Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

An optimize canny algorithm with traditional machine learning for edge detection enhancement

Russel K. Lafta*, **Zainab N. Sultani**

Computer Science Dept., College of Science, Al-Nahrain University, Jadriya, Baghdad, Iraq.
*Corresponding author Email: st.russelkareem22@ced. nahrainuniv.edu.iq

HIGHLIGHTS

- Canny edge detection was enhanced using FPA for optimal threshold selection
- A bilateral filter was used to reduce noise while preserving fine details
- The method achieved higher accuracy than traditional edge detection techniques
- Precision and recall were balanced, reducing false edge detection
- F1-score, precision, and recall were used for quantitative validation

Keywords:

Edge detection

Optimized canny

Flower Pollination algorithm

Machine learning

ABSTRACT

Edge detection still represents a major challenge in image processing and computer vision because of the complexity and variability of real-world images. The canny algorithm is powerful in detecting edges. However, it is sensitive to noise, and as a result, may produce weak edges. The use of machine learning algorithms has significantly improved the performance of edge detection techniques. In this paper, an improved Canny edge detection algorithm is proposed by replacing the Gaussian filter with a bilateral filter. Also, a new approach for estimating Canny algorithm thresholds has been developed using the Flower Pollination algorithm. Subsequently, the improved Canny algorithm with a machine learning model was integrated to enhance edge detection accuracy. The performance of the improved algorithm was evaluated using 50 images from the Berkeley Computer Vision dataset. The experiment results show that the enhanced algorithm has an AUC of 0.81 for RF (Random Forest) and 0.75 for the LR (Logistic Regression) classifier, which can detect edges more accurately than the traditional Canny algorithm, which has an AUC of about 0.57. The proposed method sets a new standard for edge detection performance.

1. Introduction

Gathering the notable edge of natural images implies a difficulty for computer vision applications [1]. Because of the intricacy of images, the elements in them (plants, houses, cars), and the conflict among the objects of images, make the extraction of edges a complex task using statistical-based methods [2]. Edges in images are the curves that describe the frontier of objects. In image processing, edge detection is basically important as it can quickly identify the boundaries of objects in an image [3]. Moreover, edge is used to classify the image to reduce the amount of data to be treated. Also, computer vision technology has been advancing; edge detection is considered crucial for more challenging tasks like object detection [4], object proposal [5] and image segmentation [6]. Consequently, it is necessary to develop a proper method for edge detection.

Classical image edge detection Techniques have been suggested previously and advanced for an extended duration. Wherefore, classical image edge detection techniques are more grown, uncomplicated but effective. However, scholars are still working tirelessly on the road to enhancement traditional image edge detection methods, working to overcome the weaknesses of previous algorithms and enhance their performance [7]. As a result, more robust methods are needed that can more accurately identify edges in noisy and complex images. For that reason, an image edge detection algorithm based on machine learning (ML) and image pixel information is suggested to explore the problem of image edge detection [8]. In this research, an algorithm that executes a categorization at the pixel level is suggested. This way permits grouping a pixel as "edge" or "non-edge," taking into account the pixel details associated with other local information that contains its neighborhood and the segment to which that pixel is part of. Furthermore, this paper's approach does not consist of receiving the feature vector from the entire image (or a sub-image) as has been achieved in so many research studies.

http://doi.org/10.30684/etj.2025.158177.1914
Received 12 March 2025; Received in revised form 07 April 2025; Accepted 15 April 2025; Available online 28 April 2025
2412-0758/University of Technology-Iraq, Baghdad, Iraq

Instead of that, the feature vector is generated for every pixel. Moreover, we address the challenge of selecting optimal variable values by applying the Flower Pollination Algorithm, an artificial intelligence technique, to set the threshold for the Canny edge detector. The Bilateral Filter is applied to suppress noise while preserving edges, enhancing image clarity. Furthermore, machine learning techniques are incorporated to refine the detection process, leading to higher accuracy and reliability. This combination effectively minimizes noise interference and improves the final results. This paper is organized as follows: Section 2 discusses related works. Section 3 explains the paper's methodology. Section 5 demonstrates the results and Discussion. Finally, Section 6 demonstrates the conclusions of this paper.

2. Related work

In the last few years, machine learning techniques have been used to find solutions for image processing problems [9-14]. Machine learning has been commonly implemented for applied research that is logically related to high-level processing tasks. As this paper illustrates, there are many characteristics of hiring ML for edge detection. Researchers have proposed using machine learning algorithms for edge detection, which has become a popular study area. Maksimovic et al. [15], presented an adaptive edge detection algorithm that relies on Spatial Information (SI) to measure image complexity, derived from Sobel filter gradients. Images were categorized into low, medium, and high complexity based on the mean SI value, with a subjective complexity assessment showing a 53% match with objective SI measurements. A machine learning-based threshold estimation approach, utilizing network and random search techniques, was tested on the BSD dataset. The results indicate that the Roberts operator is more effective for images with low to medium detail, while the Canny operator performs better on high-detail images. Although network search enhances edge detection, it demands extensive computational resources, whereas random search optimizes parameters more efficiently, balancing computational time and accuracy.

Flores-Vidal et al.[8], proposed a pixel-level edge detection method, classifying pixels as "edge" or "non-edge" based on their local neighborhood details. Unlike traditional approaches that extract feature vectors from entire images or sub-images, this method generates a feature vector for each pixel individually. The study utilized the Berkeley Segmentation Dataset and applied machine learning techniques, including Random Forest, Neural Networks, and Logistic Regression, with the best results achieved using Neural Networks. However, the algorithm relies on the Sobel operator, which, despite its common use, has limitations compared to more advanced methods like the Canny edge detector. Integrating these techniques can enhance the algorithm’s performance, improving both accuracy and robustness.

Yamagiwa et al. [16], introduced the SESAME model for zero-edge detection using the Segment Anything Model (SAM), incorporating noise reduction and spectral pooling to aggregate spatial information. SESAME improves SAM by filtering out small, insignificant masks, preserving critical edge details, and eliminating segmentation artifacts, resulting in cleaner edge maps. Experiments on the BSDS500 and NYUDv2 datasets demonstrated that SESAME performs comparably to human and older CNN-based methods on BSDS500. It nearly matches state-of-the-art CNN-based methods on NYUDv2, highlighting its effectiveness in enhancing SAM for edge detection. Overall, these studies represent significant advancements in edge detection. However, they share common limitations such as sensitivity to noise, computational complexity, and reliance on traditional methods like Sobel, which limits the accuracy of edge detection compared to more advanced techniques like Canny.

3. An optimized canny algorithm

3.1 Traditional canny

The primary aim of ED methods is to find those pixels where intensity varies significantly. An ED method converts an image into a binary image. In that image, the white pixels describe the pixels that the ED algorithm has recognized as edge pixels. In contrast, the black pixels are recognized as non-edge pixels. A popular example of an ED algorithm is the Canny operator [17]. It executes via five ED steps:

1) Smoothing: Blur the image using a Gaussian filter to take out the noise.
2) Calculate Gradient: reckon the gradient magnitude and direction.
3) Non-Maximum Suppression (NMS): Thin the edges by retaining only the pixel that has the greatest value in the gradient direction.
4) Double Threshold: Double threshold processing involves removing false edges and connecting discontinuous edges by applying a high threshold and a low threshold. This process follows non-maxima suppression.
5) Track Edge by Hysteresis: In this step, edges not connected to strong edges are removed. Strong edges, with a gradient magnitude higher than the upper threshold, are included directly. Weak edges, within magnitudes between the lower and upper thresholds, are included if connected to strong edges.

3.2 Problem with classical canny

While the Classical Canny algorithm is widely implemented, there are still some facets to enhance. First, the use of Gaussian filtering makes the edge detail look blurry because it depends on the first-order limited difference of a 2×2 neighbors area to evaluate the image gradient, which is more sensitive to noise. To solve this problem, the bilateral filter can be executed to preserve the details of the image and remove the noise. The next aspect is that the double threshold of the classical Canny algorithm is established manually. For images with a lot of information, the adaptability of the classical Canny algorithm is not perfect, and it is prone to losing a lot of important edges. This paper utilized an AI algorithm called Flower Pollination for the optimization of the threshold value based on the image content.

3.3 The improved canny algorithm

Bilateral filtering is a method for smoothing images while preserving edges. The utilization of bilateral filtering has expanded rapidly and is now applied in image processing applications such as image noise reduction, image improvement, etc. As it is easy to formulate. Each pixel is exchanged by a weighted average of its neighbors [18]. We chose to use the bilateral filter because of its superior ability to preserve edges while reducing noise. Unlike the Gaussian filter, it maintains sharp boundaries between different regions while effectively reducing noise, making it a preferred choice for applications requiring both smoothing and edge preservation. In this paper, the Flower Pollination Algorithm (FPA) is utilized to obtain optimal threshold values for the Canny Edge Detector. The FPA is a population-based optimization technique influenced by the pollination process of flowers [19]. Precisely, it imitates the global pollination process executed by pollinators such as insects, and the local pollination process that happens within a flower or between flowers of the same plant. The algorithm combines these two processes to balance exploration and harnessing in the search space.

One advantage of using FPA is its ability to efficiently explore complex, high-dimensional spaces while avoiding local optima. This makes it particularly suitable for tuning parameters in edge detection tasks. To determine the optimal thresholds for the Canny edge detection algorithm, as follows. Initially, it sets up the FPA parameters and initializes a population of solutions, with each solution representing a pair of threshold values. The objective function evaluates each solution based on the sum of gradient magnitudes above the threshold, aiming to maximize edge quality.

Through iterative optimization, the algorithm updates the thresholds by either exploring new solutions influenced by the global best solution or by local interactions between solutions. Boundary checks ensure the thresholds remain within specified limits. The best solution found by the FPA is then used to classify strong and weak edges in the suppressed image, enhancing the overall edge detection performance. In general, FPA has been used for threshold tuning because of its ability to find an optimal threshold, leading to better edge detection results across various images and conditions.

After applying the optimized Canny edge detection method, which effectively identifies the edges in the image, machine learning techniques will be employed for further enhancement. This integration will significantly optimize the edge detection results and contribute to more reliable and efficient outcomes.

4. Edge detection classification

During this research, various predictor variables are created to supply the required information for making the choice pixel by pixel. These predictors are gathered from the image presented by the optimized Canny edge detection method (Section 3) as an entry. Three groups of predictor variables are extracted: The pixel predictor variables, the neighborhood predictor variables, and the segment predictor variables. The (v1-v36) predictor variables with their corresponding ground truth used for the ML models are constructed using the Berkeley Image Segmentation Data Set (BSD) [20], following the procedure explained in [8]. To improve the performance of our machine learning models, a Chi-square feature selection is applied. This statistical technique evaluates the independence of two categorical variables and identifies the most relevant features for the model. By calculating the Chi-square statistic for each feature concerning the target variable, we selected those features that had the highest significance. This process effectively reduced the dimensionality of our dataset, removing less informative features and retaining those that help the most to the predictive power of the model. The best Logistic model found employed 20 variables: {v1, v2, v3, v5, v7, v8, v9, v15, v17, v18, v20, v21, v22, v23, v24, v25, v26, v27, v28, v29}. The dataset has been divided into training and testing sets, with a 70:30 training-to-testing ratio.

In this paper, the focus is on two notable supervised ML classification models: Logistic Regression (LR) and Random Forest (RF) [21]. These classification models are then applied for binary classification, as there are two potential results for a given pixel: “edge” or “non-edge”(pixels that do not represent edges). The LR was implemented using gradient descent accompanied by a learning rate of 0.01 and a total of 100 iterations. For the RF algorithm, the number of trees was restricted to 100. This specific number was chosen to balance computational efficiency and performance, ensuring sufficient diversity among the trees to enhance the model's robustness and accuracy. The flowchart of the suggested methodology is presented in Figure 1.

5. Results and discussion

The results presented in Table 1 emphasize the performance of two classifiers: Logistic Regression, and Random Forest, in addition to Canny, across four metrics: Area Under the Curve (AUC), Recall, Precision, and Accuracy. Logistic Regression has an AUC of 0.75, indicating afair ability to differentiate between classes. However, it exhibits a recall of 31.82%, suggesting it correctly identifies less than one-third of the positive instances. Its precision is 21.94%, indicating that only about one-fifth of its positive predictions are correct. The accuracy is 85.94%, and the F-measure, which balances precision and recall, is 25.97%. The Random Forest classifier demonstrates a more robust outcome with an AUC of 0.81, highlighting a good ability to differentiate between classes. It achieves a high precision of 95.86%, meaning that nearly all its positive predictions are correct. However, its recall is significantly lower at 10.13%; the low recall could be due to imbalanced data distribution, where the model favors precision over recall, causing it to miss many positive instances. The accuracy is 93.17%, and the F-measure is 18.32%, reflecting the trade-off between its high precision and low recall.

In contrast, the Canny algorithm performs poorly across all metrics. Its AUC is 0.57, accuracy is 84.01%, recall is 5.26%, precision is 13.85%, and F-measure is 7.71%. These values indicate that Canny struggles with both identifying positive instances and making correct positive predictions. A comparative analysis with existing edge detection methods, such as Sobel and traditional Canny, shows that our optimized approach yields a higher edge accuracy. This comparison highlights the effectiveness of our method, particularly in scenarios with high noise levels. In conclusion, the results suggest that while

Random Forest provides the best overall performance, particularly in precision, it still suffers from a low recall. Logistic Regression, while having a more balanced recall and precision, does not perform as well overall. The Canny algorithm, as expected, performs the worst across all metrics. However, the proposed methodology, which involves the subsequent processing of edge detection outputs with machine learning methods, shows promise in improving edge detection performance, as evidenced by Random Forest's superior results compared to Canny.

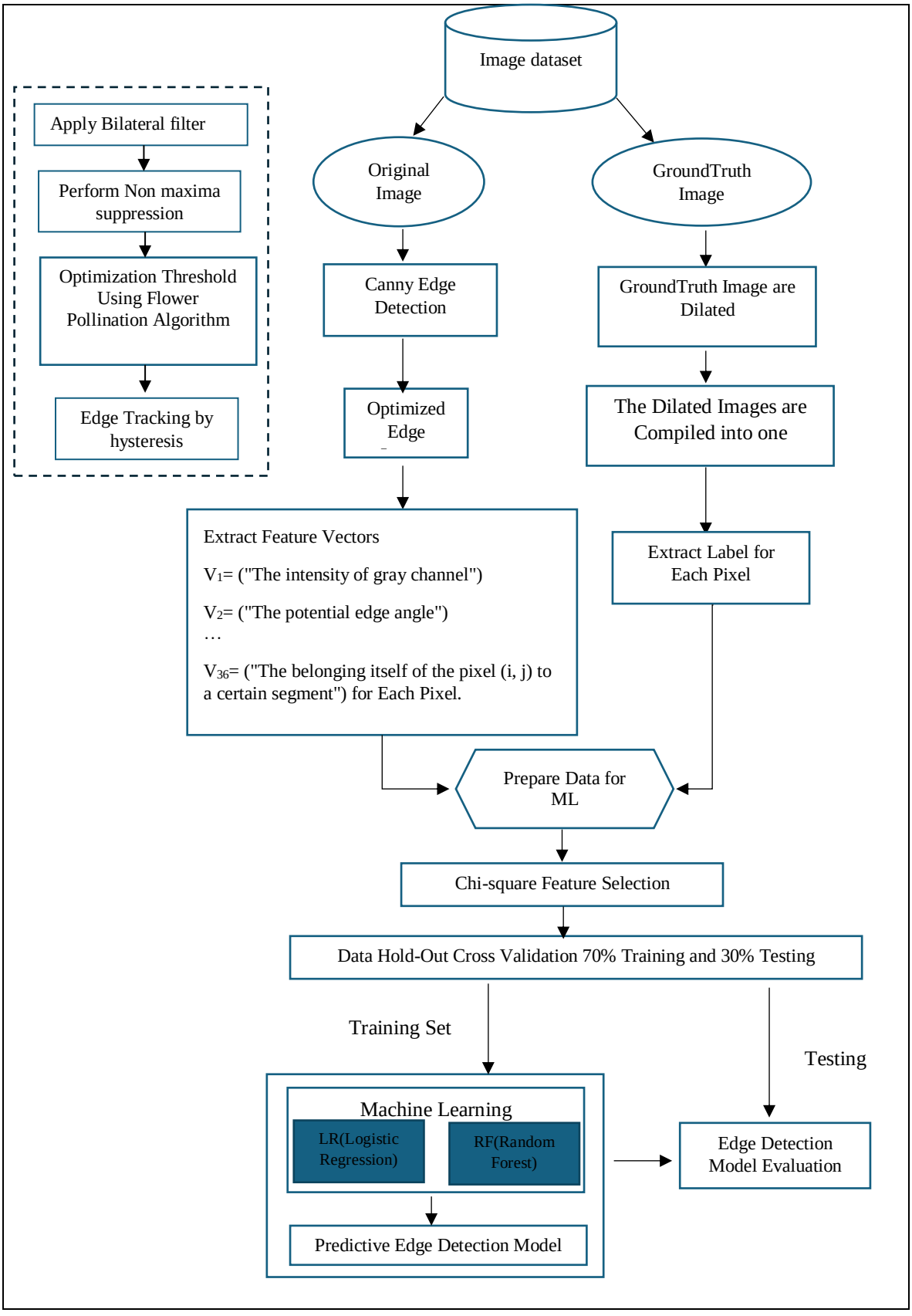

Figure 1: The optimized canny edge detection methodology flowchart

Table 1: Performance analysis

Classifier/Algorithm	AUC	Accuracy	Recall	Precision	F-measure
Random forest	0.81	93.17	10.13	95.86	18.32
Logistic	0.75	85.94	31.82	21.94	25.97
Canny	0.57	84.01	5.26	13.85	7.71
Sobel-grey [8]	0.48				35.6
Neural network [8]	0.55				40.3
Logistic [8]	0.49				38.1
Random forest [8]	0.54				40.0
Ant colony optimization (ACO) [22]					0.81

When comparing these results to the ones presented in [8], it becomes clear that the Random Forest (RF) method outperforms other algorithms like Logistic Regression, Neural Network (N N), and Sobel-grey in terms of AUC and F-measure. Specifically, RF achieves a higher AUC mean (0.552) compared to Logistic Regression (0.524) and Neural Network (0.547). Additionally, the F-measure is higher for RF as well, indicating better overall performance in edge detection. This comparison underscores the effectiveness of combining traditional edge detectors with machine learning techniques for enhanced performance. When comparing the result in [22], to what we obtained using RF**,** we find that AUC = 0.81 in our model reflects good discrimination between true and false edges, out performing the paper that achieved F-measure = 0.81, where the authors used Guided Image Filtering and Enhanced Ant Colony Optimization (ACO)**.** While AUC measures overall discriminatory ability, F-measure indicates the balance between precision and recall**.** Although our results show excellent performance in AUC, improving F-measure will remain a priority to achieve a better balance between precision and recall in the future.

In general, this research addresses the challenges faced by traditional edge detection techniques, such as fixed threshold settings and noise sensitivity. An innovative solution was proposed by using the Flower Pollination Algorithm (FPA) to dynamically set thresholds based on the image's characteristics, which helps improve edge detection accuracy and avoids issues associated with using fixed parameters. Additionally, a Bilateral Filter was applied to reduce noise while preserving edges, which is a key strength of this method compared to traditional approaches that often lead to edge loss during noise removal. The integration of machine learning techniques played a crucial role in enhancing the final results. Machine learning helped improve the system's ability to identify fine edges and discard irrelevant or misleading data. While techniques like Canny remain effective, the proposed method in this research offers significant improvements by providing integrated solutions to problems faced by traditional methods, making it more flexible and accurate when handling images with varying characteristics. One limitation identified in this study is the time required for training the model on the image dataset. Because of the difficulty of the model and the size of the dataset, training takes a considerable amount of time, which can affect the efficiency of the development cycle. Also, the strategy performance in real-time use has not been fully tested, which is an area for future study. Figure 2 illustrates an image through different classifiers. Subfigures 2 A shows the original image, 2 B represents the ground truth image, 2 C presents the results of the traditional Canny edge detection method, 2 D shows the results after applying the optimized Canny method, 2 E displays the edge detection results using Logistic Regression, and 2 F illustrates the output from the Random Forest method, which produces the highest-quality edges, followed closely by Logistic Regression, both of which show strong edge continuity.

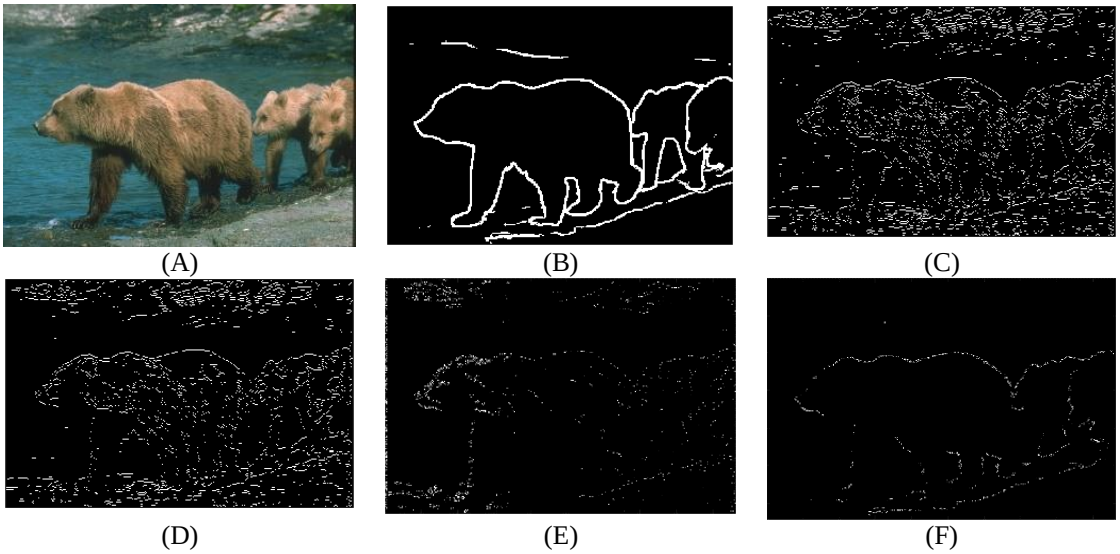

Figure 2: Illustrates edge detection results using different classifiers: (A) Original image, (B) Ground truth, (C) Traditional canny, (D) Optimized canny, (E) Logistic regression, and (F) Random forest

6. Conclusion

The suggested strategy has generated innovative and promising outcomes, motivating further research. Our subsequent processing strategy, utilizing machine learning techniques, demonstrated a significant enhancement in edge detection outputs when using the Canny algorithm enhanced by the Flower Pollination Algorithm (FPA). The edges obtained by our enhanced Canny method were more suitable than those extracted using classical edge detection techniques.

7. Future Work

Future research could explore edge detection problems using alternative algorithms, each requiring different predictor variables tailored to their specific methods. Integrating the Flower Pollination Algorithm (FPA) with other algorithms may lead to further improvements in edge detection. Additionally, applying deep learning models could offer even greater enhancement in edge detection accuracy. Another promising direction would be the implementation of the Hough transform to improve edge quality, especially for effective line detection.

Author contributions

Conceptualization, **R. Lafta** and **Z. Sultani** ; data curation, **R. Lafta**; formal analysis, **R. Lafta**; investigation, **R. Lafta**; methodology, **R. Lafta**; project administration, **Z. Sultani**; resources, **R. Lafta**; software, **R. Lafta**; supervision, **Z. Sultani**; validation, **Z. Sultani** and **R. Lafta**; visualization, **R. Lafta**; writing—original draft preparation, **R. Lafta**; writing—review and editing, **Z. Sultani**. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

[1] C. Lopez-Molina, M. Galar, H. Bustince, B. De Baets, On the impact of anisotropic diffusion on edge detection, Pattern Recognit., 47 (2014) 270–281. https://doi.org/10.1016/j.patcog.2013.07.009

[2] O. Elharrouss, Y. Hmamouche, A. K. Idrissi, B. El Khamlichi, A. El Fallah-Seghrouchni, Refined edge detection with cascaded and high-resolution convolutional network, Pattern Recognit., 138 (2023) 109361. https://doi.org/10.1016/j.patcog.2023.109361

[3] Ş. Öztürk and B. Akdemir, Comparison of edge detection algorithms for texture analysis on glass production, Procedia-Social Behav. Sci., 195 (2015) 2675–2682. https://doi.org/10.1016/j.sbspro.2015.06.477

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester; D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2009) 1627–1645. https://doi.org/10.1109/TPAMI.2009.167

[5] C. L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, Proceedings, Part V, Springer, (2014) 391–405. https://doi.org/10.1007/978-3-319-10602-1_26

[6] N. R. Pal, S. K. Pal, A review on image segmentation techniques, Pattern Recognit., 26 (1993) 1277–1294. https://doi.org/10.1016/0031-3203(93)90135-J

[7] R. Sun, T. Lei, Q. Chen, Z. Wang, X. Du, et al., Survey of image edge detection, Front. Signal Process., 2 (2022) 826967. https://doi.org/10.3389/frsip.2022.826967

[8] P. Flores-Vidal, J. Castro, D. Gomez, Postprocessing of edge detection algorithms with machine learning techniques, Math. Probl. Eng., 2022 (2022) 9729343. https://doi.org/10.1155/2022/9729343

[9] V. K. Ha, J. Ren, X. Xu, S. Zhao, G. Xie, V. M. Vorgas, Deep learning-based single image super-resolution: A survey, in: Advances in Brain-Inspired Cognitive Systems: 9th International Conference, BICS 2018, Xi'an, China, Springer, 2018. https://doi.org/10.1007/978-3-030-00563-4_11

[10] Z.-W. He, L. Zhang, F.-Y. Liu, DiscoStyle: Multi-level logistic ranking for personalized image style preference inference, Int. J. Autom. Comput., 17 (2020) 637–651. https://doi.org/10.1007/s11633-020-1244-1

[11] Y.-H. Wu, Y. Liu, L. Zhang, M-M. Cheng, EDN: Salient object detection via extremely-downsampled network, IEEE Trans. Image Process., 31 (2022) 3125–3136. https://doi.org/10.1109/TIP.2022.3164550

[12] C. Zhao, Y. Hao, S. Sui, S. Sui, A new method to detect the license plate in dynamic scene, in: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), IEEE, 2018. https://doi.org/10.1109/DDCLS.2018.8516012

[13] L. Gou, H. Li, H. Zheng, H. Li, X. Pei, Aeroengine control system sensor fault diagnosis based on CWT and CNN, Math. Probl. Eng., 2020 (2020) 5357164. https://doi.org/10.1155/2020/5357146

[14] Jingzhong, H., et al., Strip steel surface defects recognition based on SOCP optimized multiple kernel RVM, Math. Probl. Eng., 2018 (2018) 9298017. https://doi.org/10.1155/2018/9298017

[15] V. Maksimovic, M. Petrovic, D. Savic, B. Jksic, P. Spalevic,. New approach of estimating edge detection threshold and application of adaptive detector depending on image complexity, Optik, 238 (2021) 166476. https://doi.org/10.1016/j.ijleo.2021.166476

[16] H. Yamagiwa, Y. Takase, H. Kambe, R. Nakamoto, Zero-shot edge detection with SCESAME: Spectral clustering-based ensemble for segment anything model estimation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024. https://doi.org/10.48550/arXiv.2308.13779

[17] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell., 6 (1986) 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

[18] C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, 6th International Conference on Computer Vision (IEEE Cat. No. 98CH36271), IEEE, 1998. https://doi.org/10.1109/ICCV.1998.710815

[19] Yang, X.-S., Flower Pollination Algorithm for Global Optimization, Lecture Notes in Computer Science, Springer, 7445, 2012. https://doi.org/10.1007/978-3-642-32894-7_27

[20] D. Martin, C.Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV 2001), IEEE, 2001. https://doi.org/10.1109/ICCV.2001.937655

[21] L. Breiman, Random forests, Mach. Learn., 45 (2001) 5–32. https://doi.org/10.1023/A:1010933404324

[22] A. Kumar, S. Raheja, Edge detection using guided image filtering and enhanced ant colony optimization, Procedia Comput. Sci., 173 (2020) 8-17. https://doi.org/10.1016/j.procs.2020.06.003