = \frac{1}{N} \sum_{i=1}^{N} \max_{\text{threshold}_i} \left(\frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \right),$$

    where N is the total number of images. It highlights the model's ability to adapt to different image characteristics.

These metrics provide a comprehensive assessment of edge detection performance, including pixel-wise errors (MSE, PSNR), classification performance (Accuracy), and precision (AP), consistent with previous studies in edge detection benchmarking [34].

To evaluate the performance of the proposed DHP method, we conducted a comparative study using four standard test images taken from [31], namely: a white square on a black background (Fig. 5), Windmill (Fig. 6), Lena (Fig. 7), and Oranges (Fig. 8). Each image was processed using several classic edge detectors (Roberts, Prewitt, Sobel, Marr-Hildreth (Gaussian and LoG), and Haralick), as well as our proposed DHP method.

Fig. 5 compares several edge detection methods on a simple square image. The first-order methods (Roberts, Prewitt, Sobel) correctly detect the contour, with Roberts producing thinner edges and Prewitt/Sobel giving slightly thicker but more stable ones. Second-order methods introduce more distortions: Marr-Hildreth rounds the corners due to Gaussian smoothing, while Haralick produces thicker edges. In contrast, the proposed **DHP** method detects clean and well-localized edges, preserving both straight boundaries and sharp corners.

Fig. 6 shows the edge detection results on the Windmill image, which contains complex textures and shaded regions. The **first-order methods** (Roberts, Prewitt, Sobel) detect the main edges but produce fragmented and incomplete contours, missing some structures such as the lower blade. The **second-order methods** (Marr-Hildreth and Haralick) detect slightly more details but still generate irregular or overly smoothed edges. In contrast, the **proposed DHP method** produces cleaner, smoother, and more continuous edges, successfully

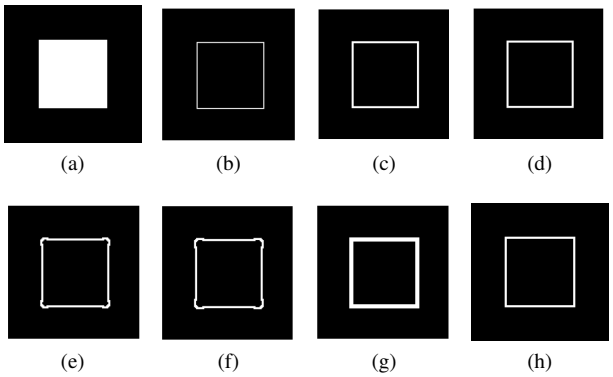


Fig. 5: Edge detection results on the white square image using different methods: (a) Original image, (b) Roberts, (c) Prewitt, (d) Sobel, (e) Marr-Hildreth (Gaussian), (f) Marr-Hildreth (LoG), (g) Haralick, and (h) Proposed DHP.

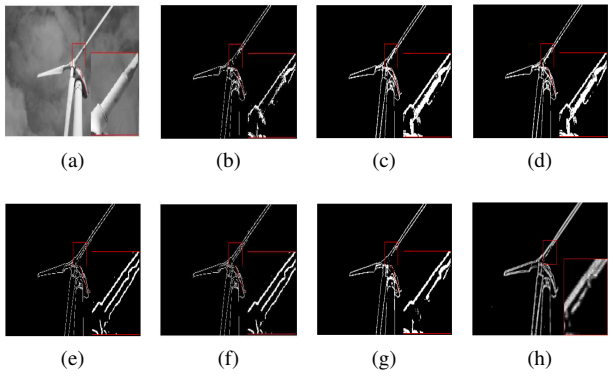


Fig. 6: Edge detection results on the Windmill image using different methods: (a) Original image, (b) Roberts, (c) Prewitt, (d) Sobel, (e) Marr-Hildreth (Gaussian), (f) Marr-Hildreth (LoG), (g) Haralick, and (h) Proposed DHP.

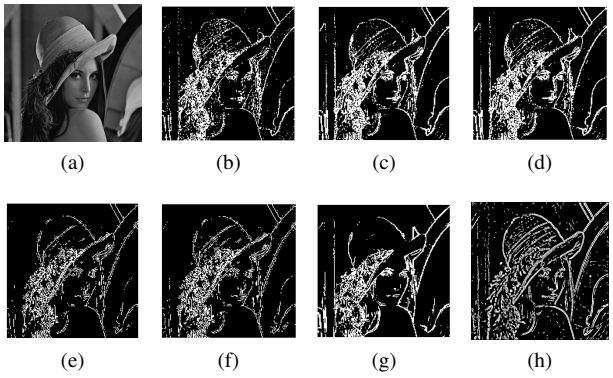


Fig. 7: Edge detection results on the Lena image using different methods: (a) Original image, (b) Roberts, (c) Prewitt, (d) Sobel, (e) Marr-Hildreth (Gaussian), (f) Marr-Hildreth (LoG), (g) Haralick, and (h) Proposed DHP.

detecting the shaded blade and preserving the overall structure of the windmill.

Fig. 7 shows the edge detection results on the Lena image,

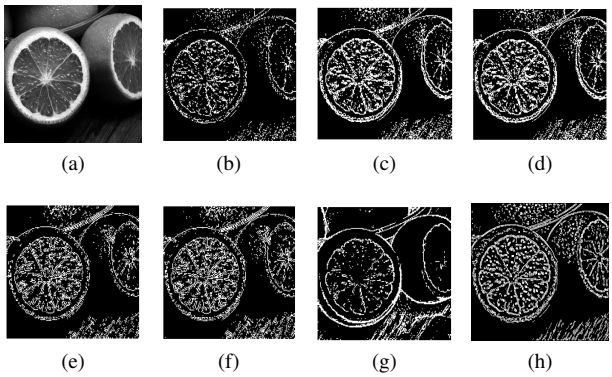


Fig. 8: Edge detection results on the Oranges image using different methods: (a) Original image, (b) Roberts, (c) Prewitt, (d) Sobel, (e) Marr-Hildreth (Gaussian), (f) Marr-Hildreth (LoG), (g) Haralick, and (h) Proposed DHP.

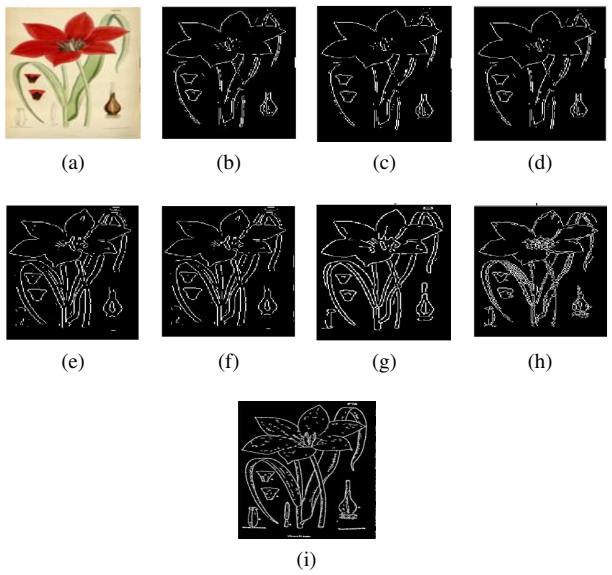


Fig. 9: Comparison of edge detection results on the red tulip image: (a) Original image, (b) Sobel, (c) Prewitt, (d) Roberts, (e) LOG, (f) Zero-Crossing, (g) Canny, (h) Proposed method by Yahya et al, and (i) Proposed DHP method.

which contains smooth textures, sharp edges, and complex details. The **first-order methods** (Roberts, Prewitt, Sobel) detect the main contours but produce fragmented edges and many spurious points, especially in textured regions. The **second-order methods** (Marr-Hildreth and Haralick) generate more recognizable contours and reduce noise, but edges remain either irregular or overly thick. In contrast, the **proposed DHP method** provides clearer, more continuous, and well-localized edges, preserving important facial features and structures while reducing false detections in the background.

Fig. 8 shows the edge detection results on the *Oranges* image, which contains complex textures, shadows, and rounded contours. The **first-order methods** (Roberts, Prewitt, Sobel) detect some fruit structures but produce many

spurious points and fragmented edges. The **second-order methods** (Marr-Hildreth and Haralick) generate smoother contours and reduce noise, but some edges remain broken or overly smoothed, causing loss of fine details. In contrast, the **proposed DHP method** produces cleaner and more consistent edges, accurately capturing both internal and external fruit contours while remaining robust to texture and illumination variations.

After evaluating the DHP method against classical edge detectors on standard test images from Spontón and Cardelino [31], we extended our comparative study to a second benchmark, proposed in the work by Yahya et al. [32]. This study focuses on natural color images drawn from diverse domains including flowers, animals, and textured backgrounds which pose additional challenges such as low-contrast transitions, curved structures, and fine textures.

Figs. 9 and 10 present qualitative comparisons on two natural images: *red tulip* and *swan*. The results of several classical edge detectors (Sobel, Prewitt, Roberts, LoG, Zero-Crossing, Canny, and the method of Yahya et al. [32]) are compared with the proposed **DHP** method to evaluate robustness and structural accuracy in complex scenes. Indeed, in the tulip image, first-order methods produce fragmented and incomplete edges, missing several fine structures. Second-order methods slightly improve detection but still generate irregular contours. Canny and the method of Yahya et al. provide more continuous edges, yet some details remain poorly detected or oversimplified. In contrast, the proposed **DHP** method detects both major and fine structures with sharp, continuous, and geometrically faithful contours. Moreover, in the swan image, classical detectors identify the general silhouette but produce thin and discontinuous edges, especially around the neck and body. While LoG, Zero-Crossing, Canny, and the method of Yahya et al. improve contour visibility, the results remain partially fragmented and some details are lost. The proposed **DHP** method clearly delineates both the swan and its reflection with sharp and continuous edges, preserving the geometry and fine structures of the scene.

Finally, to further evaluate and validate the performance of the proposed model, we tested it on another dataset and compared its results with those of other existing methods. To ensure fairness, we directly use the values reported in their published works and present the results at a single scale. Table I presents the comparison with other models on the Multicue dataset. Fig 11 presents several examples of edge detection results. Each pair shows the original image (top) and the corresponding edges detected by the proposed method (bottom). The visual results clearly demonstrate the model's ability to capture fine and complex image discontinuities with high precision, producing detailed and meaningful edge maps that accurately outline object boundaries and structural details.

Table I compares the performance of different edge detection methods on the Multicue dataset. The proposed DHP method achieves an ODS of 0.901 and an OIS of 0.93. With a precision of 0.85 and a recall of 0.96, the model shows a strong ability to detect most true edges while maintaining good accuracy. This high recall indicates that DHP successfully identifies almost all relevant boundaries, even in challenging scenes, while the moderate precision suggests that some detected edges may not exist in the ground truth.

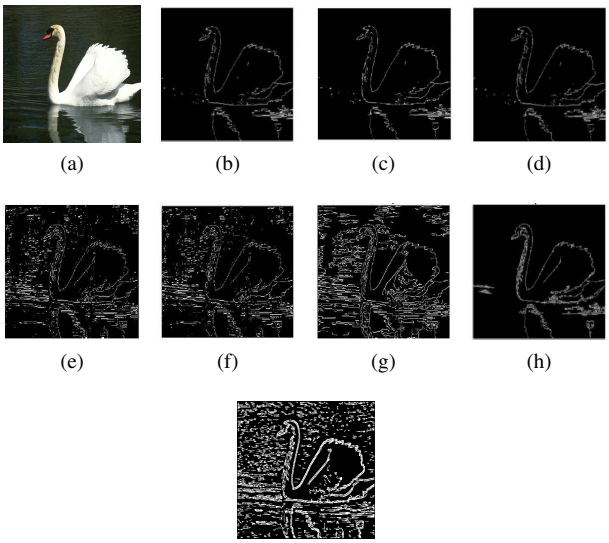


Fig. 10: Comparison of edge detection results on the swan image: (a) Original image, (b) Sobel, (c) Prewitt, (d) Roberts, (e) LOG, (f) Zero-Crossing, (g) Canny, (h) Proposed method by Yahya et al, and (i) Proposed DHP method.



Fig. 11: Results of edge detection using the proposed DHP method on the Multicue dataset.

Methods	ODS	OIS
Human [35]	0.75	-
Multicue [35]	0.83	-
HED [36]	0.851	0.864
RCF [37]	0.857	0.862
BDCN [38]	0.891	0.898
PiDiNet [39]	0.855	0.860
EDTER [40]	0.894	0.900
UAED [41]	0.895	0.902
MuGE [42]	0.875	0.879
UEDVE [43]	0.899	0.934
DHP (proposed)	**0.901**	**0.93**

TABLE I: Results on the Multicue Dataset.

B. Noise Robustness Evaluation

In order to confirm the efficiency and robustness of the proposed model, even in the presence of noise, we conducted experiments at various noise levels. We also compared our results with those previously published in [34]. Compared to the methods presented in this article, our approach follows a different strategy. Most of the published techniques rely on pre-processing steps, such as wavelet denoising and Gaussian filtering, to reduce noise and improve edge quality before detection. In contrast, our method does not require any pre-processing. It directly processes both clean and noisy images by analyzing the projected directional Hessian. This allows us to preserve fine details that might be lost during filtering. Overall, our approach is simpler, does not need parameter tuning, and is able to detect subtle edges in complex or noisy images.

This evaluation was performed under similar experimental conditions, using the Butterfly image [33] with varying levels of Gaussian noise (0.1%, 10%, 20%, and 30%) and the BSDS500 dataset[30]. Table II summarizes the performance metrics obtained for both methods on the Butterfly image. In the original EDAW experiments, three thresholding strategies were tested (soft, hard, and median), yielding comparable results with an accuracy of approximately 63.84%, an MSE of 0.3619 Pixels, and a PSNR of around 4.41 dB. In contrast, the proposed DHP method, using Otsu thresholding, achieved a significantly lower Mean Squared Error (MSE) of 0.2364, a higher PSNR of 6.26 dB, and an improved accuracy of 76.36%. This indicates that DHP not only provides sharper and more precise edge maps but also demonstrates superior robustness in capturing structural details. By checking these values in Table II, the proposed DHP method maintains high accuracy and structural coherence even under increasing noise conditions, confirming its robustness and suitability for real-world edge detection tasks in noisy environments.

Method	MSE	PSNR (dB)	Accuracy (%)
EDAW (Soft Threshold)	0.3619	4.4136	63.80
EDAW (Hard Threshold)	0.3615	4.4189	63.84
EDAW (Median Threshold)	0.3615	4.4189	63.84
DHP (Otsu Threshold)	**0.2364**	**6.2638**	**76.36**

TABLE II: Comparison of edge detection performance between EDAW [34] and the proposed DHP method on the Butterfly image.

These results demonstrate the strength of the DHP approach in capturing structural details more precisely and consistently. The lower MSE and higher PSNR imply that DHP generates edge maps with fewer errors and clearer contours, while the improved accuracy confirms its ability to detect relevant edges effectively. Overall, the comparison indicates that DHP offers superior performance to EDAW on this image, highlighting its potential as a robust and efficient method for edge detection tasks requiring high precision and structural fidelity.

A qualitative assessment was conducted under increasing levels of additive Gaussian noise (0.1%, 10%, 20%, and 30%) using the Butterfly image. The proposed *Directional Hessian Projection (DHP)* method was visually compared to the *Canny* detector and various wavelet-based methods integrated into the EDAW framework [34].

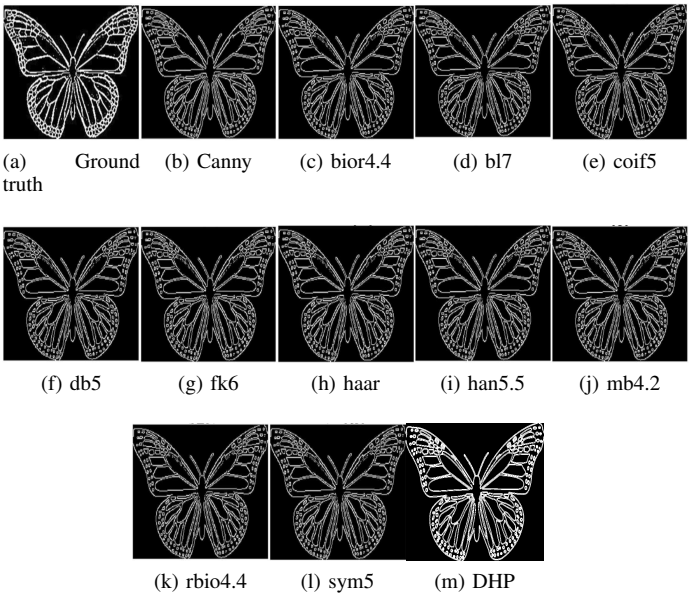


Fig. 12: Edge detection results of different denoising combinations at 0.1% Gaussian noise level from EDAW, extended with the proposed DHP method.

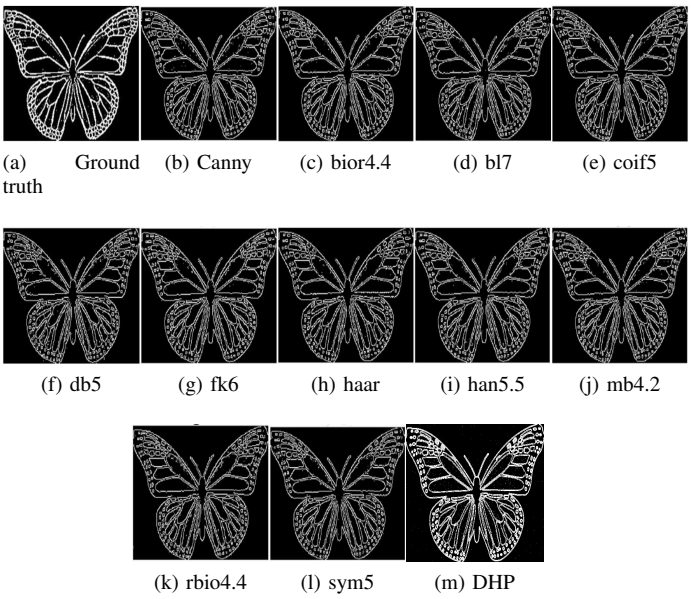


Fig. 13: Edge detection results of different denoising combinations at 10% Gaussian noise level from EDAW, extended with the proposed DHP method.

Figs. 12, 13, 14, and 15, show a visual comparison of edge detection results under different noise levels (0.1%, 10%, 20%, and 30%) using various methods, including the proposed one. Tables III, IV, V, and VI, present quantitative results using four metrics: MSE, PSNR, edge detection Accuracy, and Average Precision (AP) scores.

Visually, the proposed method stands out by producing

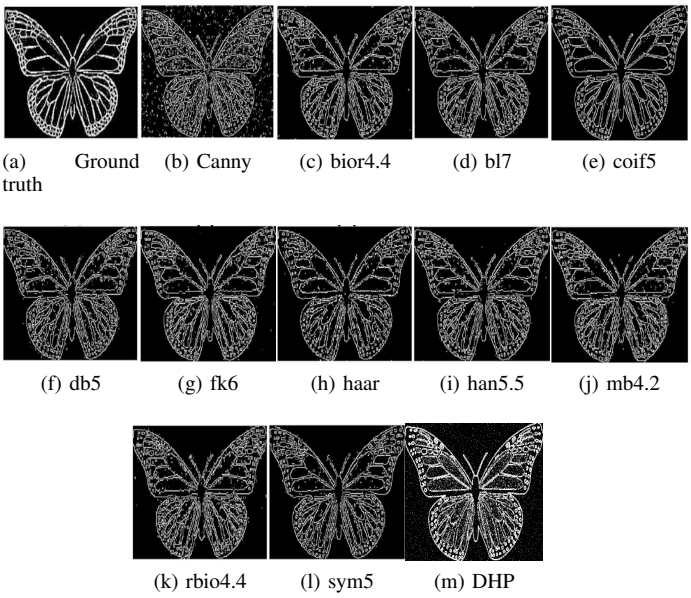


Fig. 14: Edge detection results of different denoising combinations at 20% Gaussian noise level from EDAW, extended with the proposed DHP method.

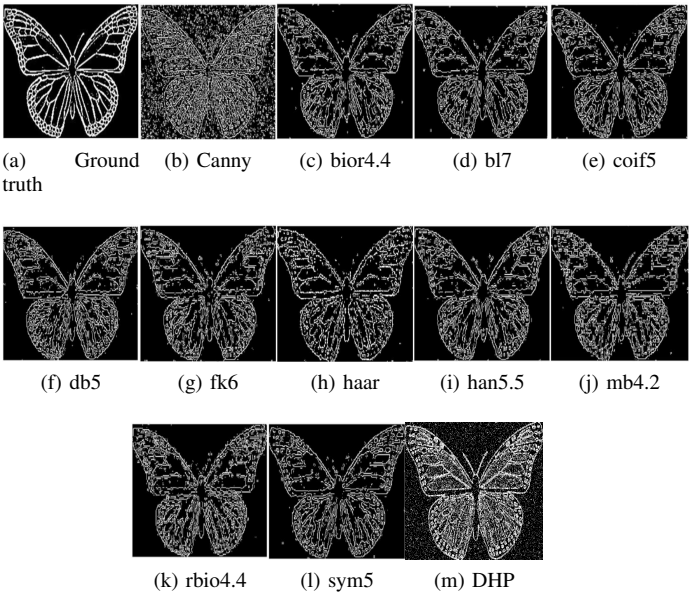


Fig. 15: Edge detection results of different denoising combinations at 30% Gaussian noise level from EDAW, extended with the proposed DHP method.

clear, continuous, and accurate edges, even when the noise level is high (up to 30%). While other methods such as Canny and wavelet-based approaches (like bior4.4, bl7, coif5, etc.) either lose fine details or produce noisy contours, the proposed method preserves the full structure of the butterfly, including the fine wing patterns.

Table III presents the Mean Squared Error (MSE) values for various edge detection methods under four different levels of additive Gaussian noise: 0.1%, 10%, 20%, and 30%. This comparison evaluates the noise robustness of traditional edge detectors Canny, wavelet-based techniques from the EDAW framework, and the proposed Directional Hessian Projection (DHP) method.

At a low noise level of **0.1%**, most wavelet-based methods perform similarly, with MSEs clustering tightly around 0.3615. Canny shows a slightly higher error (0.3641), indicating some early sensitivity to noise. In contrast, *DHP* significantly outperforms all methods, achieving an MSE of 0.2452, indicating sharper and more accurate edge preservation even with light perturbations.

At **10% noise**, performance variation becomes more visible. While wavelet methods begin to show subtle differences (e.g., mb4.2 and fk6 drop below 0.3600), Canny's error continues to increase slightly (0.3655). *DHP* again stands out with a low MSE of 0.2457, maintaining remarkable stability despite rising noise.

At **20% noise**, the degradation in wavelet performance becomes more pronounced. Most filters report higher MSEs (typically around 0.35–0.36), with only mb4.2 and rbio4.4 maintaining relatively lower error (0.3484 and 0.3520, respectively). Canny performs poorly (0.3756), indicating substantial breakdown. However, *DHP* preserves structure effectively with an MSE of 0.2765, confirming its noise resilience.

At the highest noise level of **30%**, nearly all methods degrade. Canny suffers the most, reaching an MSE of 0.4219, while wavelet approaches range from 0.3476 (mb4.2) to 0.3607 (sym5). Even under these severe conditions, *DHP* maintains a relatively low MSE of 0.3137, outperforming all competitors. Across all noise levels, the DHP method consistently delivers the lowest MSE values. Unlike Canny and many wavelet-based approaches, whose performance deteriorates significantly as noise increases, *DHP* maintains structural fidelity and noise robustness. This quantitative evaluation reinforces the DHP method's suitability for edge detection tasks in real-world noisy conditions.

Table IV presents the PSNR scores for different edge detection methods under four levels of Gaussian noise (0.1%, 10%, 20%, and 30%) on the Butterfly image. At low noise (0.1%), most wavelet-based methods give similar PSNR values around 4.41–4.42 dB, while Canny shows the lowest value. The proposed **DHP** method clearly outperforms the others with a PSNR of 6.1040 dB. As the noise increases (10%, 20%, and 30%), traditional and wavelet-based methods remain around 4–4.6 dB but gradually degrade. In contrast, **DHP** keeps the highest PSNR values (from 6.0952 dB to 5.0354 dB), showing better robustness to noise and better preservation of edge structures.

Table V presents the accuracy rates of several edge detection methods under increasing Gaussian noise levels (0.1%, 10%, 20%, and 30%). Most wavelet-based filters and Canny obtain similar accuracy values around 63.5%–65.2%, with slight variations as noise increases. In contrast, the proposed **DHP** method consistently achieves the highest accuracy, reaching 75.48% at low noise and remaining clearly above the other methods even at higher noise levels (75.42%, 72.35%, and 68.63%). These results highlight the strong robustness of DHP in preserving meaningful edge structures under noisy

conditions.

Method	0.1%	10%	20%	30%
Canny	0.3641	0.3655	0.3756	0.4219
bior4.4	0.3615	0.3612	0.3570	0.3580
bl7	0.3617	0.3605	0.3586	0.3564
coif5	0.3616	0.3619	0.3603	0.3599
db5	0.3617	0.3612	0.3592	0.3563
fk6	0.3617	0.3594	0.3534	0.3532
haar	0.3615	0.3620	0.3619	0.3584
han5.5	0.3616	0.3611	0.3570	0.3560
mb4.2	0.3618	0.3597	0.3484	0.3476
rbio4.4	0.3614	0.3599	0.3520	0.3497
sym5	0.3620	0.3636	0.3607	0.3607
DHP (Proposed)	**0.2452**	**0.2457**	**0.2765**	**0.3137**

TABLE III: MSE values of edge detection methods at different Gaussian noise levels

Method	0.1%	10%	20%	30%
Canny	4.3878	4.4186	4.2071	3.7485
bior4.4	4.4189	4.4227	4.4735	4.4610
bl7	4.4170	4.4310	4.4542	4.4802
coif5	4.4172	4.4139	4.4337	4.4384
db5	4.4170	4.4229	4.4466	4.4822
fk6	4.4170	4.4439	4.5176	4.5200
haar	4.4184	4.4134	4.4136	4.4564
han5.5	4.4174	4.4234	4.4732	4.4855
mb4.2	4.4158	4.4411	4.5791	4.5890
rbio4.4	4.4201	4.4384	4.5346	4.5633
sym5	4.4134	4.3937	4.4286	4.4286
DHP (proposed)	**6.1040**	**6.0952**	**5.5837**	**5.0354**

TABLE IV: PSNR (dB) comparison of edge detection methods under varying Gaussian noise levels.

Method	0.1%	10%	20%	30%
Canny	63.5901	63.5076	62.8429	62.8153
bior4.4	63.8496	63.8813	64.3013	64.1983
bl7	63.8337	63.9506	64.1428	64.3567
coif5	63.8357	63.8080	63.9724	64.0120
db5	63.8337	63.8833	64.0794	64.3726
fk6	63.8337	64.0576	64.6618	64.6816
haar	63.8456	63.8040	63.8060	64.1606
han5.5	63.8377	63.8872	64.2993	64.4003
mb4.2	63.8238	64.0338	65.1591	65.2383
rbio4.4	63.8595	64.0120	64.8005	65.0323
sym5	63.8040	63.6396	63.9308	63.9308
DHP (Proposed)	**75.4755**	**75.4255**	**72.3541**	**68.6338**

TABLE V: Accuracy (%) of different edge detection methods under Gaussian noise at various levels.

Comparison Against State-of-the-Art Methods: To evaluate the performance of the proposed *Directional Hessian Projection (DHP)* approach in a broader context, we compare it with several well-known edge detection techniques using the BSDS500 test dataset [30]. These include traditional methods (Canny, Sobel, Roberts), advanced algorithms (SLEDGE), and learning-based approaches (RCF and the method by Li et al.). Results are reported in terms of Average Precision (AP), a widely adopted metric for evaluating edge detection quality.

Fig. 16 provides visual examples comparing edge maps from different detectors, and Table VI summarizes the AP values.

The DHP method achieves a competitive AP score of **0.7717**, clearly surpassing all traditional detectors. In particular, Canny achieves 0.580, Sobel 0.568, and Roberts 0.512. Compared to

Method	AP Score
Canny	0.580
Sobel	0.568
Roberts	0.512
SLEDGE	0.652
RCF	0.822
Li et al.	0.605
EDAW	0.682
Proposed DHP	**0.7717**

TABLE VI: Average Precision (AP) scores for various edge detection methods on the BSDS500 test set.

EDAW, which achieves an AP of 0.682, DHP demonstrates a substantial improvement of nearly 9 percentage points. Even when compared to SLEDGE (0.652), a high-performing traditional algorithm, DHP maintains a significant lead.

While deep learning-based RCF achieves the highest AP of 0.822, DHP's performance narrows the gap without relying on data-driven training, thus avoiding the overhead of model training and dataset dependency. This makes DHP a highly practical alternative for tasks where training data is scarce or domain adaptation is costly.

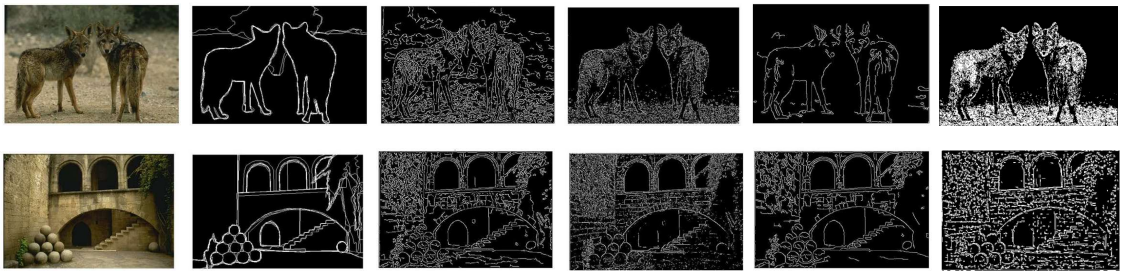
The superior performance of DHP stems from two core innovations. First, the directional second-order projection framework integrates gradient information over multiple orientations, which enhances structural sensitivity and suppresses noise. Second, the combination with simple global thresholding ensures edge maps are not only accurate but also consistent across different scenes. The algorithm's unsupervised nature, stability under noise, and lack of hyperparameter tuning contribute to its strong generalization.

In contrast, traditional detectors struggle with noisy textures and fragmented contours. As observed in visual results (Fig. 16), Canny and Roberts tend to over-detect noise while missing thin or low-contrast boundaries. EDAW, though more robust, still suffers from edge attenuation in cluttered regions. DHP, on the other hand, retains fine details and yields crisp and coherent edge structures.

These results highlight that DHP effectively balances precision, robustness, and practicality, offering a strong alternative to both classical and learning-based edge detection pipelines in real-world conditions.

IV. Conclusion

In this study, we introduced the Directional Hessian Projection (DHP), a novel edge detection technique designed for high accuracy and robustness across a wide spectrum of image types and noise conditions. The proposed method was evaluated on both synthetic and real-world datasets, including the White Square image, the Windmill and Butterfly images, as well as the BSDS500 benchmark and the MultiCue dataset. Through comparative experiments, DHP demonstrated superior performance against traditional edge detectors (such as Canny, Roberts, Sobel, and Prewitt) and modern approaches. In synthetic tests, DHP produced more continuous and structurally faithful edges and remained robust under Gaussian noise, as confirmed both visually and through numerical metrics (MSE, PSNR, and Accuracy, ...). On BSDS500, DHP achieved an Average Precision (AP) of 0.7717, outperforming classical



(a) Original image (b) Ground truth (c) Canny (d) Roberts (e) EDAW (f) DHP

Fig. 16: Visual comparisons of edge detection results on multiple images using classical methods and EDAW. New method performs the best.

baselines such as Roberts (0.512), Sobel (0.568), and EDAW (0.682). On the MultiCue dataset, it achieved an ODS of 0.901 and an OIS of 0.93, with a precision of 0.85 and a recall of 0.96, confirming its robustness and generalization ability. These results highlight DHP as an accurate, robust, and computationally simple edge detection method, suitable for integration into higher-level vision tasks such as segmentation, object detection, and pattern recognition. Future work will focus on improving computational efficiency and extending the approach to 3D image analysis.

REFERENCES

[1] N. Hafeez. (2023) The art and science of edge detection. Consulté le 27 mai 2025 sur Medium. [Online]. Available: https://medium.com/@nerminhafeez2002/the-art-and-science-of-edge-detection-6cf4b6b69ac4

[2] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[3] M. Delbracio and G. Sapiro, "Hand-held video deblurring via efficient fourier aggregation," *IEEE Transactions on Computational Imaging*, vol. 1, no. 4, pp. 270–283, 2015.

[4] M. C. Shin, D. B. Goldgof, and K. W. Bowyer, "Comparison of edge detector performance through use in an object recognition task," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 160–178, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314201909324

[5] X.-F. Wang, D.-S. Huang, and H. Xu, "An efficient local chan–vese model for image segmentation," *Pattern Recognition*, vol. 43, no. 3, pp. 603–618, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320309002982

[6] W. Luo, "Comparison for edge detection of colony images," *IJCSNS International Journal of Computer Science and Network Security*, vol. 6, no. 9A, pp. 211–215, 2006.

[7] A. BenHajyoussef and A. Saidani, "Recent advances on image edge detection," *Digital Image Processing-Latest Advances and Applications: Latest Advances and Applications*, p. 43, 2024.

[8] S. Seyyedi, M. Wieczorek, F. Pfeiffer, and T. Lasser, "Incorporating a noise reduction technique into x-ray tensor tomography," *IEEE Transactions on Computational Imaging*, vol. 4, no. 1, pp. 137–146, 2018.

[9] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[10] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," *International journal of computer science and management Research*, vol. 2, no. 2, pp. 1578–1583, 2013.

[11] G. E. Sotak Jr and K. L. Boyer, "The laplacian-of-gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output," *Computer vision, graphics, and image processing*, vol. 48, no. 2, pp. 147–189, 1989.

[12] T. F. Chan, J. Shen, and L. Vese, "Variational PDE models in image processing," *Notices AMS*, vol. 50, no. 1, pp. 14–26, 2003.

[13] V. Torre and T. A. Poggio, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 147–163, 1986.

[14] M. Lajili, "Modèles avancés pour le recalage d'images multimodales," Ph.D. dissertation, Université de Haute Alsace-Mulhouse & École Nationale d'Ingénieurs de Tunis, 2023.

[15] D. B. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, 1989.

[16] M. Zanetti and A. Vitti, "The blake-zisserman model for digital surface models segmentation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 355–360, 2013.

[17] L. J. Belaid, M. Jaoua, M. Masmoudi, and L. Siala, "Application of the topological gradient to image restoration and edge detection," *Engineering Analysis with Boundary Elements*, vol. 32, no. 11, pp. 891–899, 2008.

[18] M. Lajili, B. Rjaibi, D. Auroux, and M. Moakher, "Edge detection from X-ray tomographic data for geometric image registration," *Mathematical Methods in the Applied Sciences*, 2022.

[19] Q. Huang and J. Huang, "Comprehensive review of edge and contour detection: from traditional methods to recent advances," *Neural Computing and Applications*, pp. 1–35, 2025.

[20] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," in *Advances in neural information processing systems*, vol. 17, 2004, pp. 729–736.

[21] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[22] J. J. Lim, P. Arbelaez, A. Torralba, and J. Malik, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3158–3165.

[23] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, and G. Wang, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] M. Lajili, Z. Belhachmi, M. Moakher, and A. Theljani, "Unsupervised deep learning for geometric feature detection and multilevel-multimodal image registration," *Applied Intelligence*, pp. 1–19, 2024.

[26] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

[27] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5872–5881.

[28] Q. He, R. Zhang, W. Lin, and Y. Yang, "Bi-directional cascade network for perceptual edge detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3828–3837.

[29] X. R. Poma, E. Riba, and A. D. Sappa, "Dense extreme inception network: Towards a robust cnn model for edge detection," in *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2020, pp. 1923–1932.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 2. Ieee, 2001, pp. 416–423.

[31] H. Spontón and J. Cardelino, "A review of classic edge detectors," *Image Processing On Line*, vol. 5, pp. 90–123, 2015.

[32] S. Yahya, H. Elsanary, M. Hassan, and A. Ali, "An improved edge detection method for image analysis in diverse domains," *Aswan Science and Technology Bulletin*, pp. 1–19, 2024.

[33] G. Piosenka, "Butterfly and moths image classification – 100 species," https://www.kaggle.com/datasets/gpiosenka/butterfly-images40-species/data, 2022, accessed: 10/05/2025.

[34] H. Li and K. Xu, "Innovative adaptive edge detection for noisy images using wavelet and gaussian method," *Scientific Reports*, vol. 15, no. 1, p. 5838, 2025.

[35] D. A. M´ely, J. Kim, M. McGill, Y. Guo, and T. Serre, "A systematic comparison between visual cues for boundary detection," *Vision Research*, vol. 120, pp. 93–107, 2016.

[36] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

[37] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang, "Richer convolutional features for edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1939–1946, 2019.

[38] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-directional cascade network for perceptual edge detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3828–3837.

[39] Z. Su, W. Liu, Z. Yu, D. Hu, and L. Liu, "Pixel difference networks for efficient edge detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5117–5127.

[40] M. Pu, Y. Huang, Y. Liu, Q. Guan, and H. Ling, "Edter: Edge detection with transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1402–1412.

[41] C. Zhou, Y. Huang, M. Pu, Q. Guan, L. Huang, and H. Ling, "The treasure beneath multiple annotations: An uncertainty-aware edge detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 507–15 517.

[42] C. Zhou, Y. Huang, M. Pu, Q. Guan, R. Deng, and H. Ling, "Muge: Multiple granularity edge detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 25 952–25 962.

[43] M. Lajili and M. Fall, "Unsupervised edge detection with a variational energy-driven deep network," *The Visual Computer*, pp. 1–18, 2025.