International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)
Vol. 11-No.2 (2025) pp. 2018-2027
http://www.ijcesen.com
ISSN: 2149-9144
Research Article

Enhancing Image Processing Capabilities through Advanced Approximate Multipliers

Sreelakshmi Vadlamudi[1]*, Syamala Yarlagadda[2], R. Madhu[3]

[1]Jawaharlal Nehru Technological University, Research scholar, ECE Department, 533033, Kakinada-India
* **Corresponding Author Email:** sreelakshmi.vadlamudi@gmail.com - **ORCID:** 0000-0002-5633-5112

[2]Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, 521356, Gudlavalleru-India
Email: coolsyamu@gmail.com - **ORCID:** 0000-0002-0541-8313

[3]Jawaharlal Nehru Technological University, ECE Department, 533033, Kakinada-India
Email: madhu_ramarkula@jntucek.ac.in - **ORCID:** 0000-0002-4092-5644

Article Info:

DOI: 10.22399/ijcesen.1399
Received : 02 February 2025
Accepted : 26 March 2025

Keywords :

Approximate computing,
Image processing,
4-2 compressors,
Real-time processing,
Image enhancement.

Abstract:

This paper explores the utilization of advanced approximate multipliers and 4-2 compressors to enhance efficiency in digital image processing applications. As the demand for real-time image analysis grows, the need for computational efficiency becomes increasingly critical, especially in resource-constrained environments. The proposed approach employs approximate computing to achieve a significant reduction in processing speeds and energy consumption while maintaining acceptable levels of output quality. Through a systematic implementation using a Field-Programmable Gate Array (FPGA) and MATLAB, we demonstrate the effectiveness of approximation techniques in image blending and enhancement tasks. Experimental results highlight a favorable trade-off between accuracy and performance, indicating that minor inaccuracies do not significantly impair visual quality. Our findings suggest that integrating approximate computing methodologies can revolutionize various technology-driven industries by enabling faster and more efficient image processing solutions.

1. Introduction

Real-time image analysis and enhancement require processing vast amounts of data, making computational efficiency a critical factor in digital image processing [1,2]. Efficient algorithms ensure faster processing speeds, reduced energy consumption, and the ability to operate on resource-constrained hardware, which is particularly essential for embedded and mobile applications [3,4]. Approximate computing has emerged as a strategic approach to enhancing computational performance, especially in scenarios where absolute accuracy is not necessary [5,6]. By prioritizing a balance between performance and accuracy, this approach significantly improves processing speed and energy efficiency. In digital image processing, minor errors—often imperceptible to the human eye—can be acceptable in exchange for faster and more power-efficient computations [7-9]. Multimedia, video processing, and mobile imaging applications can greatly benefit from approximate computing's capability to handle large-scale and complex image data more efficiently.

In the field of approximate computing, addition is a fundamental arithmetic operation that plays a crucial role in various applications. Its implementation typically relies on probabilistic imprecise arithmetic or deterministic approximation logic, categorized into design-time and runtime methods, as discussed in [10]. These approaches are instrumental in the development of advanced adder designs, representing significant progress in this area. One such innovative technique is the Lower-Part OR Adder (LOA), introduced in [11], which slightly modifies the conventional truth table of a full adder using approximation logic. This intentional deviation aims to enhance computational performance with minimal accuracy loss. Similarly, approximate mirror adders (AMAs), as described in [12], reduce overall circuit complexity by decreasing the average transistor count in a mirror adder design, ultimately conserving power. Another noteworthy innovation is the Probabilistic Full Adder (PFA), which leverages probabilistic CMOS (PCMOS) technology. All these techniques were shown in Figure 1. As highlighted in [13,14], this approach is recognized for its ability to lower power consumption while replicating the behavior of nanometric devices. Its potential applications align well with future computing paradigms, offering a balance between probabilistic operations and energy efficiency.

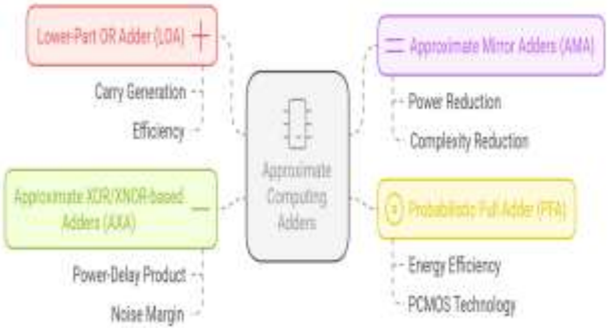


Figure 1. *Innovations in Approximate Computing*

Within the field of approximation computing, multi-bit adders are often divided into two modules: an approximate lower module that processes fewer significant bits, and an accurate upper module that processes more significant bits. A single-bit approximation adder that performs a modified, and thus imprecise, function of addition processes each lower bit. In order to achieve desired savings in speed and power utilization, as detailed in [15], this change frequently involves simplifying a full adder design at the circuit level. This is equivalent to changing certain entries in a full adder's truth table at the functional level. Approximate Mirror Adders (AMAs) demonstrate notable reductions in power dissipation and circuit complexity; they are developed from a transistor-level logic reduction. As discussed in [15], they also show a reduced latency as a result of the node capacitance being charged or discharged more quickly. This successfully strikes a balance between precision, energy, and performance. Approximate XOR/XNOR-based Adders (AXAs): AXAs use a 10-transistor adder with XOR/XNOR gates and multiplexers implemented to bypass transistors. They have a favorable power-delay product (PDP) and attractive performance profiles, but as discussed in [16], they may have a reduced noise margin. Lower-Part-OR Adder (LOA): This design generates the carry-in for the accurate upper part when both inputs to the most significant bit adder in the lower part are "1," and utilizes an OR gate to estimate the sum of each bit in the approximate lower part. As shown in [17], this technique enables efficient operation by ignoring the majority of carries in an adder's less important bottom half. In an n-bit adder, the critical path latency is asymptotically proportional to log(n), which limits the performance of parallel adders such as the carry look-ahead (CLA) [18,19]. However, speculative adders can achieve sub-logarithmic latency by making use of the fact that normal carry propagation chains are far shorter than the worst-case situation. These observations and sources offer a comprehensive overview of the topic, emphasizing the value and implementation of approximation computing methods in contemporary computer systems. To advance efficiency and accuracy in image processing, this work focuses on integrating sophisticated approximate 4-2 compressors into high-precision multipliers. The objective is to demonstrate how approximate computing can enhance image quality while significantly reducing computational demands by leveraging the capabilities of these custom-designed compressors. A key contribution of this study is the development of innovative compressor topologies that optimize critical performance parameters such as delay, area, and power consumption. Additionally, through comprehensive testing and analysis, this work provides real data and insights into the trade-offs between efficiency and accuracy, offering valuable guidance for future applications of approximate computing in multimedia.

2. Background and Related Work

Various key technologies have influenced current approaches and practices in image processing and approximate computing. Traditionally, image processing has relied on exact arithmetic operations to ensure the quality and accuracy of image enhancements and transformations [20,21]. However, these processes often demand substantial computational resources and energy, posing significant challenges to scalability and application, especially in resource-constrained environments. Approximate computing introduces a transformative opportunity by enabling controlled accuracy trade-offs during optimization [5,22]. This approach has proven particularly effective in scenarios where improvements in speed and energy efficiency outweigh minor reductions in output quality. For real-time applications such as video streaming, augmented reality, and autonomous driving systems, this translates into faster processing times for image rendering, filtering, and compression tasks.

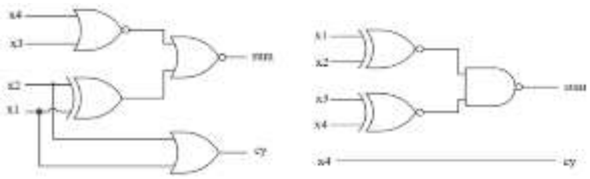
Focusing on the specific application of 4-2 compressors, these components play a crucial role in efficient binary addition within multipliers, which are fundamental to many image processing techniques involving matrix operations and convolution [23,24]. Traditional 4-2 compressor designs aim for exact results, often resulting in complex and power-intensive circuits [25]. Previous implementations have explored various designs to optimize key performance metrics such as power consumption, area, and speed. However, these approaches have generally overlooked the potential benefits of approximation techniques, which could further enhance efficiency. Despite advancements in compressor design, existing methods often struggle to balance processing overhead and accuracy. The emphasis on high precision in conventional systems can lead to unnecessary resource consumption, even in cases where lower accuracy would suffice without a noticeable loss in quality. Furthermore, as the demand for higher resolution and larger image datasets grows, conventional compressor designs often become inefficient. These challenges underscore the need for continuous innovation in compressor technology, aiming to develop designs that not only enhance core performance metrics but also offer flexibility in accuracy to improve overall system efficiency.

The drive for energy-efficient digital system design has led to significant advancements in approximation computing methodologies within the field of inexact computing. As noted by Jie Han et al., approximation computing leverages certain systems' inherent ability to tolerate minor losses in the quality or optimality of computed results. By relaxing strict accuracy requirements, these methods substantially enhance energy efficiency [26]. Jie Han et al. also highlight addition as a fundamental arithmetic operation crucial to implementing inexact computing. In nanoscale circuits, this function can manifest across different operational profiles, ranging from approximation to probabilistic behaviors. The lack of precise measures for evaluating the effectiveness of various imprecise designs has prompted the introduction of novel metrics that assess both the reliability and energy efficiency of approximate and probabilistic adders. The introduction of Sequential Probability Transition Matrices (SPTMs), along with metrics such as Mean Error Distance (MED) and Normalized Error Distance (NED), has facilitated the evaluation of multi-bit adder implementation accuracy [27].

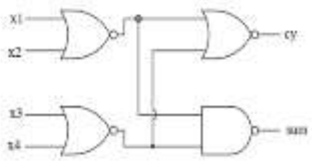
Dynamic bit-width adaptation for discrete cosine transform (DCT) applications is a technique developed by Jongsun Park that effectively trades computing energy for picture quality. According to the sensitivity of each frequency component, multiple operand bit-widths are used in this method, which minimizes loss of quality while also saving energy. A bit-width selection technique is included to further optimize power consumption reductions; in typical operations, savings range from 36% to 75% with little to no influence on image quality [28]. Jie Han et al. continue the approximate computing subject by going into algorithm-level approaches, relevant error and quality measurements, and the construction of approximation arithmetic blocks. These elements are essential for allowing approximation computing to be used practically in a variety of contexts, especially where small errors may be accepted in exchange for improved performance and lower power consumption [26]. By examining the built-in application resilience found in a number of popular apps in the recognition, data mining, and search areas, Vinay et al. add to the conversation. Through the use of approximation models, their study presents a systematic framework for Application Resilience Characterization (ARC), which divides applications into robust and sensitive sections. This research helps evaluate the capability of different approximation computing approaches for certain applications as well as comprehend the link between resilience and important application aspects [29]. A novel design strategy for the approximation of multipliers is investigated by Suganthi Venkatachalam et al. This strategy modifies the partial products to add variable probability elements. This approach achieves significant power savings and improved precision over precise multipliers by drastically reducing the circuitry complexity and is applied in two variations of 16-bit multipliers. These approximation multipliers' effectiveness and potential for wider use are demonstrated by the validation of their performance in image-processing applications [30]. Compared to an exact multiplier, this method, when applied to two types of 16-bit multipliers, yields notable power reductions of 72% and 38%, respectively. Additionally, these multipliers show improved accuracy compared to current approximation multipliers, with mean relative errors as low as 7.6% and 0.02%. One of the models gets the greatest peak signal-to-noise ratio when the performance of these multipliers is assessed in an image processing application, proving the designs' practicality in real-world scenarios [30]. Suganthi Venkatachalam et al. concentrate on approximating the radix-4 booth multiplication computation in their other work, which is essential for lowering power consumption in multimedia and deep learning applications. The new feature is the approximation used for the radix-4 partial product production and accumulation processes. Compared to an identical booth multiplier, this design results in a 41% decrease in area and a 49% reduction in power usage. In addition, our approximate booth multiplier exhibits better area, power, and error metrics in comparison to previous approximate multiplier efforts. This design's efficiency and promise for high-throughput applications are further demonstrated by its comparable performance to precise multiplication units in the discrete cosine transform (DCT) encoding step of JPEG compression [31]. In a world increasingly dominated by digital and multimedia applications, each of these contributions highlights the growing significance of approximation computing in attaining power efficiency and performance improvements. Future research and development in the area of inexact computing appears to have a bright future because to the combination of various technologies and approaches. By incorporating approximation computing ideas into new designs of 4-2 compressors, this research seeks to close these gaps and improve the compressors' ability to balance efficiency and precision. By doing this, it hopes to open the door for image processing technologies that are more scalable and adaptable and can satisfy the many requirements of contemporary multimedia applications.

3. Methodology

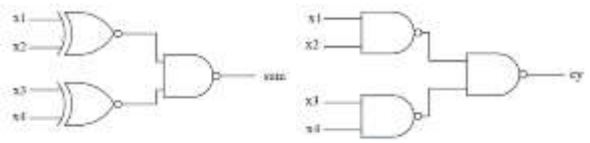
In this work, 4–2 compressors are designed to optimize the accuracy–efficiency trade-off, which is essential for image processing applications. The design of these compressors allows for small errors in binary addition, leading to faster processing times and reduced energy consumption. Each compressor configuration is specifically crafted to modify the standard architecture, where certain logic gates are altered to simplify the circuit, accepting a controlled number of computational errors. This approach takes advantage of the natural error tolerance of human vision, where minor image processing errors—such as those introduced during blending or filtering—are often imperceptible. The compressors incorporate various approximation techniques that reduce the logical depth and gate count, including eliminating specific carry outputs and simplifying gate configurations.
We have designed five different 4:2 compressors—P1, P2, P3, P4, and P5—optimized for image processing applications shown in Figure 2 , where high-speed and low-power computation are critical. These compressors play a key role in performing efficient partial product reduction in multipliers, which are essential for image filtering, enhancement, and feature extraction. Each design utilizes XOR, AND, and OR gates to generate sum and carry outputs with different levels of complexity, allowing for a trade-off between power, delay, and area. P1 and P3 incorporate multiple logic levels, which may slightly increase delay but ensure accurate computation. P2 and P4 provide a more balanced approach with reduced complexity, improving overall speed and efficiency. P5 follows a minimalistic design, aiming for lower power consumption, making it ideal for energy-efficient image processing hardware. By comparing these architectures, we can determine the most suitable compressor for real-time and high-performance image processing applications.



(a) P1 compressor *(b) P2 compressor*



(c) P3 compressor



(d) P4 Compressor

(e) P5 Compressor

Figure 2. *Various proposed compressors*

The implementation of the five designed 4:2 compressors—P1, P2, P3, P4, and P5—has yielded varying results in terms of area, delay, and power consumption, highlighting their strengths and trade-offs for different applications. P1 occupies the largest area (15.39 µm²) and consumes the highest power (205 nW) while achieving a moderate delay of 353 ps, making it suitable for applications prioritizing accuracy over power efficiency. P2 offers a balanced trade-off with a slightly reduced area (13.68 µm²) and lower power consumption (173 nW) while maintaining a delay of 340 ps. P3, on the other hand, achieves the smallest area (10.26 µm²) but suffers from the highest delay (439 ps), which may impact real-time processing applications. P4 provides an efficient balance with 12.31 µm² area, 312 ps delay, and 149 nW power consumption, making it a strong contender for power-sensitive designs. Finally, P5 emerges as the most power-efficient design (94 nW) while maintaining the lowest delay (305 ps) and compact area (9.91 μm²), making it ideal for high-speed, low-power applications. These results allow for the selection of the most suitable compressor based on specific design constraints in VLSI and image processing applications. Table 1 shows area, delay and power for different compressors.

Table 1. *Area, delay and power for different compressors.*

Compressor Type	Area (μm^2)	Delay (ps)	Power (ns)
P1	15.39	353	205
P2	13.68	340	173
P3	10.26	439	127
P4	12.31	312	149
P5	9.91	305	94

By embedding these approximate 4-2 compressors within the critical route of the multiplier architecture, where they assist in accumulating partial products during multiplication, these compressors can be integrated into high-accuracy multipliers [32, 33]. The overall speed of the multiplication operation relies on the compressors' ability to perform quick additions, making this integration crucial to understanding the multipliers' performance. These multipliers are utilized in image processing tasks that involve repetitive addition operations, such as convolution algorithms for filtering or blending images. The approximate compressors are incorporated into high-accuracy multipliers designed to strike a balance between maximizing processing speed and minimizing output quality loss. This integration is particularly important for real-time applications, such as live video processing or on-the-fly image adjustments in embedded systems.

A thorough examination of the application requirements and the permissible error thresholds in common image processing tasks guided the selection of specific approximation levels for the compressor designs [34–36]. These approximation levels were determined based on theoretical studies and experimental data that outlined the relationship between approximation-induced errors and their perceived impact on image quality. Figure 3 is integration of approximate compressors in multipliers.

Using the five designed 4:2 compressors—P1, P2, P3, P4, and P5—we have implemented five different multipliers, each optimized for varying trade-offs in area, delay, and power consumption. These multipliers leverage the unique characteristics of their respective compressors to enhance computational efficiency in arithmetic circuits, particularly for high-performance and low-power applications such as image processing and digital signal processing. M1, utilizing the P1 compressor, occupies the largest area (401.16 μm²) and has the highest gate count (218), but it also demonstrates a competitive timing of 2845 ps and power consumption of 7.59 μW. M2, based on the P2 compressor, slightly reduces area and gate count (392.27 μm² and 199 gates) while achieving improved timing (2828 ps) at a slightly higher power consumption of 7.63 μW. M3 and M4, utilizing P3 and P4 compressors respectively, exhibit nearly identical characteristics in area (392.95 μm²) and gate count (200 gates), with M3 having the highest delay (2931 ps) among all multipliers. M5, implemented with the P5 compressor, achieves the most optimized results, occupying the least area (366.62 μm²) with only 199 gates, while also maintaining competitive timing (2920 ps) and the lowest power consumption (7.51 μW). These results highlight the trade-offs involved in choosing different compressor designs, offering flexibility based on application-specific requirements, particularly for power-efficient and high-speed computing in image processing applications.

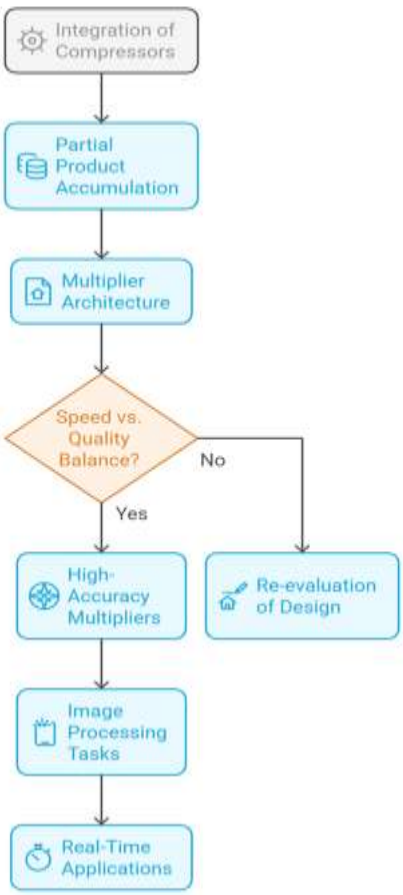


Figure 3. *Integration of Approximate Compressors in Multipliers*

To establish a baseline for the maximum allowable error, simulations were conducted to assess the error sensitivity of various image processing tasks. These simulations offered insights into how much the compressor designs could simplify their structures without exceeding the established perceptual quality limits. The selection of these approximation levels also took into account the operational context of the multipliers. The goal was to strike a balance between achieving optimal performance in scenarios where maintaining a minimum level of image fidelity was critical, while prioritizing processing efficiency. This approach ensures that systems using these multipliers benefit from enhanced performance and reduced power consumption, while still maintaining adequate quality for their intended applications. Table 2 shows area, delay, and power for different multipliers.

Table 2. *Area, delay, and power for different Multipliers.*

Multiplier using Compressors	Area (μm^2)	Gates	Delay (ps)	Power (ns)
M1	401.16	218	2845	7.59
M2	392.27	199	2828	7.63
M3	392.95	200	2931	7.62
M4	392.95	200	2922	7.62
M5	366.62	199	2920	7.51

4. System Implementation

The system architecture combines both hardware and software components that are specifically designed for image modification tasks, aiming to process digital images efficiently. At the core of the hardware setup is a high-performance computer platform equipped with a Field-Programmable Gate Array (FPGA) that hosts the advanced 4-2 compressors and approximation multipliers. The FPGA's ability to effectively handle parallel computations is crucial for meeting the real-time processing demands of image blending and enhancement, which is why it was chosen for this role. On the software side, the system utilizes MATLAB, an interactive language and high-level programming environment tailored for numerical computing, visualization, and programming. MATLAB is particularly well-suited for this project due to its extensive library of built-in image processing functions and its fast algorithm prototyping capabilities. To achieve high-quality results, the image processing method employed in this research focuses on image blending and enhancement through a series of steps. To simplify processing and reduce data complexity, the images are first imported and converted to grayscale. Each image is then subjected to several enhancement processes, including median filtering to reduce noise and histogram equalization to improve contrast. A custom-designed function for image blending utilizes the approximations of the multipliers. By adjusting each pixel's intensity values according to weighted averages determined by the multipliers, this function effectively merges two images into one, balancing computational efficiency and visual clarity. The use of approximation arithmetic in the multipliers enables faster computation times, making real-time processing possible. Figure 4 is image processing method: blending and enhancement.

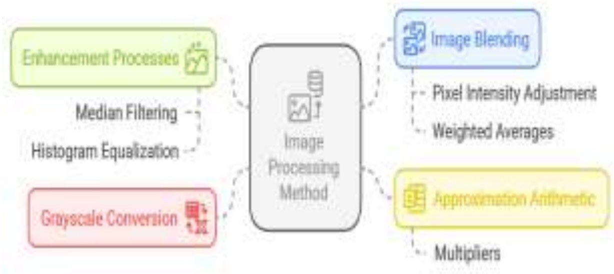


Figure 4. *Image processing method: Blending and Enhancement*

The computing environment is configured to optimize image processing tasks' accuracy and efficiency. Given the large datasets typically encountered in image processing applications, a desktop computer with a multi-core CPU and sufficient RAM is required to run the MATLAB code effectively. The computationally intensive arithmetic operations are offloaded from the main processor to the FPGA, which is integrated using a hardware description written in Verilog HDL. The FPGA is specifically designed to handle operations related to approximation multiplication. This setup mirrors a practical application scenario, where computing resources are optimized for specific tasks, while also providing a stable platform for developing and testing image processing algorithms. By integrating verilog with MATLAB, the FPGA and software work together, allowing for comprehensive analysis and fine-tuning of the approximation compressors' performance to ensure the system can deliver the desired results effectively and consistently.

5. Performance Evaluation

5.1 Methodology for Assessing Image Quality: MSE, PSNR, SSIM

The approach utilizes three well-established metrics—Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM)—to evaluate the quality of images processed using approximation multipliers [28]. These metrics provide a comprehensive assessment of perceptual quality and image integrity. Mean Squared Error (MSE) quantifies the average squared intensity differences between the original and processed images, serving as a clear indicator of distortion [28]. Peak Signal-to-Noise Ratio (PSNR) is derived from MSE and represents the ratio of the highest possible pixel value of an image to the distortion introduced by processing. It is typically expressed in logarithmic decibels, with higher values indicating better image quality [28]. Structural Similarity Index Measure (SSIM) evaluates the perceptual differences between the original and processed images by analyzing three key attributes: brightness, contrast, and structure. SSIM values range from -1 to 1, with a value of 1 indicating perfect similarity [37–39]. These metrics are computed using MATLAB algorithms integrated into the workflow, ensuring an automated assessment of each processed image. Figure 5 is image quality assessment metrics.

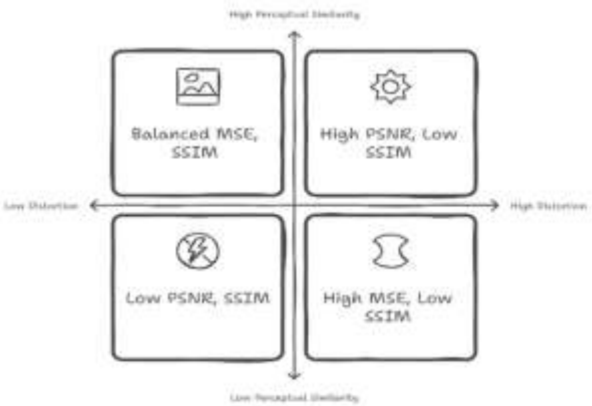


Figure 5. *Image Quality Assessment Metrics*

5.2 Comparison of Performance Metrics Between the Proposed and Traditional Methods

The final phase of the performance evaluation involves a direct comparison of the proposed system's performance metrics with those obtained using traditional, exact multipliers. This comparison not only considers image quality metrics such as MSE, PSNR, and SSIM but also includes computational efficiency parameters like processing time and power consumption [28]. This performance comparison is crucial for highlighting the advantages of approximate computing techniques in practical applications where both image quality and computational performance are critical. The results from this comprehensive evaluation provide empirical evidence supporting the feasibility of approximation multipliers in real-world image processing applications. These findings serve as a foundation for advocating the adoption of approximation computing techniques in both industry and academia [37–39].

6. Results and Discussion

The application of advanced 4-2 compressors for integrating approximation multipliers has yielded significant results in image processing, particularly in picture blending and enhancement applications. This section presents the outcomes of these processes, supported by numerical evaluations and visual evidence obtained from tests conducted using the developed MATLAB code. The primary objective was to assess the accuracy of approximation computation in processing two benchmark test images, "Department" and "Logo".
Key Processing Stages
1. Conversion to Grayscale: Both color images were initially converted to grayscale to simplify data handling and reduce processing requirements. This step is crucial in lowering the computational complexity of subsequent processes, ensuring more efficient processing.
2. Image Enhancement: Several enhancement techniques were applied to the grayscale images to improve visual quality. Histogram equalization was utilized to enhance contrast, followed by median filtering to reduce noise. Finally, a sharpening filter was applied to highlight fine details. These preprocessing steps ensured that the images were optimally prepared for blending, enhancing detail resolution and overall clarity. 3. Blending Using Approximate Multipliers: The enhanced images were blended using a custom function that incorporated approximate multipliers. This function leveraged an estimated 4-2 compressors to compute a predefined weighting scheme, adjusting pixel values accordingly. The blending process was carefully monitored to maintain a balance between computational efficiency and image quality, demonstrating the effectiveness of approximation computing. This study provides a comprehensive analysis of the approximate multiplier-based image processing pipeline, specifically focusing on the combination and enhancement of the images labeled "Department" and "Logo". The sequential flow through the image processing stages is detailed and illustrated in the following sections.

Figure 6. *Department - Histogram Equalized, Denoised, and Sharpened Images*

The histogram-equalized version of the "Department" image is presented in Figure 1. Histogram equalization enhances the contrast of the image, making details more prominent and visually striking . This step is crucial for expanding the grayscale image's dynamic range, which facilitates subsequent processing stages. Figure 6 is department - histogram equalized, denoised, and sharpened images and figure 7 shows logo - histogram equalized, denoised, and sharpened images. Following this, the "Department" image undergoes median filtering to reduce noise. This process eliminates random pixel anomalies and smooths the image texture, ensuring that further enhancements focus on relevant image features without amplifying noise. Noise reduction is essential for maintaining the integrity of image details while preventing distortions in later processing steps. Finally, the image is sharpened to enhance its edges and fine details. Sharpening plays a vital role in improving clarity and emphasizing essential features of the "Department" image. This post-processing technique enhances the visibility of key structures, making the image more suitable for analysis or presentation. The "Logo" image, after histogram equalization, exhibits improved visual clarity. Similar to the "Department" image, this process redistributes pixel intensity values, significantly enhancing the visual appeal by making the logo's features more distinct and recognizable. Following histogram equalization, noise reduction is applied to the Logo image, resulting in a smoother appearance with fewer pixel-level distortions. Noise reduction plays a critical role in preserving the integrity of the logo’s design, ensuring that its aesthetic and functional elements remain intact and are not affected by unwanted artifacts. This step is particularly important in maintaining a clean and professional look, essential for branding and visual representation. Finally, sharpening is applied to enhance the edges and lines of the Logo image. This phase ensures that the logo retains a crisp and polished appearance, improving its overall visibility and making it more effective for branding or visual identification purposes. By refining the image’s sharpness, this process enhances the logo’s clarity, reinforcing its impact in various digital and print applications.

Figure 7. *Logo - Histogram Equalized, Denoised, and Sharpened Images*

Using approximate multipliers, this blended image seamlessly combines elements from the "Department" and "Logo" images. The structured blending process ensures that both images contribute equally to the final composition without overpowering each other. This demonstrates how approximation arithmetic can be effectively applied in image fusion. To enhance visual quality and eliminate blending artifacts, post-processing techniques such as image-guided filtering have been applied. This refinement phase is essential to delivering a polished and aesthetically appealing final result. Figure 8 is blended image and enhanced blended image.

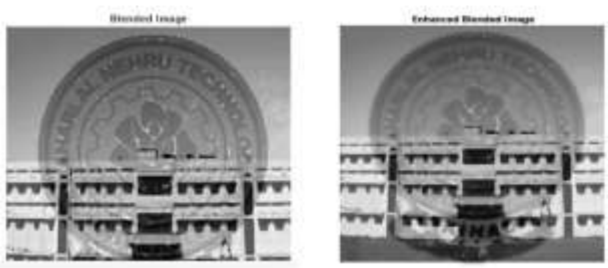


Figure 8. *Blended image and Enhanced blended image*

The performance of the enhanced "Logo" image was evaluated using three key metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). These metrics provide valuable insights into the quality and effectiveness of the applied image processing techniques. Table 3 shows area, delay, and power for different multipliers.

Table 3. *Area, delay, and power for different Multipliers.*

Image Type	**MSE**	**PSNR (dB)**	**SSIM**
Department	0.033524	14.75	0.6369
Logo	0.021081	16.76	0.7989

Despite the intentional reduction in computational accuracy, the processed images maintained a high level of visual quality, as demonstrated by the SSIM and PSNR values. The final blended outputs and enhanced images preserved their primary visual characteristics, suggesting that the approximation techniques did not introduce noticeable degradation. These findings indicate that the chosen degree of approximation effectively balances computational complexity reduction with image fidelity preservation.

6. Conclusion

This study effectively demonstrated the use of approximation multipliers with 4-2 compressors to enhance the efficiency of image processing applications. The quantitative metrics—Structural

Similarity Index Measure (SSIM) of 0.6369, Peak Signal-to-Noise Ratio (PSNR) of 14.75 dB, and Mean Squared Error (MSE) of 0.033524—confirm the effectiveness of the approximation computing approach. These results validate the trade-off strategy by showing that, despite a slight reduction in computational accuracy, the overall visual quality of the images remains high.

A key advantage of this method is the significant reduction in computation time and energy consumption without noticeably compromising image quality. This balance between efficiency and fidelity makes approximation computing particularly promising for resource-constrained environments. Faster processing speeds are crucial for real-time image processing applications such as mobile photography, augmented reality, and video streaming. The findings suggest that approximation computing could serve as a viable solution to the growing demand for efficient image processing techniques across various technology-driven industries.

Further research could focus on refining the design of approximation components to minimize the trade-offs between accuracy and performance. Investigating different levels of approximation and their impact on various multimedia applications may provide deeper insights into optimizing these techniques for specific use cases.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Ma, Y., Wang, L., Liu, P., & Ranjan, R. (2015). Towards building a data-intensive index for big data computing–A case study of Remote Sensing data processing. *Information Sciences*, 319, 171–188. https://doi.org/10.1016/j.ins.2015.02.004

[2] Umirzakova, S., Ahmad, S., Khan, L. U., & Whangbo, T. (2023). Medical image super-resolution for smart healthcare applications: A comprehensive survey. *Information Fusion*, 102075. https://doi.org/10.1016/j.inffus.2023.102075

[3] Wolf, W. (2010). High-performance embedded computing: Architectures, applications, and methodologies. Elsevier.

[4] Garcia, P., Compton, K., Schulte, M., Blem, E., & Fu, W. (2006). An overview of reconfigurable hardware in embedded systems. *EURASIP Journal on Embedded Systems*, 2006, 1–19. https://doi.org/10.1155/ES/2006/23679

[5] Agrawal, A., Choi, J., Gopalakrishnan, K., Gupta, S., Nair, R., Oh, J., Prener, D. A., Shukla, S., Srinivasan, V., & Sura, Z. (2016, October). Approximate computing: Challenges and opportunities. In 2016 *IEEE International Conference on Rebooting Computing (ICRC)* (pp. 1–8). IEEE. https://doi.org/10.1109/ICRC.2016.7738708

[6] Amanollahi, S., Kamal, M., Afzali-Kusha, A., & Pedram, M. (2020). Circuit-level techniques for logic and memory blocks in approximate computing systems. *Proceedings of the IEEE*, 108(12), 2150–2177. https://doi.org/10.1109/JPROC.2020.3025098

[7] Grundhöfer, A., & Iwai, D. (2018, May). Recent advances in projection mapping algorithms, hardware and applications. *Computer Graphics Forum*, 37(2), 653–675. https://doi.org/10.1111/cgf.13465

[8] Bock, C. H., Poole, G. H., Parker, P. E., & Gottwald, T. R. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Critical Reviews in Plant Sciences*, 29(2), 59–107. https://doi.org/10.1080/07352681003617285

[9] Lingamneni, A., Enz, C., Palem, K., & Piguet, C. (2013). Designing energy-efficient arithmetic operators using inexact computing. *Journal of Low Power Electronics*, 9(1), 141–153. https://doi.org/10.1166/jolpe.2013.1231

[10] Huang, J., & Lach, J. (2011, January). Exploring the fidelity-efficiency design space using imprecise arithmetic. In *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)* (pp. 579–584). IEEE. https://doi.org/10.1109/ASPDAC.2011.5722228

[11] Mahdiani, H. R., Ahmadi, A., Fakhraie, S. M., & Lucas, C. (2009). Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications. *IEEE Transactions on Circuits and Systems I:* 57(4), 850–862. https://doi.org/10.1109/TCSI.2009.2027626

[12] Gupta, V., Mohapatra, D., Park, S. P., Raghunathan, A., & Roy, K. (2011, August). IMPACT: IMPrecise adders for low-power approximate computing. In *IEEE/ACM International Symposium on Low Power Electronics and Design* (pp. 409–414). https://doi.org/10.1109/ISLPED.2011.5993664

[13] Cheemalavagu, S., Korkmaz, P., Palem, K. V., Akgul, B. E., & Chakrapani, L. N. (2005, October). A

probabilistic CMOS switch and its realization by exploiting noise. In *IFIP International Conference on VLSI* (pp. 535–541).
[14] Lau, M. S., Ling, K. V., Chu, Y. C., & Bhanu, A. (2010, March). A general mathematical model of probabilistic ripple-carry adders. In 2010 *Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)* (pp. 1100–1105). IEEE. https://doi.org/10.1109/DATE.2010.5457038
[15] Gupta, V., Mohapatra, D., Raghunathan, A., & Roy, K. (2013). Low-power digital signal processing using approximate adders. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(1), 124–137. https://doi.org/10.1109/TCAD.2012.2223466
[16] Yang, Z., Jain, A., Liang, J., Han, J., & Lombardi, F. (2013, August). Approximate XOR/XNOR-based adders for inexact computing. In *2013 IEEE International Conference on Nanotechnology (IEEE-NANO 2013)* (pp. 690–693). IEEE.
[17] Mahdiani, H. R., Ahmadi, A., Fakhraie, S. M., & Lucas, C. (2009). Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications. *IEEE Transactions on Circuits and Systems I:* 57(4), 850–862. https://doi.org/10.1109/TCSI.2009.2027626
[18] Lu, S. L. (2004). Speeding up processing with approximation circuits. *Computer*, 37(3), 67–73.
[19] Verma, A. K., Brisk, P., & Ienne, P. (2008, March). Variable latency speculative addition: A new paradigm for arithmetic circuit design. In *Proceedings of the conference on Design, Automation and Test in Europe* (pp. 1250–1255).
[20] Agaian, S. S., Panetta, K., & Grigoryan, A. M. (2001). Transform-based image enhancement algorithms with performance measure. *IEEE Transactions on Image Processing*, 10(3), 367–382.
[21] Shih, F. Y. (2017). Image processing and mathematical morphology: *Fundamentals and applications. CRC Press.*
[22] Hashemi, S. (2018). Approximate Computing Techniques for Accuracy-Energy Trade-offs *(Doctoral dissertation, Brown University).*
[23] Strollo, A. G. M., Napoli, E., De Caro, D., Petra, N., & Di Meo, G. (2020). Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers. *IEEE Transactions on Circuits and Systems I:* 67(9), 3021–3034.
[24] Aizaz, Z., & Khare, K. (2022, January). State-of-art analysis of multiplier designs for image processing and convolutional neural network applications. In *2022 International Conference for Advancement in Technology (ICONAT)* (pp. 1–11). IEEE.
[25] Sreeparvathy, S., George, R., Kuruvithadam, R. M., & Nalesh, S. (2021, December). Fast Booth Multipliers Using Approximate 4:2 Compressors. In *2021 IEEE International Symposium on Smart Electronic Systems (iSES)* (pp. 157–160). IEEE.
[26] Han, J., & Orshansky, M. (2013, May). Approximate computing: An emerging paradigm for energy-efficient design. *In 2013 18th IEEE European Test Symposium (ETS)* (pp. 1–6). IEEE.
[27] Liang, J., Han, J., & Lombardi, F. (2012). New metrics for the reliability of approximate and probabilistic adders. *IEEE Transactions on Computers*, 62(9), 1760–1771.
[28] Park, J., Choi, J. H., & Roy, K. (2009). Dynamic bit-width adaptation in DCT: An approach to trade off image quality and computation energy. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(5), 787–793.
[29] Chippa, V. K., Chakradhar, S. T., Roy, K., & Raghunathan, A. (2013, May). Analysis and characterization of inherent application resilience for approximate computing. In *Proceedings of the 50th Annual Design Automation Conference* (pp. 1–9).
[30] Venkatachalam, S., & Ko, S. B. (2017). Design of power and area efficient approximate multipliers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(5), 1782–1786.
[31] Venkatachalam, S., Lee, H. J., & Ko, S. B. (2018, May). Power efficient approximate booth multiplier. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–4). IEEE.
[32] Kong, T., & Li, S. (2021). Design and analysis of approximate 4–2 compressors for high-accuracy multipliers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(10), 1771–1781.
[33] Mazloum, J., & Siahkal-Mahalle, B. H. (2023). Improving the accuracy of approximate multipliers based on the characteristics of 4:2 compressors. *Microprocessors and Microsystems*, 98, 104804.
[34] Weinberger, M. J., Seroussi, G., & Sapiro, G. (2000). The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Transactions on Image Processing*, 9(8), 1309–1324.
[35] Chandrasekharan, A., Soeken, M., Große, D., & Drechsler, R. (2016, November). Approximation-aware rewriting of AIGs for error tolerant applications. In 2016 *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (pp. 1–8). IEEE.
[36] Venkataramani, S., Sun, X., Wang, N., Chen, C. Y., Choi, J., Kang, M., Agarwal, A., Oh, J., Jain, S., Babinsky, T., & Cao, N. (2020). Efficient AI system design with cross-layer approximate computing. *Proceedings of the IEEE*, 108(12), 2232–2250.
[37] Li, C., & Bovik, A. C. (2010). Content-partitioned structural similarity index for image quality assessment. *Signal Processing: Image Communication*, 25(7), 517–526.
[38] Li, C., & Bovik, A. C. (2011). Three-component weighted structural similarity index. *IEEE Transactions on Image Processing*, 20(5), 1123–1132. https://doi.org/10.1109/TIP.2010.2092435
[39] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. https://doi.org/10.1109/TIP.2003.819861