Summary of Research Paper

Underwater Image Enhancement Using FPGA-Based Gaussian Filters with Approximation Techniques
Summary Date: August 1, 2025

---

1. **Introduction / Abstract**
   The paper addresses the problem of poor visual quality in underwater images caused by natural phenomena like light absorption and scattering, which create haziness and noise. To improve these images, the study proposes an efficient method for noise reduction using **Gaussian filters** (a mathematical tool for smoothing images and reducing noise) implemented on **FPGA** (Field-Programmable Gate Array, a type of reconfigurable hardware that allows fast and parallel processing). The key innovation is a pipeline architecture combined with approximate computing techniques that use simplified adders to speed up processing and reduce power consumption. The main findings show over 150% speed improvement and more than 34% power savings, with some increase in hardware area. This trade-off is suitable for error-tolerant applications like image and video processing.

2. **Methodology**
   The researchers designed a pipeline Gaussian filter architecture on an FPGA platform to enhance underwater images corrupted by Gaussian noise. The pipeline divides the filtering process into stages, allowing simultaneous processing and faster throughput. They introduced approximate adders—simplified arithmetic units that trade some accuracy for lower power and faster operation—into the filter design. Ten different approximate adder designs were tested, focusing on approximating the least significant bits to balance quality and efficiency. The system was simulated in MATLAB with real underwater images, and the designs were synthesized and tested on an Intel MAX10 FPGA device to measure power, speed, and area.

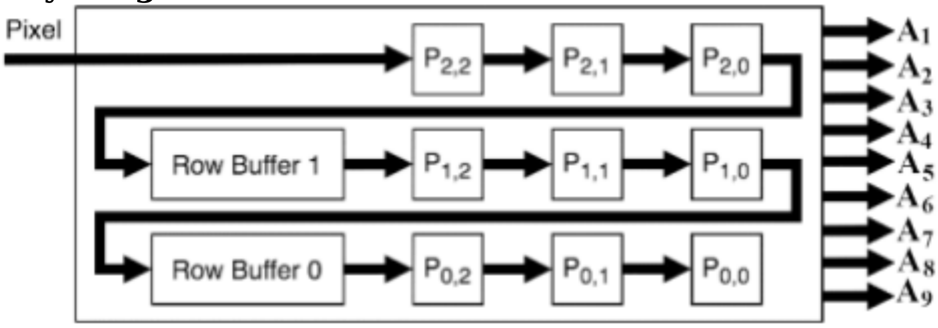
3. **Theory / Mathematics**
   The Gaussian filter applies a two-dimensional Gaussian function to smooth images by reducing high-frequency noise while preserving important features like edges. The Gaussian function is defined as:

[f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}]

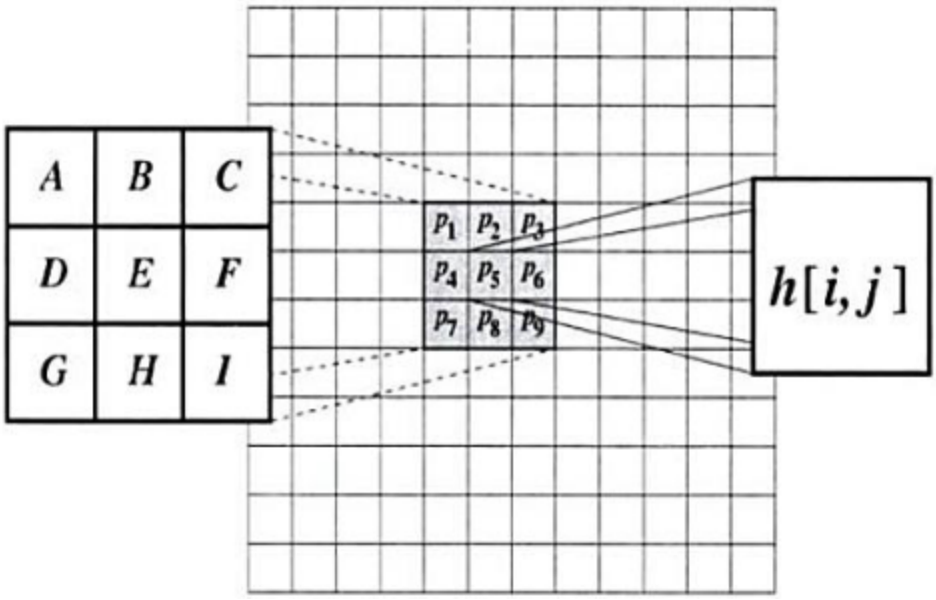
where (\sigma) controls the spread of the smoothing effect. The filter is implemented as a convolution operation using a weighted kernel (e.g., 3×3 matrix) that moves over the image pixels. The pipeline structure breaks this convolution into stages to improve speed. Approximate adders simplify the arithmetic by reducing transistor counts and logic complexity, allowing faster and more power-efficient addition at the cost of some errors, which are acceptable in error-resilient applications like image processing. Image quality was quantitatively evaluated using metrics such as Peak Signal-to-

Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Mean Square Error (MSE), which measure fidelity, perceptual similarity, and distortion respectively.

1. **Key Diagrams or Visual Elements**

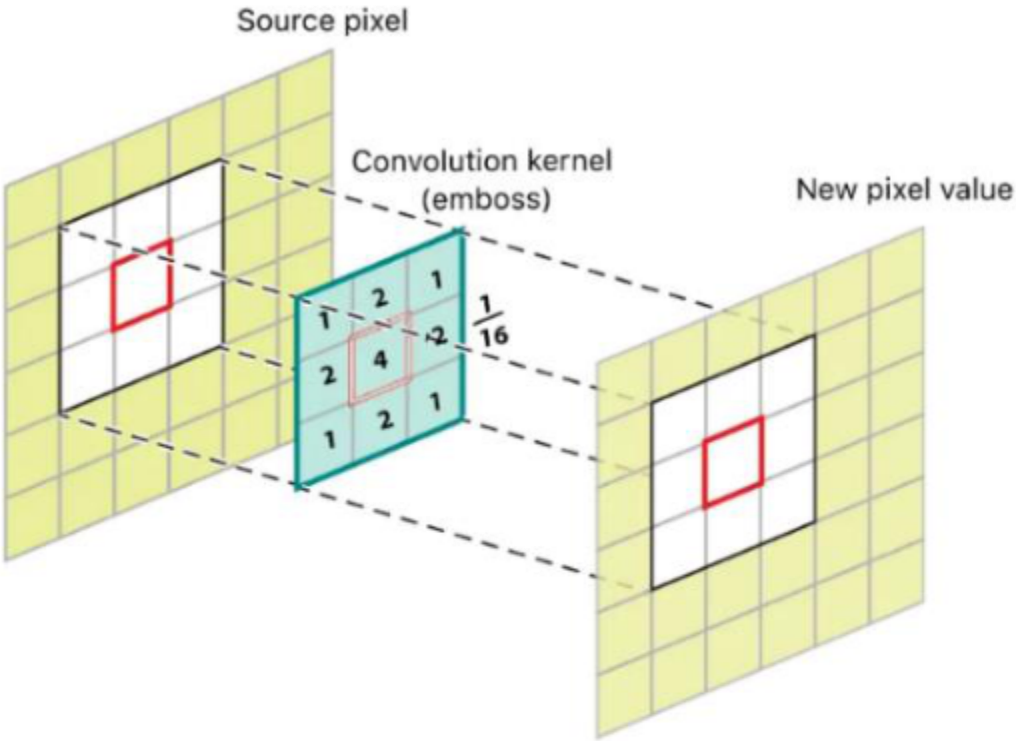


2. **Figure 1:** Shows the delay line buffer structure, which stores recent pixel values to optimize memory access during convolution, minimizing repeated reads from memory.

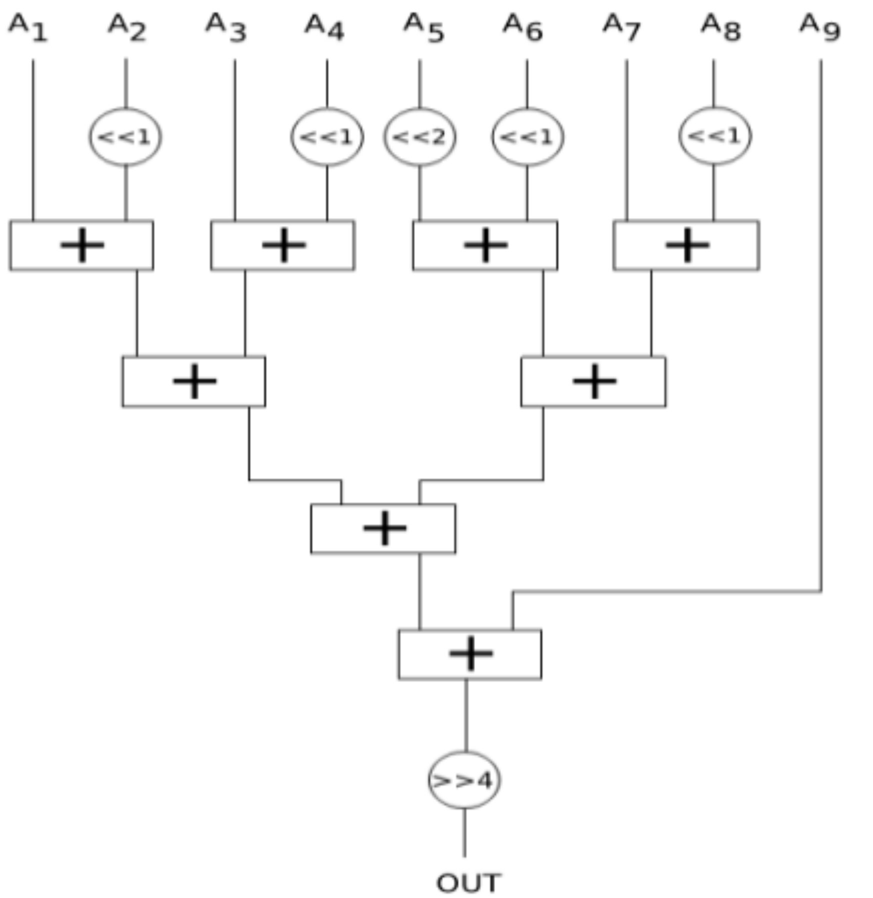


3. **Figure 2:** Illustrates the convolution operation with a 3×3 Gaussian kernel applied to an image region, explaining how pixel values are

weighted and summed.

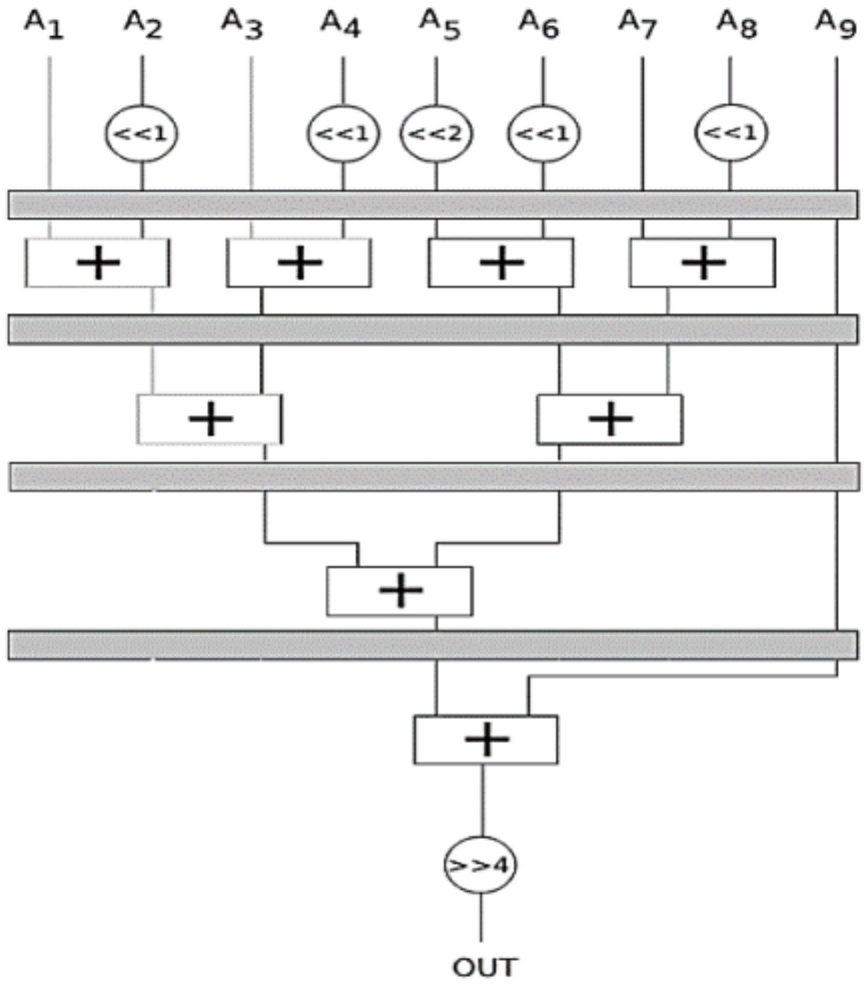


4. **Figure 3:** Displays the weighted Gaussian kernel used for filtering, highlighting the importance of central pixels.

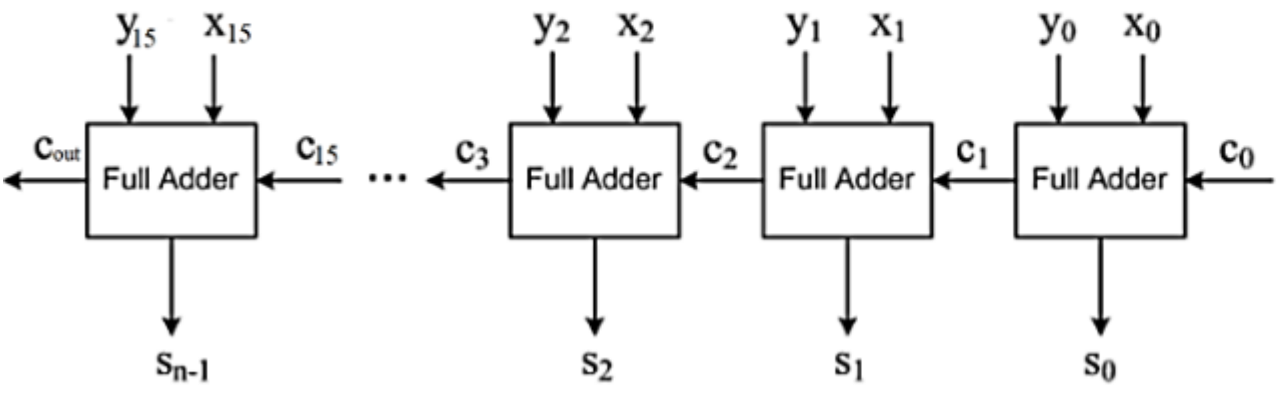


5. **Figure 4:** Block diagram of the Gaussian filter implementation, showing the use of adders and shifters for multiplication and division

by powers of two.

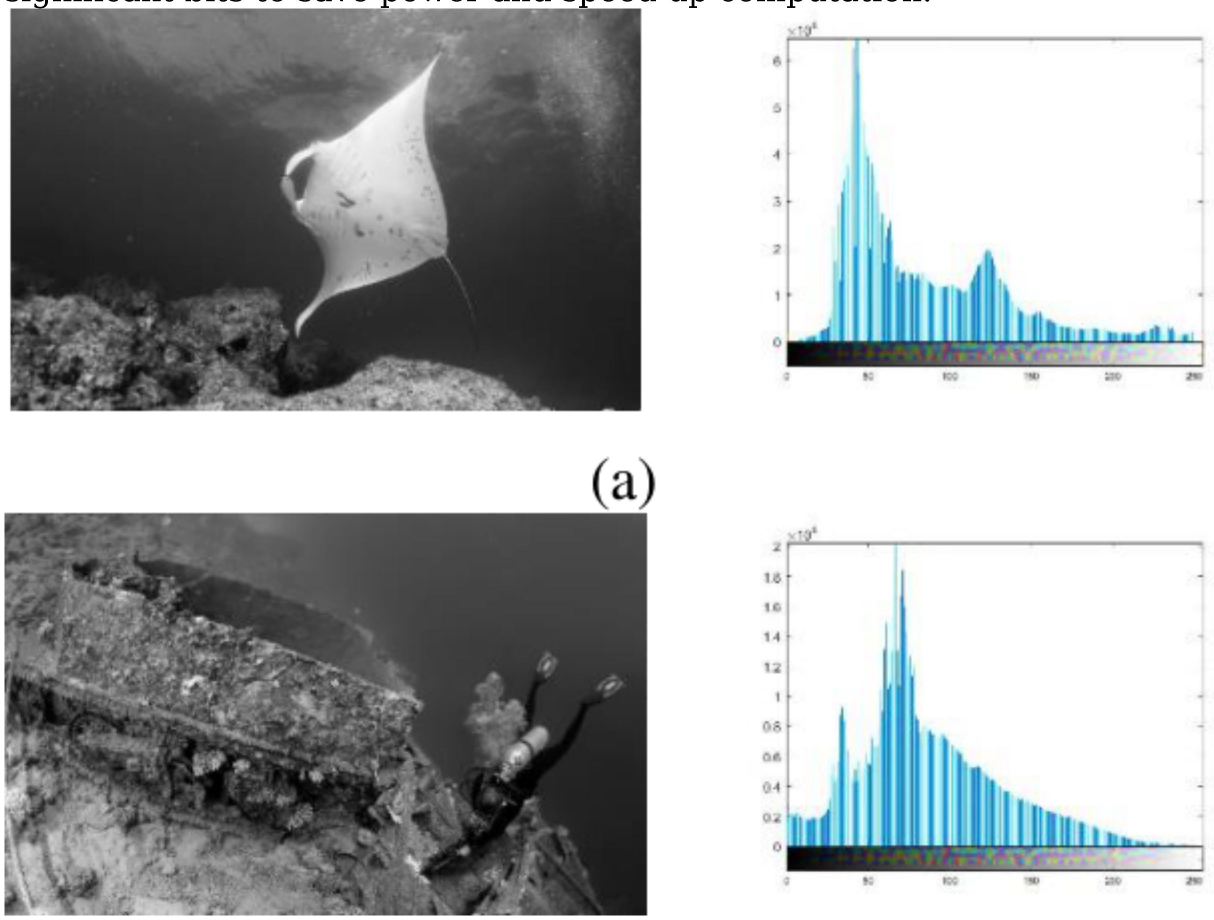


6. **Figure 5:** Depicts the pipeline Gaussian filter architecture with four stages, enabling concurrent processing to increase speed.



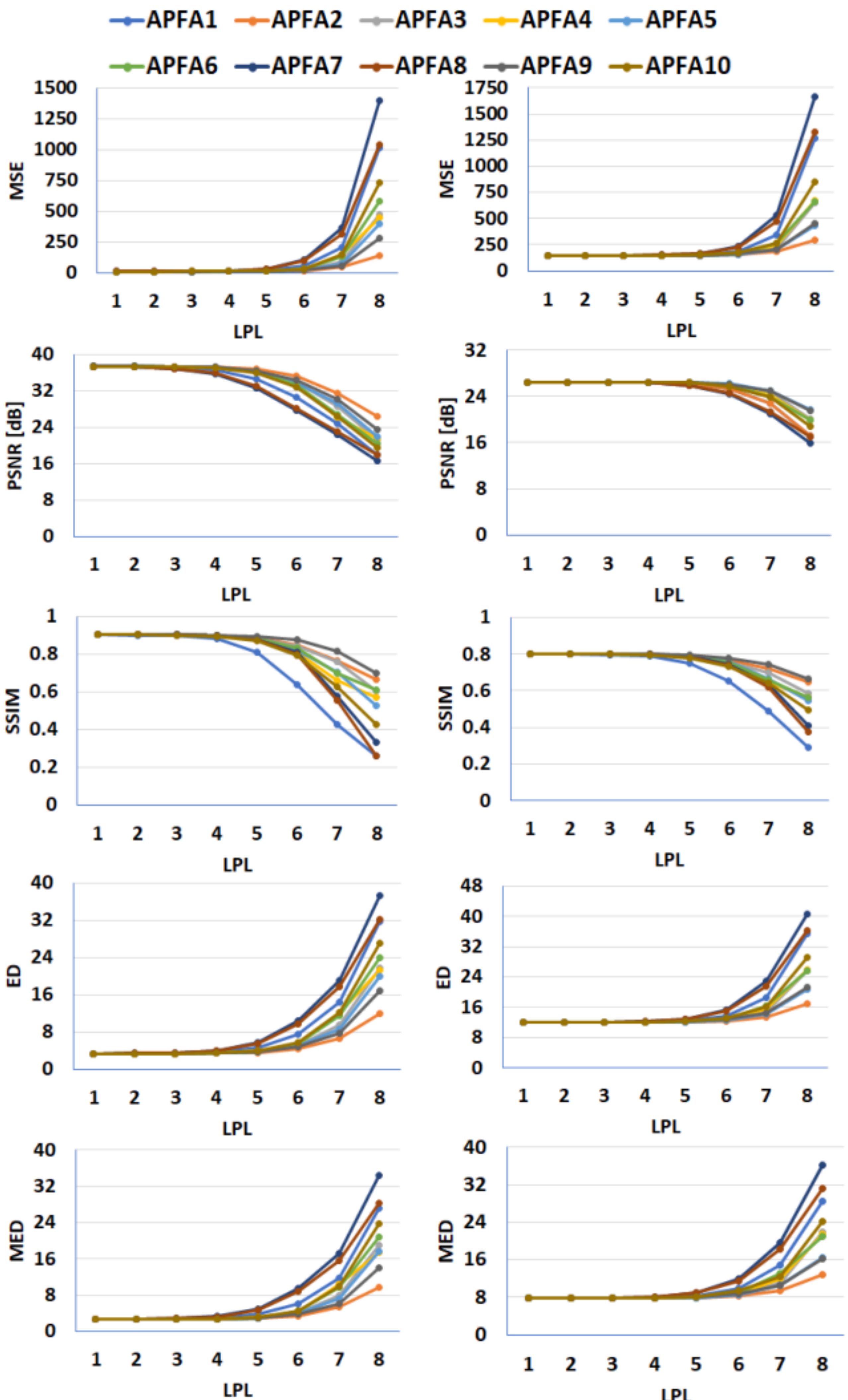
7. **Figure 6:** Shows the 16-bit carry ripple adder structure used for addition in the filter, where approximate adders modify the least

significant bits to save power and speed up computation.

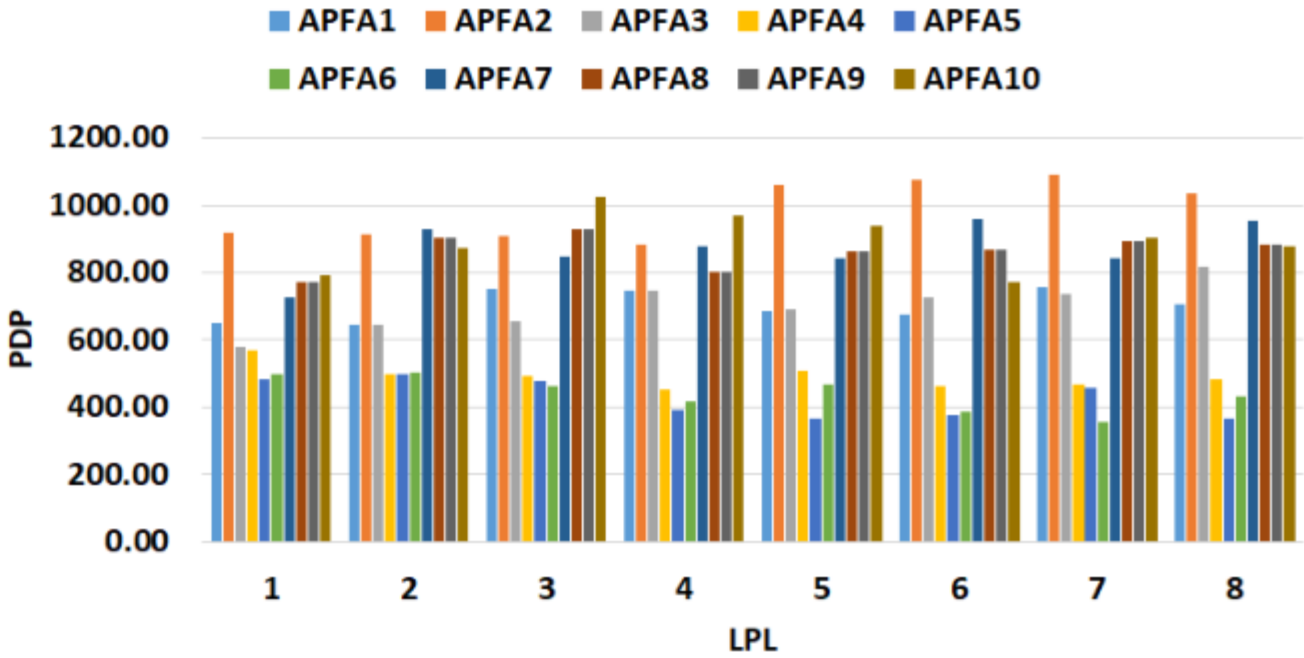


8. **Figure 7:** Presents two sample raw underwater images used for testing, illustrating typical underwater image challenges like low

contrast and noise.


APFA1
APFA2
APFA3
APFA4
APFA5
APFA6
APFA7
APFA8
APFA9
APFA10
MSE
PSNR [dB]
SSIM
ED
MED
LPL

9. **Figure 8:** Graphs evaluation metrics (PSNR, SSIM, etc.) showing that approximate adders up to 5 bits maintain good image quality, while more approximation degrades results.



10. **Figure 9:** Compares the Power-Delay Product (PDP) of different approximate adders, identifying the most power-efficient and fastest designs.

11. **Tables 1-3:** Provide detailed logic functions of approximate adders, and synthesis results comparing power, speed, and area for various filter implementations.

12. **Conclusion**
    The study successfully demonstrates that implementing a pipeline Gaussian filter on FPGA with approximate adders significantly improves processing speed (over 150%) and reduces power consumption (over 34%) for underwater image enhancement. Although this comes with increased hardware area, the trade-off is justified in error-tolerant applications like image and video processing where some loss in precision is acceptable. This approach offers a practical solution for real-time underwater image enhancement, enabling better visual quality with efficient hardware use, which is crucial for marine exploration, monitoring, and resource management.

Why It Matters:
Improving underwater image quality in real-time with low power and high speed enables more effective marine research, environmental monitoring, and underwater vehicle navigation. The proposed FPGA-based approximate Gaussian filter architecture provides a scalable and efficient hardware solution that balances image quality with resource constraints, advancing the capabilities of underwater imaging systems.