SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I8P104

Volume 13 Issue 8, 35-46, August 2026

Original Article

Analysis of Edge Detectors based on Performance Evaluation Metrics

Wazir[1], Rajeshwar Dass[2]

[1,2]*Department of Electronics and Communication Engineering, Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Murthal, Haryana, India.*

[1]*Corresponding Author : lakrawazir@gmail.com*

Received: 26 May 2026 Revised: 23 June 2026 Accepted: 24 July 2026 Published: 25 August 2026

Abstract *- A joint comparison of the outcomes of edge detection techniques provides a more thorough knowledge of the capabilities of the algorithms to identify image edges. Edge detection methods are classified as conventional edge detection, soft computing edge detection and deep learning edge detection strategies. Detection efficiency is evaluated by means of a comprehensive analysis that includes F1-score and Intersection over Union (IoU). Deep learning-based models have proven to be more efficient with regard to F1-scores due to better feature extraction. High PSNR indicates good agreement between detected boundaries and actual objects. In the course of conducted experiments, Clip-MobileNetV2-Unet has shown better results concerning Recall, Precision, F1-Score/mDC, Accuracy, PSNR (dB), and mIoU. All these metrics make up a complex evaluation strategy. From this study, it is found that deep learning algorithms perform better in terms of accuracy and reliability compared to other approaches, while traditional detectors are fast and suitable for simple real-time purposes.*

Keywords *- Edge detectors, Performance evaluation metrics, Machine Learning, Deep Learning, Soft Computing.*

1. Introduction

The concept of edge detection is based on the observation that the edges correspond to significant changes in intensity values [1-3]. These changes can be caused by sudden changes in depth, orientation, reflectance, or illumination [4-7]. Detecting these sudden changes helps to detect important features such as corners, edges, and lines [8, 9]. Over the years, many different methods for detecting edges have been proposed, and they can be broadly categorised as conventional, soft computing, and learning-based approaches [10-13]. Traditional edge detectors, such as Sobel, Prewitt, and Canny, rely on gradient-based or second-order derivative approaches to identify edges [1-13].

The techniques have shown computational efficiency and easy implementation, making them suitable for real-time applications [5-10]. However, typical algorithms often struggle to cope in the case of a noisy environment, varying illumination levels, and complicated textures. To tackle these problems, soft computing techniques like Ant Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO) have been proposed [9-12]. Such strategies allow for improving the performance of existing edge detectors through optimisation of parameters, i.e., thresholds, and therefore increase their detection precision and robustness. Even though such strategies can increase the adaptability of edge detectors, they can make them more computationally costly. Over recent years, machine learning and deep learning-based edge detection techniques have been widely researched [13-28]. CNNs, U-Net models, and their combinations can learn hierarchical features and patterns from input data [29-44]. They show high performance in terms of fine edge detection and coping with noise [33-38]

Different performance measures like precision, recall, F1-measure, and PSNR are utilized for evaluating different aspects of edge detection [14-25]. These measures highlight one specific property, be it accuracy, similarity, or localization. Overall, edge detection continues to be an important field that is constantly evolving [26-31]. With improvements in advanced techniques and the use of extensive evaluation metrics, edge detection algorithms have become increasingly accurate and reliable, making them applicable in various fields, from medicine to robotics [32-37].

The research gap in this study is that traditional edge detectors are simple and easy to use, but fail at complex images. Traditional + optimization methods improve the performance of traditional edge detectors, but they are slow and require large computation and memory. Deep learning models also require a large amount of data, huge memory and high performance environment for training, but they outperform the traditional and traditional + optimization methods. For traditional and traditional + optimization methods, F1 score, accuracy, recall, etc., are widely used, but for deep learning models, similarity index metrics mDC and IoU are used to evaluate their performance.

The problem with edge detection is that all edge detectors are application-oriented; none is universal. They behave differently under different conditions, and their performance varies across different images. Similarly, evaluation metrics are also application-specific. For example, precision measures the reliability of the model, recall measures the completeness of the model, and the F-1 score measures the effectiveness of the model. So the researchers must choose the evaluation metric, which is most effective for their application.

The novelty of this study is that the performance of traditional, soft computing, and deep learning-based edge detectors is evaluated by using a wide range of evaluation metrics, .i.e, precision, recall, accuracy, F-1 score, PSNR, mean Dice Coefficient (mDC), and Intersection over Union (IoU). The experimental results show that accuracy is not a sufficient evaluation metric for a model's performance in edge detection, because it relies on true negatives, and in the background of ground-truth images, these true negatives are around 85 to 90 per cent. So the accuracy result is not reliable. For research purposes, F-1 score/ mDC are the most reliable evaluation metrics.

2. Related Study

A brief literature survey of edge detectors is shown in Table 1. From Table (1, 2, and 3), it is observed that the Sobel, Prewitt, and Canny traditional edge detectors are commonly used for complex images with textural backgrounds and shifting illumination conditions. It is also observed that traditional edge detectors often fail to capture weak or discontinuous edges. Traditional operators (Sobel, Prewitt, Roberts, and LoG) detect hard edges but are sensitive to noise, discontinuities, and complex edge connections. It is also clear that edge connectivity, contrast, and image adaptability are improved by enhanced Canny or hybrid approaches, but such edge detectors fail under complex background conditions.PSNR, Precision, Recall, and F1-score are commonly used performance metrics for evaluating image quality. Traditional edge detectors perform better in simple environments when supported by mathematical models (such as fractional calculus and fuzzy logic). It is also concluded that no such traditional methods have explored all image types, noise levels, or modalities for edge detection. It is revealed that effective machine learning-based edge detection approaches are more robust than traditional edge detectors but fail on complex structures with textured backgrounds and variable illumination. It is also observed that insufficient training data degrades edge detection performance, especially when recognising weak or discontinuous edges. Fuzzy c-means with directional shifts has been reported to yield smoother and more accurate boundaries than traditional gradient operators [5]. Accuracy, Precision, Recall, and F1-score are commonly used performance metrics for evaluating image quality. Previous studies have widely adopted convolution-based encoder-decoder architectures, which are among the most commonly used deep learning models. The Bi-Directional Network and CHRNet refine feature maps and preserve spatial information to improve edge detection [35-39]. Compact architectures achieve high structural information scores, enabling faster and more practical deployment with respect to the similarity index [31]. The integration of RGB and depth information increases the architecture's F-scores and robustness in cluttered or low-contrast situations. Progressive edge-detail enhancement and dense residual connections improve the structural similarity index by preserving structural consistency [32]. Modern deep neural networks outperform traditional approaches in terms of structural similarity index and boundary continuity. From the exhaustive literature survey, it is observed that mIoU and the Dice coefficient are two important metrics for evaluating edge detection accuracy. An exhaustive review of the literature also indicates that most studies have focused on edge detection on the BSDS500 dataset.

Table 1. A brief literature review of traditional edge detectors

Investigator (s)	Year	Algorithm (s)	Performance Metrics	Outcomes
S. Abasi et al. [1]	2020	Sobel and Canny	Precision-recall	Compared Sobel and Canny
R. Keerthi et al. [2]	2020	Sobel, Canny and improved Canny	MSE and PSNR	Compared Sobel, Canny and improved Canny
S. Wang et al. [3]	2021	Enhanced Canny edge detector with bilateral filtering and global thresholding	Visual edge clarity and Robustness under noise and variable illumination	Noise-robust classical edge detection
V. Meherhomji et al. [4]	2021	Adaptive regional thresholding and Otsu-based local threshold computation	PSNR and Computational complexity analysis	Achieved better edge preservation and adaptability under varying illumination conditions
A. Lisowska [5]	2022	Local adaptive edge detection method with sliding-window k-means clustering	Compared qualitatively and quantitatively against classical approaches	Successfully identifies both prominent and subtle edges in images.

			(e.g., Canny, Sobel)	
Utama et al. [6]	2022	Roberts, Sobel, Prewitt, Canny	PSNR, Histogram, Contrast	Classical operator performance compared.
Y. I. Ibrahim et al. [7]	2023	Canny, Sobel, Prewitt, LOG, Roberts	BRISQUE, correlation coefficient	Compared classical edges and a hybrid with morphological operations.
R. Kumari et al. [8]	2023	Canny, Sobel, Prewitt	Accuracy, MSE, PSNR	Multi-metric evaluation
Najjar et al. [9]	2024	Ant colony optimization	MSE and PSNR	It reduced broken and fragmented edges commonly seen in gradient-based methods.
M. Chen et al. [10]	2024	Distributed ACO	Accuracy, PSNR, and MSE	Producing more continuous and well-defined edges
A. Aljohaniet al. [11]	2024	Particle swarm optimisation	Accuracy, precision and recall	This improved classification performance while reducing redundant features.
F. Nie et al. [12]	2025	Improved particle swarm optimization	Recall, Precision, F1, SSIM, FoM	Provides more stable optimal thresholds
Zou et al. [13]	2026	Zernike-moment edge detection method based on improved adaptive threshold using two-dimensional Otsu	Location accuracy, denoising effectiveness and robustness	Sub-pixel edge detection is more accurate, more automatic, and more robust to noise

Table 2. A brief literature survey on ML--based edge detectors

Author(s)	Year	Dataset	AdoptedMethod	Assessment Parameter(s)
Fuetal. [14]	2014	BSDS-500	GA	Acc., Pre, Re, & F1,
Ryuetal.[15]	2018	GrayScale	FCM	Comparison
Cuietal.[16]	2019	MUSI	Adaand DT	Visual
Liuetal.[17]	2019	CT Images	RF	Visual
Lietal.[18]	2019	GIS	SVM	Acc.
F. Vidaletal.[19]	2019	BSDS-500	FC	Visual
Cogranneetal.[20]	2019	BSDS-500	RSD	ODS, OIS, and AP
Hanetal.[21]	2019	MovingTarget	WT and NN	Comparison
Liuetal. [22]	2020	USI	SVM	RR
Vilasinietal.[23]	2020	LeafImages	KNN	Re, Preand F1
Wangetal. [24]	2021	SI	RF, LR and SVM	ReandPre
F. Vidaletal.[25]	2022	BSDS-500	SVM	Acc., Pre, Re, & F1,
Wuetal. [26]	2022	Standard	KMC	Acc., Pre
Prasadetal.[27]	2022	CT Scan	KNN	Comparison
Batistaetal.[28]	2024	UAV	SVR	MSE
X. Soria[29]	2025	BIPED	CNN	OIS, ODS and PSNR
Li, Y et al. [30]	2026	BSDS-500	Encoder-decoder	ODS

Table 3. A brief literature survey on DL-based edge detectors

Author(s)	Year	Dataset	Adopted Method	Assessment Parameter(s)
F. Wang et al. [31]	2023	BSDS-500	Bi-Directional Cascade Network	ODS, AP and OIS
O. Elharrouss et al. [32]	2023	BSDS-500, NYUD	A Cascaded and High-Resolution Network Named (CHRNet)	SI
D Jing et al. [33]	2023	BSDS-500	Edge Detection Network-EDD	SI

Al-Amaren et al. [34]	2023	BSDS-500	DCNNs	SI
Li Fuzhang et al. [35]	2024	BSDS-500, BIPED, and NYUD	UHNet	AP, ODS and OIS
J Zhang et al. [36]	2024	BSDS-500	PCNNs	SSIM and PSNR
F.Abedi et al [37]	2024	BSDS-500	DRNet	PSNR
A. Ying et al. [38]	2024	NYUDv2 and BSDS-500	U-net	ODS F1-score
K. Muntarina et al. [39]	2025	BSDS-500	HED, FCN, DeepEdgeUnet,	PSNR and MSE
Ming Wang [40]	2025	BSDS-500	PiDiNet	SI
Wenlin Li et al.[41]	2025	BSDS-500, Multicue, and BIPED	CNN	ODS F-score
X. Chen et al. [42]	2025	BSDS-500 and NYUDv2	GCN	SI
X. Yang et al. [43]	2025	BSDS-500, BIPED, NYUD	TANET	SI
Cetinkaya et al.[44]	2026	BSDS, Multi-Cue, BIPED	MatchEDAC	ODS, OIS, AP

Note - USI-US Image, GIS- Gas-Insulated Switchgear Images, SI-Satellite Images, SVM-Support Vector Machine, Ada-AdaBoost, DT-Decision Tree, SVR-Support Vector Regression, RSD-Regression Surface Descriptor RSD, KNN-K-Nearest Neighbours, RF-Random Forest, FCM- Fuzzy C-Mean, LR-Logistic Regression (LR), Re-recall, Pre-precision, F1-F1 Score, WT-Wavelet Transform, NN-Neural Network Activation Function, MAE- Mean Absolute Error, MSE-Mean Squared Error, Acc.-Accuracy, RR-Recognition Rate, FC-Fuzzy Clustering, UAV- Unmanned Aerial Vehicle, KMC-K-means clustering, GP -Genetic Programming, PB - Probability of Boundary

3. Research Methodology Adopted for Analysis of Edge Detectors

The research methodology used to analyse edge detectors follows a structured procedure that involves selecting the dataset, performing pre-processing steps, implementation of algorithms and analysing their performance using statistical metrics. The methodology used to analyse edge detectors is shown in Figure 1. The first step involves the selection of image datasets to be used, including popular standard benchmark sets consisting of natural images. After the datasets (BSDS500 & BIPED) have been selected, edge-detection methods are then implemented. These include the use of classic algorithms like the Sobel, Prewitt, Roberts, Laplacian of Gaussian (LOG), and Canny detectors, as well as advanced methods that utilise soft computing and deep learning techniques, including the ACO and PSO algorithms.

The evaluation is an important component of the methodological approach. Performance of the model is analysed using quantifiable parameters such as PSNR, Precision, Recall, F1-score/DC, and IoU. In addition to that, visual analysis is conducted to assess continuity of the edges, localisation capability, and susceptibility to noise. Moreover, statistical analysis is used to analyse different edge detection approaches in terms of consistency and reliability on several test images. A phase-wise description of the methodology is given in Figure 1.

3.1. Phase 1- Benchmark Dataset-BSDS500

The BSDS500 is commonly used for the detection of edges and the segmentation of images. It contains 500 natural images, split into four hundred for training and one hundred for testing. Each image in the dataset has been labelled by experts, so each image has many ground-truth masks [29-42].

3.2. Phase 2 - Analysis of the Dataset

For the first technique, the 400 images of the BSDS-500 set are expanded to 2,800 images through data augmentation. For each image, six data augmentation procedures are performed, including rotation, flipping, and scaling. The original image is kept during this process, leading to seven different images for each image (six images resulting from data augmentation plus one original image). Therefore, the number of images in the data set will be 400 x 7 = 2,800 images.

With the other technique, there are multiple ground truth annotations (five to six) per image in the BSDS500 dataset that represent the subjective nature of humans' edge detection. The dataset has 400 training images and 100 testing images; it contains around 2100 ground-truth maps and 500 ground-truths, respectively. In order to increase the size of the 2100 training images used for developing the models, an augmentation process was used for every image.

The augmentation involved one rotation per image along with the original image, resulting in two images. Thus, 4200 images resulted after this process. These images are further divided into training and validation datasets for model development purposes, where 3200 images and 1000 images are assigned to training and validation sets, respectively. Through this arrangement, learning from a large number of samples could be achieved while simultaneously evaluating the model's performance through the validation set [44].

3.3. Phase 3 - Edge Detectors Algorithms for Edge Detection

There are three main categories of edge detection techniques: conventional, soft computing, and deep learning. The conventional edge detectors include the Sobel, Prewitt, Roberts, LoG, and Canny operators. This category of techniques relies on the detection of edges through gradient changes and intensity differences. Although these algorithms are efficient in terms of computation, they are susceptible to noise and may have trouble detecting edges in complex images. The soft computing-based techniques rely on optimisation algorithms like ACO and PSO to improve edge detection results [44]. These techniques provide more accurate results compared to conventional edge detection algorithms in the presence of noise. The deep learning-based techniques use deep convolutional neural networks such as CNN, U-Net, and ResNet to learn hierarchical features.

The performance evaluation parameters are key to evaluating the efficiency of edge detectors. Parameters like precision, recall, accuracy, F1 score, PSNR, and IoU provide numerical performance indicators of edge detection. Precision and recall help in measuring the correctness and completeness of the results, whereas accuracy gives a holistic indicator of the performance. The F1 score combines recall and precision for a balanced evaluation of the algorithm's performance. PSNR measures the difference between the reconstructed edges from the original images, indicating how well they have been reconstructed. IoU measures how many pixels in the predicted edges overlap with actual edges.

All these parameters help in providing an overall evaluation of edge detector performance. Accuracy means the ratio of correct classifications to total classifications made. Recall or True Positive Rate (TPR) is the ratio of correctly classified positive examples. PSNR is always used to measure reconstruction accuracy. The F1 score and Dice Coefficient (DC) are the same, as they are the harmonic mean of precision and recall. These parameters are calculated using the following Equations [29-44]:

$$mDC = \frac{2T_{+ve}}{2T_{+ve} + F_{+ve} + F_{-ve}} \tag{1}$$

$$IoU = \frac{T_{+ve}}{T_{+ve} + F_{+ve} + F_{-ve}} \tag{2}$$

$$Accuracy = \frac{T_{+ve} + T_{-ve}}{T_{+ve} + T_{-ve} + F_{+ve} + F_{-ve}} \tag{3}$$

$$Precision = \frac{T_{+ve}}{T_{+ve} + F_{+ve}} \tag{4}$$

$$Recall = \frac{T_{+ve}}{T_{+ve} + F_{-ve}} \tag{5}$$

$$F1\ Score = 2 \times \frac{Prec. \times Recall}{Prec. + Recall} \tag{6}$$

$$\text{PSNR} = 10 \log_{10}\left(\frac{R^2}{MSE}\right) \tag{7}$$

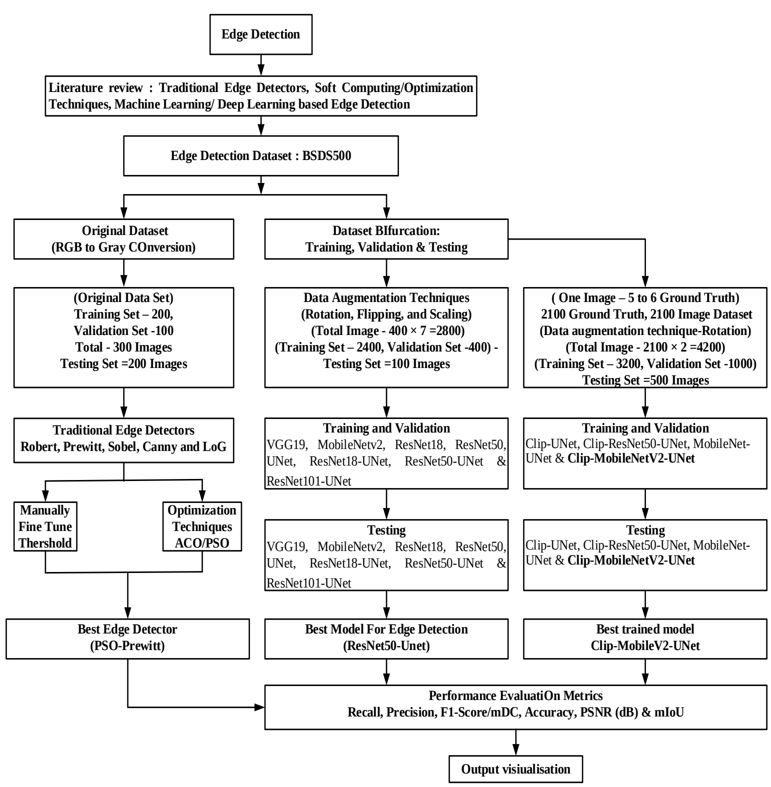


Fig. 1 The research methodology adopted for the analysis of edge Detectors

4. Edge Detector Architectures

The traditional Edge Detector calculates edges by estimating the first-order derivative (intensity gradient) of an image. It relies on 2D convolution using two separate 2 × 2 or 3 × 3 kernels to look for sudden intensity changes horizontally and vertically. The basic structural architectures based on traditional edge detector with example Prewitt is shown in Figure 2.

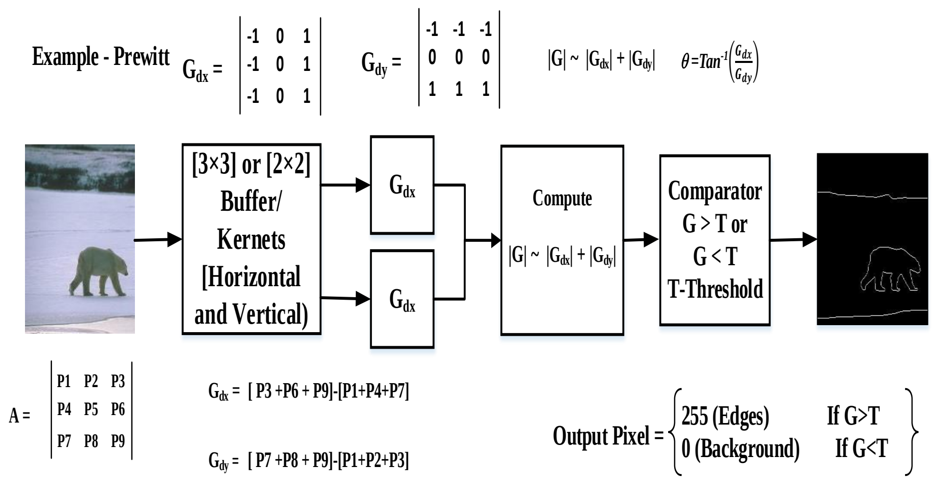
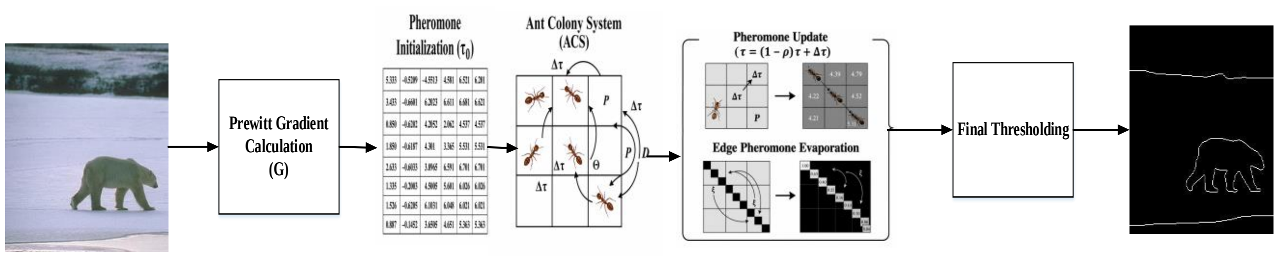
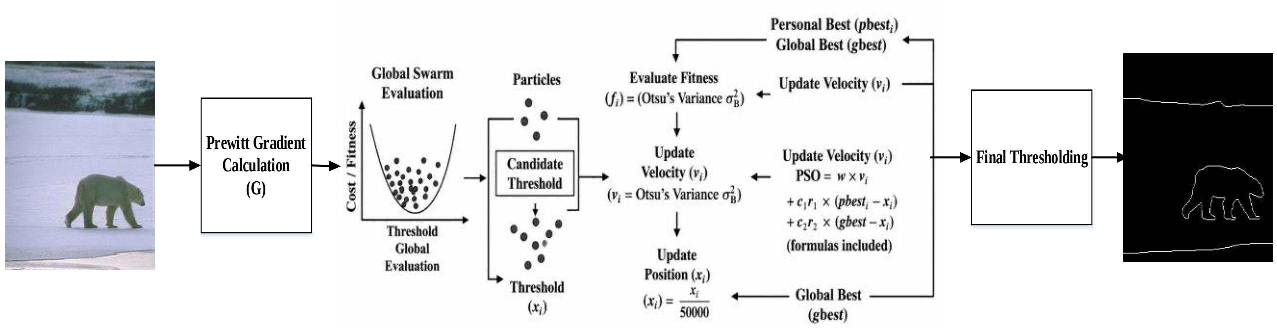


Fig. 2 Architecture based on traditional edge detector (example-prewitt)

The combination of the Prewitt Edge Detector with ACO or PSO does not change the core Prewitt math. Instead, ACO and PSO are used as metaheuristic optimization algorithms to solve the two biggest weaknesses of standard Prewitt: finding the optimal edge threshold and linking broken edge pixels. The basic structural architecture is shown in Figure 3.



(a)



(b)

Fig. 3 Architecture based on traditional + optimization based edge detector (example - (a) Prewitt+ACO, and (b) Prewitt+PSO).

The Clip_MobileNetV2_UNet architecture is a lightweight, hybrid deep learning framework designed primarily for edge detection and semantic image segmentation on resource-constrained mobile and embedded devices. The model structurally merges three primary engineering methodologies: the lightweight MobileNetV2 architecture as a feature extractor (encoder), the symmetric upsampling path of a standard U-Net for pixel-level boundary rebuilding (decoder), and Clip Rectified Linear Unit (Clip ReLU) bounded activation functions instead of standard activations to stabilize numerical gradients. The basic structural architecture of Clip-MobilNetv2-Unet is shown in Figure 4.

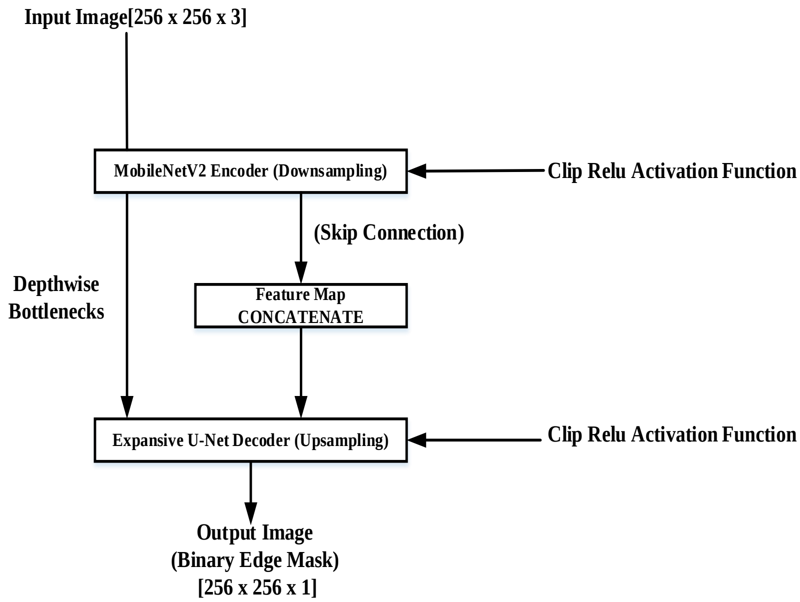


Fig. 4 The basic structural architecture of Clip-Mobilenetv2-Unet

4.1. Training Strategy and Hyperparameter Selection

An effective training strategy for a deep learning-based edge detector must preserve the Convolutions while properly constraining gradients to leverage the Clip ReLU activation. Edge detection and pixel-level edges suffer from severe class imbalance (background pixels heavily outnumber edge pixels); the loss function, learning rate schedule, and optimization selection require specialized tuning. The selected hyperparameters used for experimental purposed to avoid overfitting are shown in Table 4.

Table 4. Training parameters used to prevent overfitting

Serial No.	Parameters	Selected Value
1	Epochs	30
2	Learning Rate	0.0001
3	Batch Size	32
4	Optimizer	Adam

5. Result and Discussion

The experimental results show the comparative performance of traditional, soft computing-based, and deep learning-based edge detection algorithms using performance evaluation metrics. Soft computing approaches, ACO and PSO-based methods, improve edge continuity and optimize threshold selection, resulting in better Precision and F1-scores compared to traditional techniques. However, the performance of soft computing-based algorithms depends on parameter tuning. Table 5 presents the experimental results using the evaluation metrics.

Table 5. Experimental results based on evaluation metrics

Dataset Used	Edge Detectors	Recall	Precision	F1-Score/mDC	Accuracy	PSNR (dB)	mIoU
Without Augmentation (Original Dataset)	Robert	0.14	0.54	0.20	0.97	17.88	0.16
	Prewitt	0.53	0.49	0.46	0.93	14.78	0.39
	Sobel	0.68	0.44	0.50	0.90	13.72	0.36

	LoG	0.94	0.31	0.45	0.73	16.18	0.39
	Canny	0.65	0.40	0.46	0.94	15.17	0.38
	Robert + PSO	0.76	0.45	0.54	0.87	10.31	0.46
	Prewitt+ PSO	0.76	0.46	0.54	0.87	16.24	0.45
	Sobel+ PSO	0.76	0.45	0.54	0.87	13.34	0.46
	LoG + PSO	0.79	0.41	0.51	0.86	11.93	0.41
	Canny + PSO	0.71	0.44	0.52	0.94	13.26	0.44
	Robert + ACO	0.71	0.44	0.50	0.87	13.37	0.39
	Prewitt + ACO	0.71	0.46	0.51	0.87	15.56	0.40
	Sobel + ACO	0.76	0.45	0.54	0.87	12.4	0.46
	LoG + ACO	0.75	0.40	0.49	0.86	10.07	0.37
	Canny + ACO	0.72	0.45	0.52	0.94	13.21	0.44
Augmentation – Type 1 Training-2400 Validation-400 Testing-200	VGG19	0.74	0.45	0.56	0.83	14.01	0.47
	MobileNet	0.73	0.52	0.61	0.89	14.89	0.53
	ResNet18	0.72	0.56	0.63	0.91	15.14	0.55
	ResNet50	0.70	0.61	0.65	0.93	15.38	0.57
	UNet	0.89	0.55	0.68	0.92	14.15	0.62
	ResNet18-UNet	0.82	0.63	0.71	0.94	14.89	0.66
	ResNet50-UNet	0.98	0.72	0.90	0.97	16.16	0.82
	ResNet101-UNet	0.82	0.63	0.71	0.93	15.59	0.65
Augmentation – Type 2 Training-3200 Validation-1000 Testing-500	Clip-U-Net	0.90	0.58	0.86	0.93	15.34	0.79
	Clip-ResNet50-UNet	0.83	0.73	0.89	0.92	18.06	0.82
	MobileNet-UNet	0.89	0.67	0.75	0.90	16.69	0.80
	Clip-MobileNetV2-UNet (Hybrid Model)	0.95	0.89	0.92	0.96	19.54	0.84

It is observed that Prewitt+PSO, ResNet50-UNet and hybrid architecture based on Clip-MobileNetV2-UNet outperformed in their categories.

For cross-validation, the BIPED dataset validates the model's accuracy, resilience, and compatibility for a range of natural settings.

The results using the BIPED dataset of the best performing technique in their category are represented in Table 6. The experimental results concluded that the Prewitt edge detector achieved comparatively better performance among the other traditional methods. Prewitt shows improved Precision and Recall, effectively detecting true edges while minimising false detections.

The model accuracy and model loss values are shown in Figure 5 (a) and (b) for the database without augmented and the augmented database (BSDS-500), respectively, for Clip-ResNet50-Unet and Clip-MobileNetV2-UNet. It also performs well in terms of F1-score and IoU, indicating better arrangement with ground-truth edges.

It is also concluded that deep learning algorithms achieve higher overall performance in terms of metrics. From the results, it is observed that the Clip-based MobileNetv2-UNetArchitecture outperformed the other methods in terms of performance metrics. It is found that the Clip-MobileNetv2-UNet achieved 0.95, 0.89, 0.92, 0.96, 19.54 and 84.19 Recall, Precision, F1-Score/mDC, Accuracy, PSNR (dB) and IoU, respectively.

Table 6. The best performing network is analysed using the BIPED dataset

Dataset →	**Testing of BIPED Dataset**					
P.E.M → Model (s) ↓	**Rec.**	**Pre**	**F1 Score/mDC**	**Acc.**	**PSNR(dB)**	**IoU**
Prewitt+ PSO	0.67	0.37	0.44	0.83	13.48	0.37
ResNet50-UNet	0.70	0.52	0.59	0.87	16.01	0.49
Clip-MobileNetV2-UNet	0.89	0.81	0.84	0.88	16.84	0.78

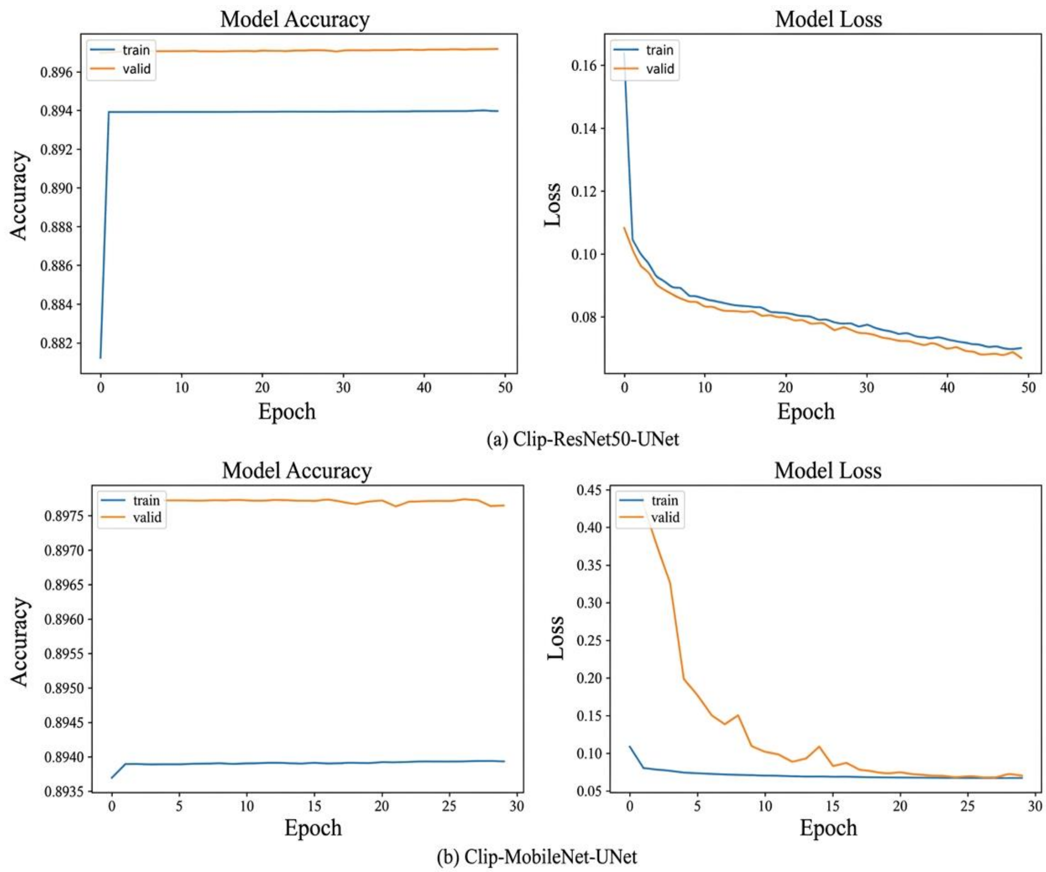


Fig. 5 The model accuracy and model loss are based on the number of epochs

The visual interpretation of the best-performing edge detection models is shown in Figure 6.

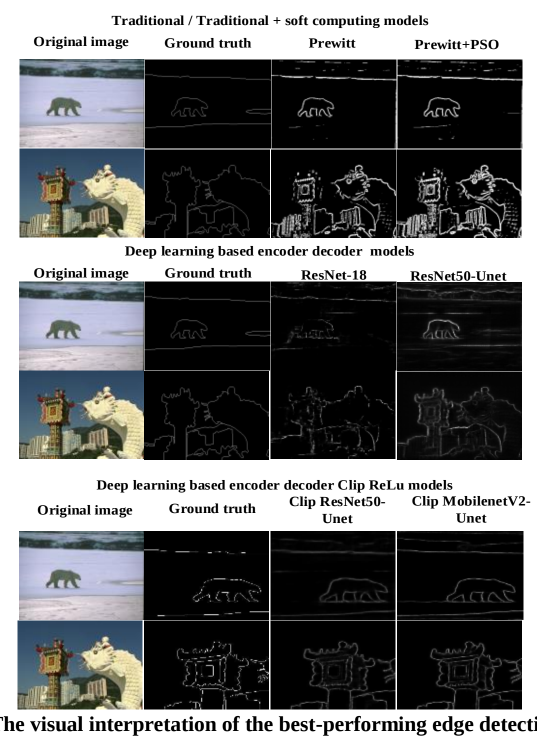


Fig. 6 The visual interpretation of the best-performing edge detection models

6. Conclusion

Prewitt+PSO, ResNet50-UNet and Clip-MobileNetV2-Unet models are trained using the BSDS-500 and validated using the BIPED dataset, and it is observed that performance deteriorates in the initial stage due to differences in the datasets' features. The BSDS500 delivers only five hundred low-resolution images compared to subjective type annotations, whereas BIPED provides high-resolution, authentic images with more specific, reliable edge annotations, particularly designed for edge detection. As a result of the domain shift, the model will have few problems retaining its accuracy, leading to lower values of Precision, F1-score, Recall, PSNR, and IoU. Nevertheless, evaluating the model's performance on BIPED is necessary in order to assess the model's efficiency and its ability to generalize. The performance drop observed from the experiment indicates that using only BSDS-500 for training reduces the resilience of the model, but proper tuning or transfer-learning with the BIPED dataset notably boosts the model's performance. On the other hand, this technique not only increases edge detection accuracy but also proves the adaptability of the model across various datasets. Thus, BIPED can be considered a better measure of validating the effectiveness of BSDS500. Training loss and validation loss are two measures of estimating the performance of a deep learning model over the period of the learning stage. Training loss indicates errors made on the training set, while validation loss determines the generalization capability of the network on unseen data. Ideally, both training and validation losses should decrease as the model learns. Overfitting occurs when the training loss decreases but the validation loss increases.

References

[1] Saeedeh Abasi, Mohammad A. Tehran, and Mark D. Fairchild, "Colour Metrics for Image Edge Detection," *Color Research and Application*, vol. 45, no. 4, pp. 632-643, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ramya Keerthi M.R., and T.P. Mithun, "Edge Connectivity Techniques for Image Analysis - A Survey," *International Research Journal of Engineering and Technology*, vol. 7, no. 5, pp. 5374- 5379, 2020. [Google Scholar]

[3] Shigang Wang, Xianghua Liao, and Guoqiang Wu, "Infrared Image Edge Detection based on Improved Canny Algorithm," *2021 IEEE 3rd Eurasia Conference on IoT, Communication and Engineering*, Yunlin, Taiwan, pp. 280-284, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Vishtasp Meherhomji, and K.B. Ajitha Shenoy, "An Improved Edge Detection Technique," *International Journal of Computational Vision and Robotics*, vol. 11, no. 6, pp. 653-670, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Agnieszka Lisowska, "Efficient Edge Detection Method for Focused Images," *Applied Sciences*, vol. 12, no. 22, no. 1-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kgs Muhammad Rizky Alditra Utama, Rusydi Umar, and Anton Yudhana, "Edge Detection Comparative Analysis using Roberts, Sobel, Prewitt, and Canny Methods," *Journal of Technology and Computer Systems*, vol. 10, no. 2, pp. 67-71, 2022. [Google Scholar]

[7] Yahya Ismail Ibrahim, and Israa Mohammed Alhamdani, "Practical Study for Comparing Edge Detection Filters in Digital Image Processing," *Tikrit Journal of Pure Science*, vol. 28, no. 5, pp. 201-215, 2023. [CrossRef] [Publisher Link]

[8] Rajshree Kumari, and Divyanshu Chandra, "Real-Time Comparison of Performance Analysis of Various Edge Detection Techniques based on Imagery Data," *Current Journal of Applied Science and Technology*, vol. 42, no. 24, pp. 22-31, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Fallah H. Najjar, Ola N. Kadhim, and Salman kadum Abd, "Ant Colony Optimization based Edge Detection Algorithm," *Al-Furat Journal of Innovations in Electronics and Computer Engineering*, pp. 479-487, 2024. [CrossRef] [Google Scholar]

[10] Min Chen, "A Distributed Ant Colony Optimization Applied in Edge Detection," *Journal of Computer and Communications*, vol. 12, no. 8, pp. 161-173, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Abeer Aljohani, "Optimized Convolutional Forest by Particle Swarm Optimizer for Pothole Detection," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Fangyan Nie, Mengzhu Liu, and Pingfeng Zhang, "Multilevel Thresholding with Improved Particle Swarm Optimization for Crack Image Segmentation," *Scientific Reports*, vol. 14, no. 1, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yuanyuan Zou, Shilong Xu, and Boxuan Chen, "An Improved Zernike Moment Subpixel Edge Detection Algorithm based on Adaptive Threshold," *Internet of Things, Artificial Intelligence and Mechanical Automation*, Springer, vol. 641, pp. 211-222, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Wenlong Fu, Mark Johnston, and Mengjie Zhang, "Unsupervised Learning for Edge Detection using Genetic Programming," *2014 IEEE Congress on Evolutionary Computation*, Beijing, China, pp. 117-124, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Y. Ryu et al., "Image Edge Detection using Fuzzy C-Means and Three Directions Image Shift Method," *IAENG International Journal of Computer Science*, vol. 45, no. 1, pp. 1-6, 2018. [Google Scholar]

[16] Jichao Cui, and Kun Tian, "Edge Detection Algorithm Optimization and Simulation based on Machine Learning Method and Image Depth Information," *IEEE Sensors Journal*, vol. 20, no. 20, pp. 11770-11777, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Caixia Liu, Ruibin Zhao, and Mingyong Pang, "Lung Segmentation based on Random Forest and Multi-Scale Edge Detection," *IET Image Processing*, vol. 13, no. 10, pp. 1745-1754, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Xi Li et al., "Partial Discharge Source Localization in GIS based on Image Edge Detection and Support Vector Machine," *IEEE Transactions on Power Delivery*, vol. 34, no. 4, pp. 1795-1802, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Pablo A. Flores-Vidal et al., "A New Edge Detection Method based on Global Evaluation using Fuzzy Clustering," *Soft Computing*, vol. 23, no. 6, pp. 1809-1821, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Rémi Cogranne et al., "A New Edge Detector based on Parametric Surface Model: Regression Surface Descriptor," *arXiv preprint*, pp. 1-21, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Fangfang Han et al., "Algorithm Design for Edge Detection of High-Speed Moving Target Image under Noisy Environment," *Sensors*, vol. 19, no. 2, pp. 1-27, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Yangyang Liu et al., "Breast Tumours Recognition based on Edge Feature Extraction using Support Vector Machine," *Biomedical Signal Processing and Control*, vol. 58, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] M. Vilasini, and P. Ramamoorthy, *Leaf Recognition using Prewitt Edge Detection and K-NN Classification*, New Trends in Computational Vision and Bio-inspired Computing, Coimbatore, India, Springer, pp. 1507-1515, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Yu Wang et al., "Machine Learning-based Ship Detection and Tracking using Satellite Images for Maritime Surveillance," *Journal of Ambient Intelligence and Smart Environments*, vol. 13, no. 5, pp. 361-371, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Pablo Flores-Vidal, Javier Castro, and Daniel Gómez, "Postprocessing of Edge Detection Algorithms with Machine Learning Techniques," *Mathematical Problems in Engineering*, vol. 2022, no. 1, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Fangsheng Wu et al., "Research on Image Text Recognition based on Canny Edge Detection Algorithm and K-Means Algorithm," *International Journal of System Assurance Engineering and Management*, vol. 13, no. 1, pp. 72-80, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] G.S.K. Ganesh Prasad et al., "Detection of CKD from CT Scan Images using KNN Algorithm and using Edge Detection," *2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies*, Patna, India, pp. 1-4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Gracieth Cavalcanti Batista et al., "Machine Learning Algorithm Partially Reconfigured on FPGA for an Image Edge Detection System," *Journal of Electronic Science and Technology*, vol. 22, no. 2, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] Xavier Soria et al., "Tiny and Efficient Model for the Edge Detection Generalization," *2023 IEEE/CVF International Conference on Computer Vision Workshops*, Paris, France, pp. 1356-1365, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] Yachuan Li et al., "A New Baseline for Edge Detection: Make Encoder-Decoder Great Again," *Signal Processing: Image Communication*, vol. 142, pp. 1-12, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[31] Fuping Wang, and Min Zhang, "Deep Learning-based Edge Detection Algorithm for Noisy Images," *Proceedings of the 2023 6th International Conference on Artificial Intelligence and Pattern Recognition*, pp. 465-472, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Omar Elharrouss et al., "Refined Edge Detection with Cascaded and High-Resolution Convolutional Network," *Pattern Recognition*, vol. 138, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Dongdong Jing et al., "Edge Detection in Dark Industrial Environments," *2023 9th International Conference on Computer and Communications*, Chengdu, China, pp. 1689-1693, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Abdullah Al-Amaren, M. Omair Ahmad, and M.N.S. Swamy, "A Low-Complexity Residual Deep Neural Network for Image Edge Detection," *Applied Intelligence*, vol. 53, no. 9, pp. 11282-11299, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[35] Fuzhang Li, and Chuan Lin, "UHNet: An Ultra-Lightweight and High-Speed Edge Detection Network," *arXiv preprint*, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[36] Jiahao Zhang, Wei Wang, and Jianfei Wang, "Edge Detection in Colored Images using Parallel CNNs and Social Spider Optimization," *Electronics*, vol. 13, no. 17, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[37] Firas Abedi et al., "Dense Residual Network for Image Edge Detection," *Multimedia Tools and Applications*, vol. 83, no. 42, pp. 90227-90242, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[38] Ying An et al., "An Exclusive U-Net for Fine and Crisp Edge Detection," *Multimedia Tools and Applications*, vol. 83, no. 18, pp. 54657-54672, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[39] Kanija Muntarina et al., "Deep Learning-based Edge Detection for Random Natural Images," *Neuroscience Informatics*, vol. 5, no. 1, pp. 1-10, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[40] Ming Wang, "AttnEdge: An Enhanced Edge Detection Method based on Self-Attention Mechanism," *Fourth International Conference on Computer Vision, Application, and Algorithm*, vol. 13486, pp. 438-446, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[41] Wenlin Li et al., "Pixel-Patch Combination Loss for Refined Edge Detection," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 2, pp. 1341-1354, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[42] Xiaodiao Chen et al., “Sufficient Learning: Mining Denser High-Quality Pixel-Level Labels for Edge Detection,” *Neural Computing and Applications*, vol. 37, no. 14, pp. 8245-8260, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[43] Xi Yang et al., “Textureness-Aware Neural Network for Edge Detection,” *Chinese Conference on Pattern Recognition and Computer Vision*, Springer, Singapore, pp. 254-268, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[44] Bedrettin Cetinkaya, Sinan Kalkan, and Emre Akbas, “MatchED: Crisp Edge Detection using End-to-End, Matching-based Supervision,” *arXiv preprint*, pp. 1-27, 2026. [CrossRef] [Google Scholar] [Publisher Link]