Optimized Median Filtering Algorithm for Image Denoising

1st Anbumani V
Electronics and Communication Engineering
Kongu Engineering College
Perundurai, Erode, India
anbumanivenkat@gmail.com

4th Abirami S
Electronics and Communication Engineering
Kongu Engineering College
Perundurai, Erode, India
abiramis.21ece@kongu.edu

5th Abinaya S
Electronics and Communication Engineering
Kongu Engineering College
Perundurai, Erode, India
abinayas.21ece@kongu.edu

6th Archaya S
Electronics and Communication Engineering
Kongu Engineering College
Perundurai, Erode, India
archayas.21ece@kongu.edu

***Abstract*—Image denoising is essential in image processing domain, particularly in real-time applications where noise, such as salt-and-pepper noise, deteriorate the quality of an image very badly. The conventional median filter architectures are often challenged by the computational complexity, which limits their suitability for efficient real-time implementation. In this research, it can use ultra-low resource and multi-stage count on the basis of median select as well as specification precedence. Such an implementation is then proposed as an implementation of a median filter designed to work on a 3x3 pixel window and using a comparator–based logic in order to identify the median value with a minimal number of comparisons. The design minimizes the number of comparisons and comparators, which reduces critical path delay and increases performance. On the other side, the proposed architecture overcomes the shortcomings of traditional architectures. The results demonstrated notable improvements in noise reduction and image quality, confirming that the optimized design outperforms traditional architectures. The reduction in comparators and processing stages ensures enhanced efficiency and optimal performance in real-time image processing applications.**

***Index Terms*—Median filter, Image processing, Impulse noise, salt and pepper noise, comparator based design, noise reduction and hardware optimization.**

I. Introduction

One of the most common process in the image processing is that of the image filtering which is used for removing noise from the image to give a good quality for the further analysis and interpretation of the image. There is a range of filters that are widely used for the purpose of noise suppression, but should still keep a large percentage of those features of an image, like edges and smaller details, should not be too sensitive to noise sensitivity. This involves the median filtering which is alleged to work well [1] in this kind of impulse noise as salt-and-pepper and also to restore the acuteness of the picture. Median filters are a simple yet effective technique to remove noise in an image. This approach is extremely effective at reducing noise while maintaining the edges of the boundaries without losing any sharpness. There are plenty of software implementations available but they still have scalability issues whenever it comes to real time processing with use cases such as high resolution images. This project is based on designing and implementing an efficient image process architecture which has a high throughput for real-time applications and has low area and power. Moreover, the median is calculated through efficient optimized comparators, thus minimizing components and ensuring fast execution. Real-time median filtering can be achieved using parallel processing and optimized logic design, making it usable for applications like medical imaging, video surveillance, and autonomous vehicle navigation. However, when processing high resolution image streams, there is a significant practical advantage to using special-purpose hardware, including for median filtering algorithms. Advanced VLSI (Very Large Scale Integration) design methodologies can be applied to reduce latency, increase the degree of parallelism, and enhance overall performance [2]. However, it is suitable for use in embedded systems and portable devices as it is internally optimized hardware blocks to ensure fast computations and low power consumption. Because of the architectural optimizations that ensure scalability, the design can also handle various picture sizes and resolutions without requiring significant reconfiguration. The proposed median filter provides performance and functionality in applications with stringent deployment conditions and thus features as a robust implementation alternative for a wide variety of real-time image processing systems, due to these innovations. As a result, the proposed median filter design can

be considered as a flexible solution for various real-time image processing applications with reliable and efficient operation even in difficult operating environments.

II. Literature Survey

Median filters have been around a long time in image processing and are one of the best tools for removing impulse noise such as salt-and-pepper noise. Broader, they are still acting by replacing the central pixel of the window with the median value of the surrounding pixels. Despite being effective for noise cancellation, this method is often incapable of meeting the demands of real-time applications due to its computational and hardware complexity. Several optimizations have been developed over time to improve the performance of median filters. Many of them, especially comparator-based designs have been popular as they help to minimize the delays and conserve hardware resources by going down with the number of comparisons with the exploration of hierarchical comparator networks.

However, these designs often hit obstacles such as too many connections, complicating routing and increasing power consumption and chip area needs. These designs, however, often encounter problems such as excessive connections that make routing difficult and result in greater power consumption and chip space requirements. Recent advances have mainly concentrated on the design of scalable architectures tailored for FPGA implementations. These architectures attempt to gain a trade-off between computing befits and resource usage by minimizing the comparator levels. Moreover, developments show adaptive filtering methods to enable the filtering parameters to adapt to the levels of noise in the input image. The filter does well in a variety of noise scenarios because of its versatility. However, many of these systems still depend on many comparators, making them more complex.

The work improves upon the median filter of the previous work by reducing the number of comparators from 19 to 18 through a substitute comparator network and interconnection optimizations [3]. This novel technique overcomes key disadvantages faced by traditional methods to offer a new real-time image processing solution that is much more interesting. This suggested architecture is a good choice for hardware-limited applications because it can achieve significant improvements in performance and in power and area utilization. The proposed architecture tries to do a trade-off between Hardware complexity and computing performance to address these limitations. The reduced number of comparators used for the design leads to a much more compact and efficient arrangement with simpler interconnections and reduced chip area requirement. This advantage makes the architecture ideal for embedded or portable devices as it consumes less power and provides faster computation. Moreover, the architecture ensures that filtering precision is preserved, which is particularly important for real-time applications where reliable performance relies on efficient noise suppression, such as autonomous navigation, medical imaging, and video surveillance.

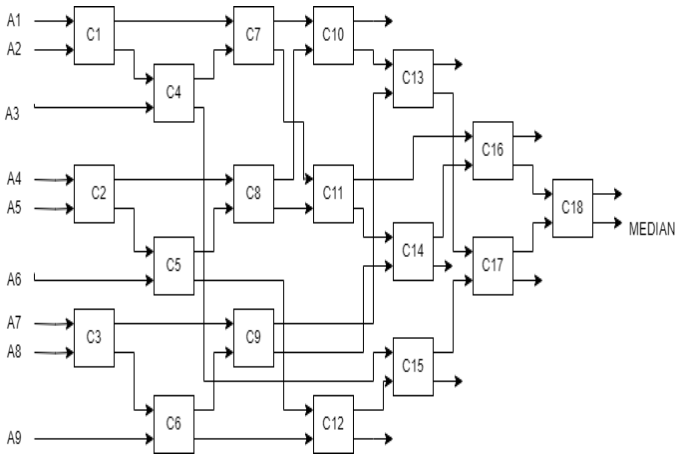

Fig. 1. Proposed Median Sorting Architecture

III. Proposed Work

The desired architecture is to eliminate inefficiencies in the computation time and hardware resource usage with similar performance levels in terms of median calculation speed and accuracy. Based on the same reference of the median filter, this proposed method aims to bring out the architecture that reduces the overall complexity of the comparator network by minimizing the number of stages and comparator. This is essentially aimed at optimizing the objective of calculating the median as quickly and accurately as possible in several applications involving image processing that focus on noise reduction. The simplified median filter architecture consists of a systematic arrangement of comparator (denoted as C1 to C18), as shown in Figure 1. The input data (A1 to A9) are processed in parallel through a series of comparators. The design efficiently organizes the comparator stages to ensure that the output is the median of the nine input values.

In the existing design, the median filter uses a series of comparator to determine the median of a 3x3 pixel window [4]. However, the design has several drawbacks: The number of comparator (19 comparators) introduces significant delays, The hierarchical connections among comparators contribute to a larger area and power consumption. To address these issues, the proposed method uses a simplified comparator architecture. It reduces the number of comparators from 19 to 18 by streamlining the comparison process. The result is that the number of comparators reduces the overall area and timing delay in the design [5]. The use of a simplified architecture further reduces routing complexity for performance improvement.

Primary comparators (C1-C6) take the first set of inputs and start comparing them. C1, C2, and C3 are adjacent pair comparators which compare pairs of adjacent inputs, while comparators C4, C5, and C6 further compare their outputs. Intermediate comparators (C7- C15) makes fewer comparison because intermediate results are now passed through another set of comparators. Again, only the necessary comparisons to be done considered by this design. In the last stage, final comparator C18 computes the median by making the output

from the outputs of comparator stages earlier available to it. With an optimal number of comparators and optimized data flow, it leads to an efficient design. Compared to the previous method, this one enhances overall performance of the median filter. The generation of median element in the 3x3 is achieved using the architecture shown in Figure 4. The proposed architecture introduces several key modifications to the original median filter design to enhance its performance, reduce latency, and minimize complexity.

These changes are crucial in achieving faster computation times while optimizing resource utilization. The primary architectural improvements are detailed below: In the Existing median filter design, 19 comparator were used for comparing pixel values. In the proposed architecture, this count is reduced to 18 due to an efficient arrangement of comparators, that is done to avoid redundant comparison otherwise to the increase in count of unnecessary comparison results. This can reduce the overall comparison levels and propagate the delays. By restructuring, it ensures more streamlined compared to others since this will reduce the overall time taken for comparison. It would contribute towards a smaller hardware and lower power consumption.

The Existing median filter architecture, though it provided a complex interconnection among the comparator and caused some significant routing delays due to increased wiring complexity, was not very efficient since these complex interconnections both slowed up computation and consumed valuable area on the chip. The interconnections are simplified due to the structured and optimized rearrangement of comparator in the architecture proposed [6]. It reduces the area required for wiring, thereby making the design more compact and efficient. Complexity of interconnection is also reduced, and hence the power consumed by routing reduces, which makes this more energy-efficient solution. In the proposed design, the data flowing through the comparator were optimized to make them more structured and linear than in the existing architectures [7]. Data inputs have taken a less structured and perhaps convoluted route in the original architectures, causing unstable timing and increased delay [8].

IV. Propsed Median filtering Algorithm Steps

A. Identify Noisy Pixels

- A pixel is considered noisy if its value is **0 (black)** or **255 (white)**.
- If the pixel is not noisy, keep it unchanged.

B. Extract a Fixed Window (e.g., 3×3)

- Collect the pixel values from the window, excluding noisy pixels (0 or 255).
- If valid pixels exist, compute their **median** and replace the noisy pixel.
- If no valid pixels exist, replace the pixel with the **mean** or last non-noisy pixel.

C. Output

- Denoised image with minimal blurring.

Algorithm 1 Proposed Median Filtering Algorithm (PMFA)

1: **Input:** Noisy image $I(x, y)$
2: **Define four directional neighborhoods:**
3: Horizontal ($\rightarrow$ left-right), Vertical ($\uparrow\downarrow$ top-bottom)
4: Diagonal ($\backslash$ top-left to bottom-right), Diagonal ($/$ top-right to bottom-left)
5: **for** each noisy pixel $I(x, y)$ **do**
6: Extract pixel values along each direction, excluding noisy pixels (0 and 255)
7: Compute the median in each direction
8: Select the median that best preserves edges
9: Replace the noisy pixel with the selected median
10: **end for**
11: **Output:** Denoised image with preserved edges

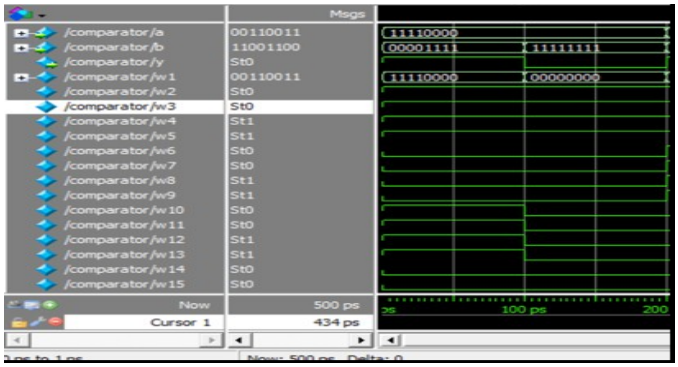

Fig. 2. Simulation Result of Proposed Comparator

V. Software Implementation

The implementation of the efficient median filter design is on optimizing VLSI architecture by block reduction in strategic amounts. Traditional median filtering techniques normally use a higher number of processing blocks in order to achieve noise reduction and commonly result in increases in area and power consumption. Its able to reduce the number of processing comparators from 19 to 18 in this design, which directly eliminates redundant operations and, consequently, simplifies the circuit layout. The entire architecture benefits significantly from reduced redundancy and consolidated operations, making it more efficient in terms of resource utilization and thus more suitable for integration in compact systems where the area and time constraints are critical constraints. Simulation result of the comparator is shown in the figure 2. The output shows reduced execution times with reduced resource usage while maintaining the performance of the median filter. The efficiency gained through comparator and stage reduction is critical in developing high-performance filtering solutions that can be utilized in a number of applications, including imaging systems and real-time data processing environments. These results confirm the practicality of designing an efficient median filter as a viable solution and demonstrating optimizations that enables a compact, area-efficient, and high-speed architecture [9].

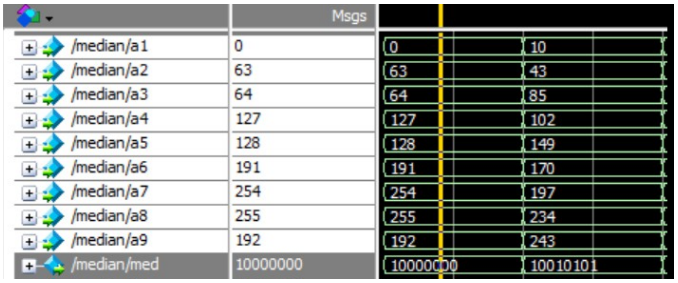

Fig. 3. Simulation Result of Proposed Median Sorting Architecture

The employed comparator allows the median selection algorithm to robustly compare pixel intensity levels, yielding the median. The comparator is based on an existing architecture and significantly decreases the number of comparisons required, thereby improving latency and throughput during the filtering phase. The simulation results show that the comparator is able to handle different levels of noise and accurately detect noise, which enhances the signal quality of the filtered output. Simulation result of the median sorting architecture is illustrated in Figure 3.

The area occupied by the proposed architecture is smaller than the existing design. The area occupied by the proposed architecture is smaller than the existing design. This leads to a smaller footprint on the FPGA because a smaller number of comparators and simpler interconnections are needed. Thus the proposed method will become power efficient and also reduction in area will lead to less power consumption.

The performance of an existing FPGA design is compared with an optimized version in Table 1, where the three main parameters considered are the number of slices, LUTs, and combinational path delay. Notably, the optimized design not only used significantly fewer slices and LUTs but also achieved a shorter combinational path delay. Such improvements presume optimized resource usage and high processing throughput. These enhancements prove beneficial in the realm of FPGA development, where efficient hardware utilization and increased speed correlate with more robust and reliable designs.

Table 1 demonstrates the comparison table of the existing and proposed median filter. It shows a comparison of the resource usage and performance of an existing design vs. an optimized one. For example, the improved design requires 200 slices (compared to 210 for the original design), giving a reduction of only 10% on slices, while the number of LUTs declines to 368 (compared to 395 for the original design).The combinational path delay is reduced from 34.289ns to 32.072ns, which shows improved efficiency and performance of the design.

VI. RESULTS AND DISCUSSION

In the following section, the performance of the proposed median filter design is evaluated using four critical parameters, namely Signal-to-Noise Ratio (SNR), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Mean Squared Error (MSE) [10]. These metrics give a numerical assessment of the filter's effectiveness in suppressing noise cost-ally while maintaining critical features within an image. [2] introduction of the common metrics SNR, PSNR, SSIM, and MSE and their definitions [3] SNR expresses the signal in comparison to disturbances (random noise rate)[4] PSNR indicates overall quality[5] SSIM tells us how two images (original and filtered) have similar organization in the structure[6] MSE denotes the average error, how much one image deviates from the other as a function of the pixel intensity.

Before applying the median filter, the test images have been severely corrupted by salt-and-pepper noise, which is a type of noise readily found in image processing prior to the median filter being used. This noise introduced low SNR and PSNR, indicating low image quality, and low clarity. High MSE values indicate a lot of noise being observed in the image. These metrics are indicative of the need for quality preservation against noise and of a robust filtering method.

After applying the median filter, it shows a significant increase on all performance metrics after applying the median filter. The SNRs showed greater value, indicating a higher strength of the useful signal compared to the noise and thus better clarity of detail retention. The appropriate PSNR values also increase, proving that the filter does a better job of reconstructing the image. Accordingly, SSIM values also increased, suggesting that the structural details of the image have been better preserved an important consideration in applications that require high visual accuracy.

This improvement underlines that the median filter not only effectively reduces noise, but also preserves important image information. In particular, garbage-in garbage-out principle applies in application domains such as medical imaging, surveillance systems, satellite image analysis since the output of the system, that make high-level decision or analysis relies on the validation of the input images [11].

The results, confirmed via detailed simulations and presented in Tables II, underscore the efficiency and robustness of the proposed median filter design [12]. Thus, it has optimized architecture, fewer stages of processing with less computational load, making it possible to process images faster.

The various design improvements provide that the filter can perform on-line in real time while maintaining high levels of image quality and speed. And the filter performs well over the whole range of noise intensities and image degradations, demonstrating the flexibility of our approach.

In the tables presented, the comparisons show that the Proposed Method (PM) consistently outperforms the Existing Method (EM) across SNR, PSNR, SSIM, and MSE metrics, providing a clear case for the efficacy of PM. The findings affirm the rationale underlying the design of the proposed filter, demonstrating its suitability for deployment in tangible image processing frameworks that demand effective noise suppression while retaining high image quality.

Figure 4 shows the noisy image with 50% noise content. Figure 5, shows the donoised image after performing median filtering.

Fig. 4. Image with 90% of noise in original Image

Fig. 5. Denoised Image

Across different noise levels, the VLSI-based median filter successfully removed salt-and-pepper noise, demonstrating its reliability and efficiency in controlling the range of image deformation [13]. The filter utilized a systematic architectural architecture and optimization approach that led to substantial improvements in various image quality metrics such as SNR, PSNR, MAE, SSIM and MSE.

The implementation was optimized for minimal hardware occupancy, e.g. usage in terms of logic gates, Look-Up Tables (LUTs) and flip-flops, without any penalty for rate or real-time. Moreover, the architecture also managed to achieve low-power consumption, hence suitable for portable and embedded systems [14].

To validate the performance of the median filter in real-time, an implementation was performed in FPGA, demonstrating its efficiency in terms of the speed of processing, power consumption, and approximation. These properties realize the filter for real-time processing tasks in various domains from medical imaging (e.g., X-ray and MRI image clarification) to satellite imaging (e.g., pre-processing for improved geographic observations) and embedded vision systems (e.g., real-time monitoring and life cycle imaging in autonomous vehicles).

This VLSI design can pave the path for other applications of image processing technologies. A scalable architecture provides the flexibility to expand into systems with larger image windows and more advanced noise models. The success of deploying and validating this filter indicates its long-term potential to innovate and evolve with future smart imaging technologies.

TABLE I
COMPARISON OF EXISTING AND PROPOSED MEDIAN FILTER

Hardware Resources	Existing	Proposed
Slices	230	200
LUT	422	368
Combinational Path Delay	36.079 ns	32.072 ns

VII. COMPARATIVE ANALYSIS

Table II shows the comparison of image quality metrics for noised images and denoised images, at three levels of noise (10%, 20%, and (50%) with respect to existing (EM) and proposed method (PM) design. Table II is about the evaluation results of some considerations such as SNR, PSNR, SSIM, MSE, etc., before the image denoising. We can see that with the increase of noise SNR, PSNR, SSIM decreases, indicating that the quality of the image shows a downward trend with the increase of noise, and MSE increase, indicating that the pixel intensity error is more severe. On the other hand, Table III considers these same measures post denoising, with significant increases in SNR, PSNR, and SSIM, with reductions in MSE exhibit a considerable reduction.

VIII. CONCLUSION AND FUTURE SCOPE

In conclusion, this work illustrated the design and evaluation of an optimized 3x3 window based median filter architecture using comparator based logic. Hence the issue of finding the least number of processing stages and comparators to enable the sum computation, with the goal of simplifying the median selection process has overcome as optimized as possible. The architecture explores comparator and stage reduction to yield an efficient implementation of traditional approaches hence making it ideal for noise reduction in real time. It is optimized to overcome the common computational bottlenecks of median filtering, thus promoting more effective resource management. MATLAB simulations were carried out to validate the proposed filter in terms of functionality and effectiveness. The performance metrics such as Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Structural Similarity Index(SSIM) were calculated and analyzed before and after applying the filter. Higher PSNR values and lower MSE values, which showed that the design was capable of removing salt-and-pepper noise, improved the quality of image. This architecture showed a strong ability to trade off simplicity and performance.

One such major breakthrough is the design of filters to utilize larger window sizes like 5x5 or 7x7 enabling reduction of noise for high-resolution images. The increased aperture enables a more precise pixel data analysis which results in clearer, more accurate, and a lot more cleaner results in practices with a higher level of noise. An implementation of

TABLE II
COMPARISON OF PROPOSED VS. EXISTING METHOD MEDIAN FILTER FOR DIFFERENT NOISE LEVELS

NOISE LEVEL (%)	Performance Metrics							
	PSNR		SNR		MSE		SSIM	
	Existing [20]	Proposed	Existing [20]	Proposed	Existing [20]	Proposed	Existing [20]	Proposed
10%	22.75	35.33	9.81	29.580	4690.33	2130.44	0.782	0.985
30%	18.29	27.12	10.64	18.63	4973.99	2790.22	0.72	0.951
50%	8.08	18.22	3.95	12.52	5112.33	3102.47	0.52	0.693

hardware for a new staggered multilevel adaptive filter can be used to implement more of these digital adaptive filtering techniques, increasing the flexibility and adaptability of the filter in order to better suit different noise levels as necessary. Real-time adjustment of this sort allows the filter to be more efficient over and above many forms of noise. In addition, machine learning algorithms can be used to further optimize the filter process by learning and discriminating the type of noise according to the context of the image, and using the result in the filtering process. This makes a more accurate and smarter filtering process that learns from the specific properties of each image Another area of focus in filter design is power efficiency. In doing so, they become a more appropriate choice for continuous applications in real-time settings, e.g., enabling them across the board for autonomous systems, video surveillance and IoT devices. Battery Powered Device - This power efficiency is important in battery-powered devices, where conserving energy is paramount. Future filters may also develop multi-channel processing capability for colour images and video streams. This would broaden filter usage in multimedia applications especially in video editing, medical imaging and more.

REFERENCES

[1] Appiah, O.,Opoku, E.,Ezekiel, M.M., and Quayson,E.(2022).Decision-based median- of-median filtering algorithm to denoise salt-and-pepper noise in colour images.Graphics,Visual Computing,2022,1–16.

[2] Aslam,N.,Ehsan,M.K.,Rehman,Z.U.,Hanif,M., and Mustafa,G.(2023). A modified form of different applied median filter for removal of salt and pepper noise. Multimedia Tools and Applications, 82, 7479–7490

[3] Bevara,V., and Sanki,P.K.(2020). VLSI implementation of high throughput parallel pipeline median finder for IoT applications.Sadhana,45(75),1–5.

[4] Charmouti,B.,Junoh,A.K.,Abdurrazzaq,A., and Mashor,M.Y.(2022). A new denoising method for removing salt and pepper noise from image.Multimedia Tools and Applications,81(4), 3981–3993.

[5] Doe,J., and Smith,R. (2023). The behavior of happy numbers And detection of cycles. International Journal of Number Theory,10(2),123–135

[6] Draz,H.H.,Elashker,N.E.,and Mahmoud,M.M.A.(2023). Optimized algorithms and hardware implementation of median filter for image processing. Circuits, Systems, and Signal Processing, 42, 1550–1570.

[7] Erkan,U.,Enginoglu,S., and Thanh,D.N..(2020). Adaptive frequency median filter for the salt and pepper denoising problem. IET Image Processing, 14(7), 1291–1302.

[8] Gupta,S.,and Kumar,N.(2021).VLSI architecture for real-time median filter.IEEE Transactions on Very Large-Scale Integration(VLSI)Systems,29(8),1482–1490.

[9] Huang,T.,Yang,G., and Tang,G.(1979). A fast two-dimensional median filtering algorithm.IEEE Transactions on Acoustics,Speech,and Signal Processing,27(1),13–18.

[10] Liu,P.,Zhang,X., and Li,G.(2020). Fault-tolerant median filter for image noise suppression in FPGA.IEEE Transactions on Circuits and Systems I: Regular Papers,67(12), 4678–4688.

[11] Nodes,T., and Gallagher,N.(1982). Median filters: Some modifications and their properties.IEEE Transactions on Acoustics,Speech,and Signal Processing,30(5),739–746.

[12] Priyadharsini, M.S. and Sathiaseelan, J.G.R. (2023). "The new robust adaptive median filter for denoising cancer images using image processing techniques." Indian Journal of Science and Technology, vol. 16, no. 1, p. 35.

[13] Rathod,N., and Chauhan,A.(2017). High density salt and pepper impulse noise removal using modified median filter.Journal of Emerging Technologies and Innovative Research(JETIR),4(11),572–575.

[14] Rathod,N., and Chauhan,A.(2017). High density salt and pepper impulse noise removal using modified median filter.Journal of Emerging Technologies and Innovative Research(JETIR),4(11),572–575.

[15] Sakra,F.,Mohamed,N.R.,Muthusamy,S.,Pandiyan,S.,Manickam,C., Murugesan,A., Gnanavel,D.,Goma,I.A.E., and Elminaam,D.S.A.(2022). An method for designing an efficient switching median filter using VLSI architecture to remove salt and pepper noise.Journal of Computing and Communication,1(2), 57–68.

[16] Seetharaman,R.,Tharun,M., and Anandan,K.(2021). A novel approach in hybrid median filtering for denoising medical images. IOP Conference Series: Materials Science and Engineering, 1187(1), 012028.

[17] Vasanth, K. (2014). "VLSI Architecture of Decision based Modified Selection Sort Filter for Salt and Pepper Noise Removal". International Journal on Intelligent Electronic Systems, vol. 13, no. 4, pp. 41-56.

[18] Yildirim, M. (2021). Analog circuit implementation based on median filter for salt and pepper noise reduction in image.Analog Integrated Circuits and Signal Processing, 107,195–202.

[19] Wang, Biaobiao, and Qiang Xiang (2020). "Fast median filter image processing algorithm and its FPGA implementation". Frontiers in Signal Processing, vol. 4, no. 4,pp.88-94.

[20] Nanduri, S., Kamaraju, M. (2024). Energy-Efficient Median Filter Core Architecture for Impulse Noise Removal in Smart Measurement Systems. SN Computer Science, 5(1), 154.