Volume 10, Issue 12, December – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

FPGA Implementation of Edge Detection Algorithms for Image Processing

Thatipally Sharanya[1]; M. Sampath[2]

[1]Department of Electronics and Communication Jawaharlal Nehru Technological University Hyderabad (JNTUH) Hyderabad, India
[2]Assistant Professor; JNTUH UCESTH

Publication Date: 2026/01/03

Abstract: In the field of image processing, edge detection plays a crucial role in extracting meaningful structural information from visual data. It is widely used in applications such as object recognition, feature extraction, and motion analysis in computer vision. Among various edge detection techniques, the Canny edge detector is recognized for its optimal performance in detecting true edges while minimizing false detections. This work presents a comparative study of edge detection methods with a special focus on the Canny algorithm, highlighting its efficiency and precision over conventional techniques. To enhance its performance on blurred and noisy images, a Median filter is employed as a preprocessing step. The Median filter effectively reduces noise while preserving edges, resulting in a more accurate and robust edge detection pipeline. The proposed improvement demonstrates superior edge preservation in challenging visual conditions, validating the effectiveness of integrating edge-preserving noise reduction with traditional Canny edge detection.

Keywords: *Canny Edge Detector, Median Filtering, FPGA-Based Designs, Image Denoising.*

How to Cite: Thatipally Sharanya; M. Sampath (2025) FPGA Implementation of Edge Detection Algorithms for Image Processing. *International Journal of Innovative Science and Research Technology*, 10(12), 2281-2287. https://doi.org/10.38124/ijisrt/25dec1562

I. INTRODUCTION

Edge detection is a fundamental task in image processing, as it plays a critical role in identifying object boundaries and structural details within an image. These extracted edges serve as essential inputs for higher-level applications such as image segmentation, object recognition, medical imaging, surveillance systems, and autonomous navigation. The effectiveness of these applications strongly depends on the accuracy and reliability of the edge detection process, especially when images are affected by noise or blur.

In practical scenarios, digital images are often corrupted by noise and blur introduced during image acquisition, transmission, or environmental conditions. Conventional edge detection techniques, including Sobel and Prewitt operators, are simple and computationally efficient but are highly sensitive to noise and blur, leading to broken or false edges. Among advanced methods, the Canny edge detector is widely regarded as one of the most robust algorithms due to its optimal detection, precise localization, and minimal false responses.

The Canny algorithm achieves its performance through a multi-stage process that includes noise reduction, gradient computation, non-maximum suppression, and hysteresis thresholding. However, the traditional implementation relies on Gaussian filtering for noise suppression. While Gaussian smoothing effectively reduces high-frequency noise, it also introduces edge blurring, which results in the loss of fine details and weak edge information. This limitation becomes more prominent in images affected by impulse noise or blur.

To address this issue, this work proposes an enhanced Canny edge detection approach by incorporating a Median filter in the preprocessing stage. The Median filter is a non-linear, edge-preserving filter that is particularly effective in removing impulse noise while maintaining sharp intensity transitions. By replacing Gaussian smoothing with Median filtering, the proposed method improves noise suppression without compromising edge clarity.

The proposed approach is validated through MATLAB-based simulations using multiple noisy and blurred images. Furthermore, the algorithm is implemented on FPGA using Verilog HDL and Xilinx Vivado to evaluate real-time performance and hardware feasibility. This combination of software modeling and hardware realization ensures that the proposed system is accurate, efficient, and suitable for real-world embedded vision applications.

- *Objective*

The main objective of this work is to develop an enhanced edge detection system by improving the noise and blur suppression capability of the conventional Canny edge detector. The proposed approach focuses on integrating a

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

Median filter into the preprocessing stage to effectively reduce noise and blur while preserving important edge information. The system aims to produce clearer and more continuous edges, even in images affected by impulse noise and blur. The methodology is designed to be efficient and suitable for real-time implementation, with validation carried out through MATLAB simulations and FPGA-based hardware realization using Verilog HDL and Xilinx Vivado. Ultimately, the proposed system seeks to improve the reliability and practical applicability of edge detection in real-world image processing applications.

II. LITERATURE REVIEW

Image processing and edge detection have been widely explored due to their importance in extracting structural information from digital images. Early edge detection techniques such as Sobel and Prewitt operators relied on first-order gradient calculations to detect intensity variations. Although these methods were simple and computationally efficient, they were highly sensitive to noise, blur and often produced broken or false edges when applied to real-world images.

John F. Canny introduced a mathematically optimal edge detection algorithm that significantly improved detection accuracy through a multi-stage process involving noise reduction, gradient computation, non-maximum suppression, and hysteresis thresholding. The Canny edge detector became a benchmark due to its good localization and reduced false responses. However, the algorithm employs Gaussian smoothing, which leads to edge blurring and loss of fine details, particularly in images affected by impulse noise.

Rafael C. Gonzalez and Richard E. Woods analysed various image filtering techniques and highlighted that linear filters such as Gaussian filters are effective in suppressing high-frequency noise but tend to smooth sharp edges. Their study emphasized the trade-off between noise reduction and edge preservation, which remains a critical challenge in edge detection applications. Similarly, Wiener filtering techniques were explored for image restoration, showing strong performance under Gaussian noise but limited effectiveness for non-Gaussian noise conditions.

To overcome these limitations, Tukey and subsequent researchers investigated non-linear filtering approaches. Their studies demonstrated that Median filtering is highly effective in removing salt-and-pepper noise while preserving edge sharpness. Comparative analyses between Median and Gaussian filters showed that Median filtering produces improved edge continuity and reduced distortion, making it more suitable for preprocessing in edge detection algorithms.

With the growing demand for real-time image processing, C.-H. Huang explored FPGA-based hardware implementations of image processing algorithms and highlighted the advantages of parallelism, low latency, and energy efficiency offered by FPGA platforms. The study demonstrated that FPGA implementations using hardware description languages such as Verilog significantly outperform software-only approaches, making them suitable for embedded and real-time vision systems.

Several researchers have adopted MATLAB as a simulation and validation platform prior to hardware realization. MATLAB-based studies enabled algorithm verification, noise analysis, and parameter optimization before translating the design into hardware. Following validation, edge detection algorithms were implemented using Verilog HDL and synthesized using FPGA design tools to achieve real-time performance.

Although significant advancements have been made in edge detection and hardware acceleration, many existing approaches still rely on conventional Gaussian smoothing, which compromises edge quality. Integrating edge-preserving filters such as the Median filter with classical edge detection algorithms and implementing them on FPGA platforms offers an effective solution. These findings motivate the proposed work, which focuses on the FPGA implementation of enhanced edge detection algorithms for image processing applications.

III. TRADITIONAL PRACTICES

Edge detection has long been a fundamental task in image processing, as it plays a crucial role in identifying object boundaries and structural details within images. Traditionally, edge detection and image analysis were performed using simple software-based or manual methods. While these approaches were effective during early stages of image processing research, they often lacked robustness, accuracy, and real-time capability, particularly when applied to noisy or large-scale image data. The following sections describe commonly used traditional practices and their limitations.

- *Manual and Visual Analysis*

In the early stages of image processing, edge information was often analyzed visually by observing grayscale intensity variations and image gradients. Researchers relied on manual inspection of images to interpret boundaries and structural features. Although this method required minimal computational resources, it was highly subjective and dependent on human perception. Visual analysis is time-consuming, inconsistent, and impractical for large image datasets, making it unsuitable for automated or real-time applications.

- *Software-Based Edge Detection*

Traditional edge detection techniques such as Sobel, Prewitt, and Roberts operators were widely used due to their simplicity and ease of implementation. These methods detect edges by computing first-order intensity gradients. While computationally efficient, they are highly sensitive to noise, blur and often generate false or broken edges in degraded images. Their performance significantly degrades under low contrast or noisy conditions, limiting their reliability in practical applications.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

➢ *Gaussian-Based Noise Reduction*

To improve edge detection performance, Gaussian filtering was commonly employed as a preprocessing step to suppress noise. Gaussian smoothing effectively reduces high-frequency noise but introduces edge blurring, resulting in the loss of fine details and weak edges. This trade-off between noise suppression and edge preservation remains a major drawback of traditional Gaussian-based edge detection approaches.

➢ *Canny Edge Detection*

The Canny edge detection algorithm represented a significant advancement over basic gradient methods by introducing a multi-stage framework that includes noise reduction, gradient computation, non-maximum suppression, and hysteresis thresholding. Although Canny provides better localization and reduced false detections, its dependence on Gaussian smoothing limits its effectiveness in images affected by impulse noise or severe degradation.

➢ *Software-Only Processing*

Most traditional edge detection systems were implemented purely in software using platforms such as MATLAB or general-purpose processors. While software-based processing allows flexibility and ease of development, it suffers from high computational latency and limited real-time performance. As image resolution and data volume increase, software-only implementations become inefficient for time-critical applications such as surveillance, robotics, and medical imaging.

➢ *Limitations of Traditional Practices*

Although traditional edge detection techniques laid the foundation for modern image processing, they are constrained by blur and noise sensitivity, lack of adaptability, and limited processing speed. These limitations highlight the need for improved edge-preserving techniques and hardware-accelerated solutions. Integrating advanced filtering methods with FPGA-based implementations offers a promising approach to overcoming these challenges and achieving real-time, high-performance edge detection.

IV. PROPOSED METHODOLOGY

The proposed edge detection methodology was implemented using a combination of MATLAB and Xilinx Vivado tools to achieve both algorithmic accuracy and hardware-level realization. MATLAB was used as the primary platform for algorithm development, testing, and visualization, while Xilinx Vivado enabled translation of the validated design into FPGA-compatible hardware logic. This dual-stage approach ensured correctness at the software level and feasibility at the hardware level.

The methodology focuses on improving edge detection performance in noisy and degraded images by integrating Median filtering with the Canny edge detection algorithm. The complete workflow includes degraded image generation, preprocessing, edge detection, hardware modelling, simulation, and FPGA execution.

➢ *Input Image Selection and Degradation Modelling*

In the initial stage, a grayscale image of fixed resolution 256 × 256 pixels was selected as the input. This resolution was chosen to maintain uniformity between MATLAB simulation and FPGA-based hardware implementation. To realistically simulate real-world image degradation, the input image was first subjected to Gaussian blurring in MATLAB to model optical defocus and motion blur effects.

Following Gaussian blurring, salt-and-pepper noise was added to the image to represent impulse noise caused by sensor imperfections and transmission errors. The combined degradation ensured that the test image reflected challenging conditions commonly encountered in practical imaging systems.

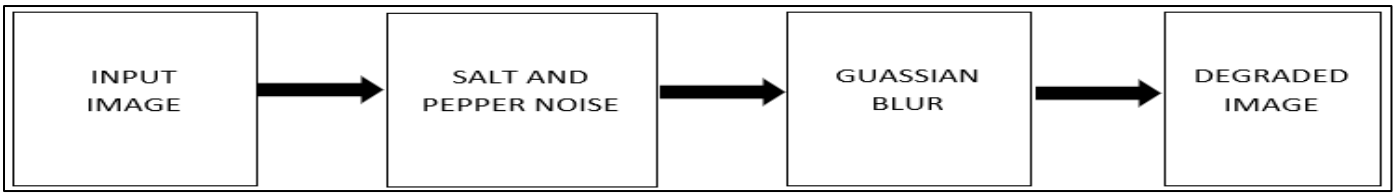

Fig 1 Process of Degraded Image Generation

➢ *Data Conversion for Hardware Compatibility*

After degradation, the noisy image was converted into a hardware-compatible format. Pixel intensity values were extracted and converted into hexadecimal representation using MATLAB. These values were stored in a text-based memory initialization file, enabling consistent use of the same input image across MATLAB verification, Verilog simulation, and FPGA execution.

This conversion ensured identical input conditions during software validation and hardware processing, allowing accurate comparison between MATLAB results and FPGA outputs.

➢ *Median Filter–Based Preprocessing*

Preprocessing was carried out using a Median filter, which served as the first stage of the proposed edge detection pipeline. The Median filter was implemented and tested in MATLAB using customized routines. Filter parameters were tuned to effectively suppress impulse noise while preserving important edge structures.

Unlike linear filters, the Median filter replaces each pixel with the median value of its neighbourhood, making it highly effective for removing salt-and-pepper noise without introducing edge blurring. This preprocessing stage significantly improved the reliability of subsequent edge detection operations.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

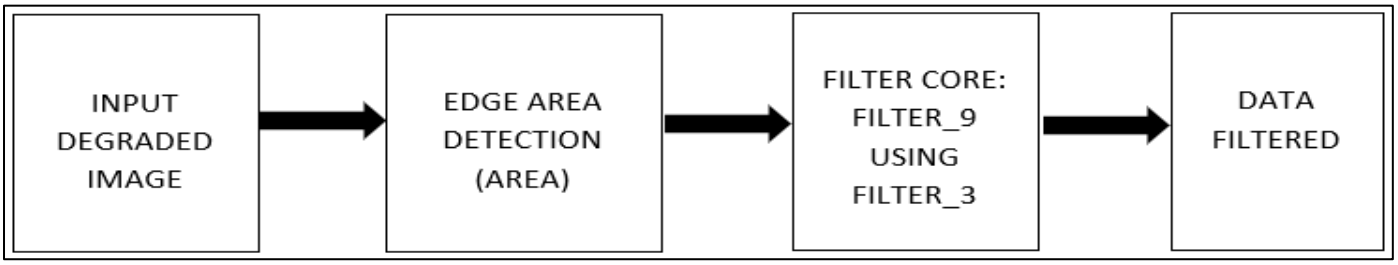

Fig 2 Internal Processing Flow of Median Filter

➢ *Canny Edge Detection Implementation*

Following noise suppression, the Canny edge detection algorithm was applied. MATLAB's built-in Canny function was initially used as a reference, while a custom implementation provided full control over individual processing stages.

Gradient computation was performed using Sobel operators to calculate horizontal and vertical intensity changes. The resulting gradient magnitude and direction were used to identify potential edge locations. Non-maximum suppression was then applied to retain only local maxima along the gradient direction, producing thin and precise edges.

Finally, hysteresis thresholding was performed using two thresholds to differentiate between strong and weak edges. Weak edges were retained only if they were connected to strong edges, reducing false detections and improving edge continuity.

Comparative analysis showed that Median-based preprocessing produced more continuous and accurate edges than conventional Gaussian-based Canny detection, particularly in noisy images.

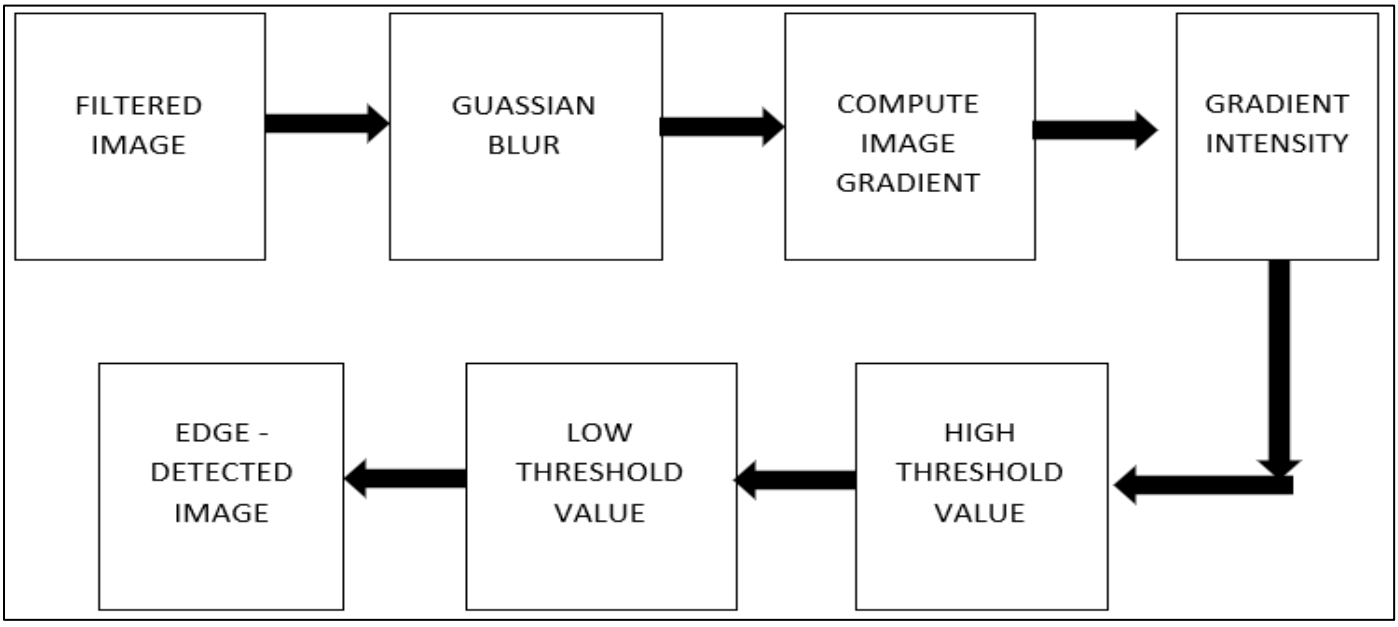

Fig 3 Processing Stages of Canny Edge Detection Algorithm

➢ *System Architecture*

The overall system architecture consists of sequential processing stages including image input, Median filtering, gradient computation, non-maximum suppression, hysteresis thresholding, and edge output generation. Each stage was verified independently in MATLAB before being translated into hardware logic.

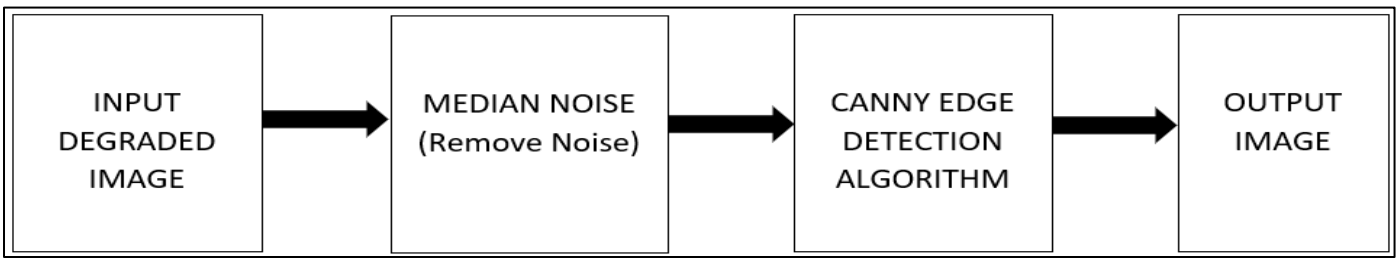

Fig 4 Top-Level Architecture of the Proposed Design

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

➢ *FPGA Hardware Implementation*

After software verification, the algorithm was implemented in hardware using Verilog HDL. The design was developed and synthesized using Xilinx Vivado. Each processing block, including Median filtering, Sobel gradient computation, edge magnitude estimation, and thresholding, was implemented as a separate hardware module.

To reduce computational complexity, fixed-point arithmetic was used throughout the design. The median filter was implemented using a 3×3 sliding window, realized with line buffers and shift registers to minimize memory usage and latency.

Gradient magnitude calculation avoided costly square-root operations by using absolute-value approximations, which provided sufficient accuracy while conserving FPGA resources.

➢ *Hardware Simulation and Verification*

Simulation of the Verilog design was performed using Vivado's testbench environment. Input images stored in memory files were loaded into on-chip memory blocks, and the resulting edge-detected outputs were compared with MATLAB reference results.

Minor differences were observed due to quantization and approximation effects inherent to hardware implementation; however, overall edge quality and detection accuracy closely matched the software model.

➢ *FPGA Execution and Output Reconstruction*

For hardware execution, the input image was stored in on-chip Block RAM (BRAM) using memory initialization files. Pixel data was streamed sequentially through the processing pipeline, enabling full-frame image processing directly on the FPGA.

After execution, the output pixel values were generated in hexadecimal format and stored in a text file. MATLAB was then used to convert these values back into pixel intensities and reconstruct the final edge-detected image for visualization and comparison.

V. RESULTS AND DISCUSSION

This section presents and analyses the experimental results obtained from MATLAB simulation and FPGA-based implementation of the proposed edge detection methodology. The performance of the system is evaluated by observing the visual quality of edge detection at different stages of processing, including image degradation, noise suppression, and final edge extraction.

Fig. (a) represents the original grayscale input image used for experimentation. This image serves as the reference for evaluating the impact of noise addition and subsequent processing stages. The original image contains clear structural boundaries and intensity transitions, which are essential for accurate edge detection.

To simulate real-world imaging conditions, the original image was intentionally degraded, as shown in Fig. (b). Gaussian blurring was applied to model optical defocus and motion blur, followed by the addition of salt-and-pepper noise to represent impulse noise caused by sensor imperfections and transmission errors. The degraded image exhibits significant loss of clarity, reduced contrast, and noisy pixel distributions, making direct edge detection highly challenging.

Fig. (c) shows the output obtained after applying the Median filter to the degraded image. The Median filter effectively suppresses impulse noise while preserving important image structures. Compared to the degraded image, the Median-filtered output demonstrates noticeable noise reduction without excessive smoothing. Unlike linear filtering techniques, the Median filter maintains sharp intensity transitions, ensuring that essential edge information is retained for further processing.

The final edge detection result obtained using the Canny edge detection algorithm is illustrated in Fig. (d). After Median-based preprocessing, the Canny algorithm successfully extracts clear and continuous edges from the image. The detected edges accurately correspond to object boundaries present in the original image, with minimal false edge detection. The combination of edge-preserving noise suppression and multi-stage edge detection results in improved edge localization and continuity, even under noisy conditions.

A comparative analysis of the results confirms that Median filtering significantly enhances the performance of the Canny edge detector. Without proper preprocessing, noise and blur lead to fragmented and false edges; however, the proposed Median-enhanced approach produces cleaner and more reliable edge maps. Minor variations observed between MATLAB and FPGA outputs are attributed to fixed-point arithmetic and hardware approximations, but these differences do not affect overall edge quality.

Overall, the experimental results validate the effectiveness of the proposed methodology in handling noisy and degraded images. The integration of Median filtering with Canny edge detection, followed by FPGA implementation, ensures both algorithmic robustness and real-time feasibility. This makes the proposed system suitable for practical image processing applications where accuracy, speed, and hardware efficiency are critical.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

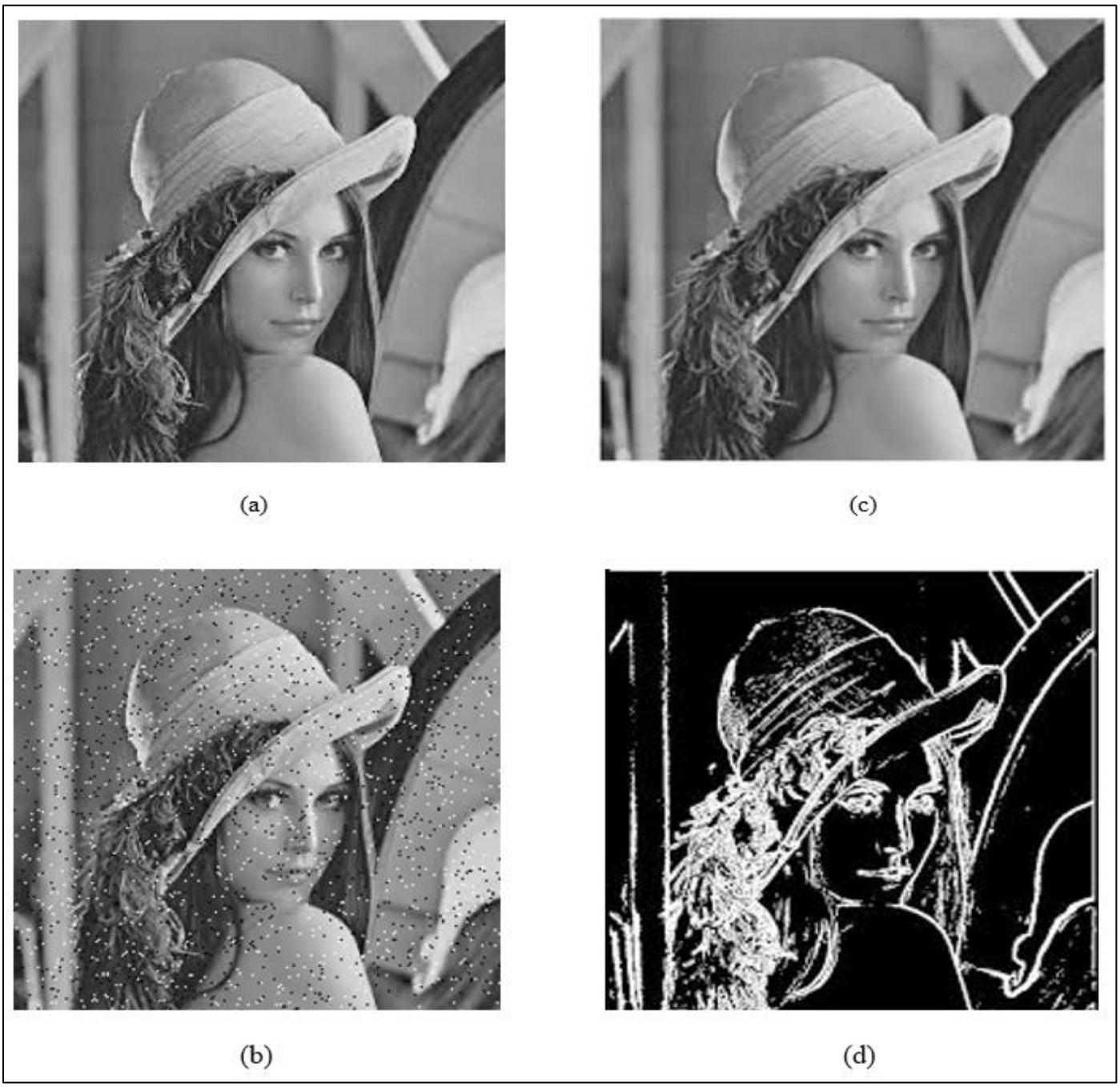

Fig 5 Final Results of the Proposed Algorithm: FPGA Implementation of Edge Detection Algorithms for Image Processing: (a) Original Grayscale Image, (b) Image Degraded by Blur and Salt-and-Pepper Noise, (c) Output after Median Filtering, (d) Final Edge-Detected Image Using the Canny Algorithm.

VI. CONCLUSION

In conclusion, this study highlights the importance of edge detection as a fundamental operation in image processing and demonstrates the effectiveness of enhancing the Canny edge detection algorithm through Median filtering. The experimental results confirm that the incorporation of a Median filter significantly improves edge preservation by effectively suppressing noise and blur while maintaining important image structures. Unlike conventional Gaussian smoothing, which often leads to edge blurring, the Median-based approach retains sharp boundaries and reduces false edge detection in noisy and blurred images.

The proposed Median-enhanced Canny edge detection framework delivers more accurate and continuous edge maps, making it well suited for real-world image processing scenarios where noise and blur are unavoidable. MATLAB-based simulations validated the correctness and performance of the algorithm, while FPGA implementation using Verilog HDL and Xilinx Vivado demonstrated its feasibility for real-time hardware deployment. The hardware results closely matched software outputs, with only minor differences due to quantization effects, confirming the reliability of the design.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec1562

Overall, the integration of edge-preserving preprocessing with hardware acceleration strengthens the robustness and efficiency of traditional edge detection techniques. The proposed approach bridges the gap between algorithmic accuracy and practical implementation, making it suitable for embedded vision, surveillance, robotics, and industrial image processing applications.

REFERENCES

[1]. Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.
[2]. Gota, A., & Min, Z. J. (2013). Analysis and Comparison on Image Restoration Algorithms Using MATLAB. International Journal of Engineering Research & Technology (IJERT) Vol, 2, 1350-1360.
[3]. Mahalakshmi, A.,& Shanthini, B. (2016, January). A survey on image deblurring. In 2016 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.
[4]. Flusser, J., Farokhi, S., Hoschl, C., Suk, T., Zitov ¨ a, B., ´ & Pedone, M. (2015). Recognition of images degraded by Gaussian blur. IEEE transactions on Image Processing, 25(2), 790-806.
[5]. Ramya, S.,& Christial, T. M. (2011, March). Restoration of blurred images using Blind Deconvolution Algorithm. In 2011 International Conference on Emerging Trends in Electrical and Computer Technology (pp. 496-499). IEEE.
[6]. Sada, M. M., & Mahesh, M. G. (2018). Image deblurring techniques—a detail review. Int. J. Sci. Res. Sci. Eng. Technol, 4(2), 15.
[7]. Verma, R., & Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. International Journal of advanced research in computer science and software engineering, 3(10).
[8]. Syahrian, N. M., Risma, P., & Dewi, T. (2017). Vision-based pipe monitoring robot for crack detection using canny edge detection method as an image processing technique. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 243-250.
[9]. Sekehravani, E. A., Babulak, E., & Masoodi, M. (2020). Implementing canny edge detection algorithm for noisy image. Bulletin of Electrical Engineering and Informatics, 9(4), 1404-1410.
[10]. Yadav, S., Jain, C., &Chugh, A. (2016). Evaluation of image deblurring techniques. International Journal of Computer Applications, 139(12), 32- 36.