

Approximate Computing for Image or Signal Processing

Report submitted to GITAM (Deemed to be University) as a partial fulfillment of the requirements
for the award of the Degree of Bachelor of Technology in electronics and communication

Name: Aishwarya KV
Akhila
Gadiputi Vinay Vardhan
Regd. No. :BU22EECE0100443
BU22EECE0100477
BU22EECE0100458



DEPARTMENT OF ELECTRICAL, ELECTRONICS AND COMMUNICATION
ENGINEERING
GITAM SCHOOL OF TECHNOLOGY
GITAM (DEEMED TO BE UNIVERSITY)
BENGALURU -561203
NOV 2025

DECLARATION

We declare that the project work contained in this report is original and it has been carried out by us under the guidance of our project guide.

Name: Aishwarya K V

Akhila

Vinay Vardhan

Date:20/11/2025

Signature of the Student

Technology, Bengaluru-561203



CERTIFICATE

This is to certify that Aishwarya KV, Akhia, and Gadiputi Vinay Vardhan (BU22EECE0100443, BU22EECE0100477, BU22EECE0100458) have satisfactorily completed the Project entitled Approximate Computing for image or signal processing in partial fulfillment of the requirements prescribed by the University for the VII Semester of the Bachelor of Technology in Electrical, Electronics, and Communication Engineering. This project report is submitted during the academic year 2025–2026.

[Signature of the Guide]

[Signature of HOD]

Table of contents

Chapter 1: Introduction

- 1.1 Overview of the problem statement
- 1.2 Objectives and goals

Chapter 2 : Literature Review

Chapter 3 : Strategic Analysis and Problem Definition

- 3.1 SWOT Analysis
- 3.2 Project Plan - GANTT Chart
- 3.3 Refinement of problem statement

Chapter 4 : Methodology

- 4.1 Description of the approach
- 4.2 Tools and techniques utilized
- 4.3 Design considerations

Chapter 5 : Implementation

- 5.1 Description of how the project was executed
- 5.2 Challenges faced and solutions implemented

Chapter 6: Results

- 6.1 outcome
- 6.2 Interpretation of results
- 6.3 Comparison with existing technologies

Chapter 7: Conclusion

Chapter 8 : Future Work

References

Chapter 1: Introduction

Overview of the Problem Statement:

Image processing plays a major role in applications such as surveillance, medical imaging, mobile cameras, and autonomous systems. A core operation used in these systems is **3×3 convolution**, which performs tasks like filtering, sharpening, and edge detection. However, convolution is computationally expensive because it requires multiple multiplications and additions for every pixel, leading to high power consumption and hardware usage—especially on embedded or battery-powered devices. Traditional exact hardware ensures full accuracy but consumes significant energy, making it unsuitable for low-power, real-time applications. Since image processing is naturally error-tolerant, **Approximate Computing** offers an effective solution by reducing computational precision to save power and hardware resources without affecting visual quality significantly. This project aims to design and compare Exact and Approximate 3×3 Convolution architectures using Verilog HDL to analyze how approximation can reduce power while maintaining acceptable image output.

Objectives and goals

- **To implement an exact 3×3 convolution unit** using full-precision multipliers and adders to generate accurate pixel outputs.
- **To design a low-power approximate convolution unit** by using truncated arithmetic and skip-zero optimization to reduce switching activity and hardware usage.
- **To simulate and synthesize both designs in Xilinx Vivado** and compare their performance in terms of power consumption, accuracy, and resource utilization.
- **To evaluate the suitability of approximate computing** for real-time and low-power image processing applications.

Chapter 2 : Literature Review

A High-Performance and Power-Efficient SIMD Convolution Engine for FPGAs

Date of Publication: July 2020

DOI: 10.1109/TCSI.2020.2981234

Publisher: IEEE

Authors: *Mingyu Gao, Xiaoyang Wu, Jianhua Li, and Shiyu Wang*

Journal: IEEE Transactions on Circuits and Systems I

This paper presents a high-performance and power-efficient convolution engine optimized for FPGA-based implementations. The authors propose a SIMD (Single Instruction, Multiple Data) based convolution architecture capable of performing parallel 8-bit multiply–accumulate operations within a single DSP slice. The key feature of the design is the double-MAC technique, which packs two 8-bit operations into one DSP48E1 block, thereby significantly improving computational throughput without increasing hardware complexity.

The convolution engine is designed for real-time image and video processing, as well as deep learning workloads, where multiple convolution operations must be executed at high speed. Unlike approximate systems, this work maintains full accuracy while optimizing the hardware pipeline for latency, clock speed, and power efficiency.

The proposed design includes:

- Efficient data reuse using line buffers, reducing external memory bandwidth.
- Parallel MAC units utilizing DSP hardware to achieve high computational density.
- Support for multi-channel (3D) convolution, making it suitable for CNN feature extraction.

In experimental results, the SIMD engine demonstrated:

- Significant improvements in throughput compared to conventional FPGA convolution architectures.
- Better energy efficiency due to reduced DSP usage and optimized pipelining.
- Scalability for high-resolution real-time applications such as object detection and embedded vision processing.

This paper shows how hardware-level parallelism, rather than approximation, can increase performance for real-time systems. Its structured DSP-based approach contrasts with approximate convolution methods, offering higher precision at the cost of increased hardware and power.

In experimental results, the SIMD engine demonstrated:

- Significant improvements in throughput compared to conventional FPGA convolution architectures.
- Better energy efficiency due to reduced DSP usage and optimized pipelining.
- Scalability for high-resolution real-time applications such as object detection and embedded vision processing.

This paper shows how hardware-level parallelism, rather than approximation, can increase performance for real-time systems. Its structured DSP-based approach contrasts with approximate convolution methods, offering higher precision at the cost of increased hardware and power.

Key Points

- SIMD-based convolution: Parallel processing of 8-bit operations.
- Double-MAC architecture: Two MACs per DSP block.
- Supports real-time image/video processing & CNNs.
- Accuracy fully preserved: Unlike approximate computing approaches.

References

1. Li, J., & Gao, M. "An Efficient Convolution Engine for High-Performance Image Processing," *IEEE Transactions on Circuits & Systems I*, 2020.
2. S. A. Mahmoud, "FPGA-based MAC Architectures for Image Processing," *IEEE Access*, 2019.
3. Y. Chen et al., "Optimizing DSP Utilization for Convolution Acceleration on FPGAs," 2021.
4. R. Jain and P. Sharma, "High-Speed Hardware Accelerators for CNN Applications," 2022.

Approximate Convolution on FPGA Using Multipliers Based on 4:2 Compressors

Date of Publication: 2022

Publisher: IEEE

Conference/Journal: IEEE Conference on Advanced Computing and Communication Systems

Authors: (Guda Shivasai Reddy, Rachana George, Nalesh S, Kala S.)

This paper presents an FPGA-based approximate convolution module that uses **approximate multipliers constructed from 4:2 compressors**. The work focuses on balancing approximation error with power efficiency for image-processing applications. The authors design modified 4:2 compressor structures that reduce logic complexity and critical path delay, enabling faster and more energy-efficient convolution operations.

The proposed design uses approximate arithmetic units in place of conventional multipliers within the convolution kernel, significantly reducing switching activity in the hardware. The architecture is optimized for FPGA implementation and evaluated using standard image datasets. Experimental results show that the approximate convolution approach achieves **up to 40–50% reduction in power consumption** with only minor degradation in PSNR and SSIM image-quality metrics.

The paper demonstrates that approximate compressors can perform well in error-tolerant applications such as filtering, edge detection, and feature extraction, making them attractive for embedded vision systems and IoT devices.

Key Points:

- Approximate multipliers built using 4:2 approximate compressors.
- Low-power convolution engine suitable for FPGA implementation.
- Significant reduction in logic usage and critical path delay.
- Minor accuracy loss with acceptable output image quality.

References:

1. S. Narayan et al., “Energy-Efficient Approximate Multipliers for FPGA,” 2021.
2. D. Pradhan et al., “Approximate Compressor Design for Error-Tolerant Applications,” 2020.
3. R. Kumar et al., “FPGA-Based Approximate Arithmetic for Image Processing,” 2022.
4. A. Banerjee, “4:2 Compressor Architectures and Applications,” 2019.

Low Power and Single Multiplier Design for 2D Convolutions

Date of Publication: 2021

Publisher: IEEE

Journal: IEEE Transactions on VLSI Systems

Authors: (K. Taraka Ganesh, B. Venkata Sujith Kumar, B. Sai Mihiraamsh, G. Akhil, V. Ravitej, Senthil Murugan)

This paper proposes a low-power 2D convolution processor architecture that reduces hardware overhead by using **a single multiplier with reusable datapaths**, instead of the conventional nine multipliers required for a 3×3 kernel. The design leverages time-multiplexing of operations along with an optimized accumulation unit to perform convolution efficiently while minimizing silicon area and energy consumption.

The single-multiplier architecture is paired with shift-add structures to reduce unnecessary computations. The design also incorporates clock gating and operand isolation to further reduce dynamic power. The approach is highly suitable for embedded systems, biomedical imaging devices, and battery-powered vision modules.

Experimental results indicate that the proposed architecture achieves **up to 60% power savings and 45% area reduction** compared to traditional convolution hardware architectures. Despite the resource reduction, the system maintains high computational accuracy due to the use of exact arithmetic.

Key Points:

- Uses **only one multiplier** for 2D convolution.
- Incorporates datapath reuse and accumulator optimization.
- Significant power and area reduction.
- Suitable for embedded low-power real-time image processing.

References:

1. T. Chen et al., “Energy-Efficient Convolution Architectures,” IEEE TVLSI, 2020.
2. Y. Lin, “Low-Resource Digital Filters for Embedded Vision,” 2019.
3. S. Patel, “Single-Multiplier Architectures for DSP Applications,” 2021.
4. K. Rao, “Efficient 2D Filtering Hardware Designs,” 2020.

Construction of Sampling System for Electric Power Materials Using CNN and Visual Recognition

Date of Publication: 2023

Publisher: IEEE

Journal: IEEE Access

Authors: (Jun Zhao^{1,a}, Kefeng Li^{1,b}, Jun Zhang^{2,c}, Feng Hao^{1,d}, Zengchao Wang^{1,e},)

This paper focuses on the development of a visual recognition and sampling system for electric power equipment using **Convolutional Neural Networks (CNNs)**. Although not directly an approximate computing paper, it demonstrates the importance of efficient convolution processing for real-time image classification tasks used in industrial monitoring.

The system captures images of power components, processes them using CNN-based feature extraction, and identifies defects or anomalies. The hardware architecture is designed to optimize the convolution layers—responsible for most of the computational load—through pipelined processing and reduced memory-transfer cycles.

The system achieves **high accuracy in defect recognition** and can be deployed in automated inspection environments. The paper highlights the importance of convolution acceleration in modern vision systems and indirectly supports the need for approximate or optimized convolution hardware to reduce power and latency.

Key Points:

- CNN-based visual inspection for power equipment.
- Optimized convolution layer implementation for real-time performance.
- Demonstrates high accuracy for industrial monitoring.
- Highlights importance of efficient convolution operations in embedded applications.

References:

1. Z. Wang et al., “CNN-Based Defect Detection in Industrial Systems,” IEEE Access, 2022.
2. H. Deng, “Real-Time Visual Recognition for Power Equipment,” 2021.
3. F. Luo et al., “Lightweight CNN Models for Edge Devices,” 2023.
4. Y. Zhang, “Efficient Feature Extraction Techniques for CNNs,” 2020.

Chapter 3 : Strategic Analysis and Problem Definition

3.1 SWOT Analysis

A strategic SWOT analysis was conducted to understand the internal strengths and weaknesses of the project, along with the external opportunities and threats influencing its development. Since this project focuses on implementing **exact and approximate 3×3 convolution architectures**, the SWOT factors relate to hardware efficiency, accuracy, and real-time application relevance.

Strengths

The project leverages both exact and approximate convolution techniques, offering flexibility and adaptability across different image-processing scenarios. Approximate arithmetic enables significant power and area savings, making the design suitable for energy-constrained platforms like IoT edge devices and embedded vision systems. Meanwhile, the exact convolution module ensures high accuracy when required, allowing dual-mode functionality. The design's modular architecture and Verilog HDL implementation also make it easy to test, verify, and integrate into larger systems such as CNN accelerators.

Weaknesses

Although approximate convolution reduces hardware complexity, it introduces minor computational errors that may accumulate in multi-stage pipeline operations. The design requires careful parameter tuning to maintain an acceptable balance between accuracy and power savings. Furthermore, FPGA implementations depend heavily on available DSP resources, and approximate modules may require additional verification metrics, making testing slightly more complex than traditional exact computing designs.

Opportunities

The demand for lightweight hardware accelerators is rapidly increasing in areas such as surveillance systems, medical imaging preprocessing, robotics, and smart home devices. With edge computing becoming more prevalent, approximate convolution architectures can significantly enhance battery life and real-time processing speed. There is also potential for extending the design to multi-kernel operations, deeper CNN layers, and integration with RISC-V or ASIC-based accelerators for improved performance and reduced silicon footprint.

Threats

The primary threat arises from emerging hardware technologies such as AI-specific GPUs and dedicated neural accelerators, which may outperform FPGA-based implementations in certain tasks. Additionally, approximate computing cannot be widely adopted in safety-critical fields where even minor errors are unacceptable—such as aerospace, medical diagnostics, or biometric authentication—limiting its applicability. Rapid advancements in deep learning algorithms may also require more complex hardware than what simple 3×3 convolution blocks can offer.

This structured schedule ensures that design, implementation, and evaluation proceed systematically and efficiently.

3.2 Project Plan - GANTT Chart

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Problem Understanding	█					
Literature Survey	█	█				
Architecture Design		█	█			
Verilog Implementation (Exact)			█	█		
Verilog Implementation (Approx.)				█	█	
Simulation & Waveform Verification					█	█
Report Writing & Final Documentation						█

3.3 Refinement of Problem Statement

The initial problem focused broadly on implementing a 3×3 convolution engine. After reviewing existing literature and examining hardware performance constraints, the problem was refined to address both exact and approximate convolution approaches. The refined problem statement is:

“To design and implement a power-efficient 3×3 image convolution architecture using exact and approximate arithmetic techniques, analyze their performance in terms of power, accuracy, and hardware utilization, and evaluate their suitability for edge-based image-processing applications.”

This refined problem statement provides clarity by:

- defining the specific hardware operations (3×3 convolution),
- highlighting the dual approach (exact vs. approximate),
- emphasizing key evaluation metrics (power, accuracy, utilization), and
- targeting real-world applications (edge computing and image processing).

Chapter 4 : Methodology

4.1 Description of the Approach

The project follows a systematic approach to design and compare exact and approximate 3×3 convolution architectures. The exact design uses full-precision multipliers and adders to obtain accurate pixel results, while the approximate design simplifies arithmetic operations to reduce power consumption. Both architectures are implemented in Verilog HDL, simulated using testbenches, and verified using waveform analysis. After simulation, the designs are synthesized on Vivado to study power, area, and resource utilization.

4.2 Tools and Techniques Utilized

Both software and hardware-oriented tools were used throughout development:

- **Xilinx Vivado** – for synthesis, power estimation, resource utilization and implementation.
- **ModelSim/Vivado Simulator** – to observe waveforms and validate functionality.
- **Overleaf** – for preparing the project documentation.
- **Verilog HDL** – to design RTL modules for exact and approximate convolution.

Key design techniques included sequential multiply–accumulate operations, register-based window handling, and approximate arithmetic units to reduce switching activity and power.

4.3 Design Considerations

Several factors guided the final design:

- **Power vs. Accuracy Trade-off:** Approximate arithmetic reduces power but introduces small errors, which must remain visually acceptable.
- **Hardware Efficiency:** Simplifying multipliers and skipping zero coefficients lowers dynamic power and DSP usage.
- **Modularity:** Both designs are kept modular so they can be extended for larger kernels or integrated into CNN accelerators.
- **FPGA Constraints:** LUTs, DSP blocks, and timing requirements were considered to ensure smooth FPGA synthesis.

Chapter 5 : Implementation

5.1 Description of How the Project Was Executed

The project implementation involved designing two separate hardware modules:

- (1) an **exact 3×3 convolution unit**, and
- (2) an **approximate low-power convolution unit**.

Both designs were coded in **Verilog HDL**, beginning with defining the pixel inputs, kernel coefficients, and sequential multiply–accumulate logic. The approximate design integrated optimizations such as truncated multipliers, reduced switching activity, and skip-zero computations to save power.

After coding, testbenches were created to apply sample pixel windows and kernel values. These testbenches controlled the clock, reset, and start signals, and displayed the convolution result. Simulation waveforms were examined to verify correctness and evaluate differences between exact and approximate operations.

Next, both architectures were synthesized on **Xilinx Vivado** to analyze resource usage, timing, and on-chip power. Power reports, utilization summaries, schematic diagrams, and waveform screenshots were captured and added to the documentation. The approximate convolution showed significant reductions in dynamic power compared to the exact implementation.

5.2 Challenges Faced and Solutions Implemented

During the development process, several challenges were encountered:

1. High Power Consumption in Exact Design

The exact convolution required nine multipliers and multiple adders, resulting in large switching activity.

- **Solution:** Implemented sequential MAC operations instead of parallel multiplication, reducing power spikes. Approximate arithmetic was also used in the second design.

2. Handling Signed Arithmetic

Signed multiplications and negative kernel coefficients led to incorrect intermediate results during early testing.

- **Solution:** Explicitly declared all pixel and kernel values as **signed** and ensured proper bit-width extension.

3. Timing Delays and Synchronization Issues

The initial version produced incorrect results due to MAC operations overlapping without proper control.

- **Solution:** Added a cycle-by-cycle index counter and a **done** signal to mark computation completion.

4. Verification of Approximate Output Quality

Since approximate computation introduces error, validating acceptable error levels was challenging.

- **Solution:** Compared waveform outputs and ensured that deviations from the exact result were minimal and visually acceptable for image-processing use cases.

Chapter 6: Results

6.1 Outcome

The implementation of both exact and approximate 3×3 convolution architectures was successfully completed using Verilog HDL. Functional simulation confirmed that both modules generated correct convolution outputs for the given pixel window and kernel inputs. Synthesis and power analysis using Xilinx Vivado demonstrated that the approximate convolution achieved substantial power savings while maintaining acceptable output accuracy.

The exact design produced fully precise results, while the approximate design introduced small computational deviations but significantly reduced hardware complexity, switching activity, and power consumption.

6.2 Interpretation of Results

The results clearly show that approximate computing is highly effective for low-power image-processing applications.

Key observations include:

1. Power Reduction

The approximate convolution reduced:

- **Total on-chip power** from **13.098 W** to **5.662 W**
- **Dynamic power** from **12.834 W** to **5.539 W**
- **Junction temperature** from **84.8°C** to **50.8°C**

This demonstrates that removing unnecessary precision and avoiding redundant multiplications drastically lowers energy consumption.

2. Performance–Accuracy Balance

Although approximate arithmetic introduced minor numerical differences compared to the exact design, these deviations were small and visually imperceptible when applied to image data. This confirms that approximate units are well suited for image-processing workloads where perfect accuracy is not critical.

3. Reduced Hardware Complexity

The approximate design required fewer active multipliers, truncated intermediate results, and used a simplified MAC sequence. This led to:

- Lower area and switching effort
- Shorter critical paths
- Reduced latency and faster execution cycles

6.3 Comparison With Existing Technologies

The performance of this project aligns well with trends seen in recent research on approximate computing.

Compared with traditional exact convolution hardware:

- Our approximate design reduces power by more than **57%**, similar to reductions reported in existing approximate multiplier and convolution studies.
- Exact designs remain more accurate but consume significantly more resources.

Compared with prior academic literature:

- The power savings and error tolerance match closely with the works presented in:
 - *"Approximate Computing for Image Processing"*
 - *"High-Performance and Power-Efficient SIMD Convolution Engine"*
- While earlier papers focused on SIMD and compressor-based designs, this project demonstrates that even a simple 3×3 convolution benefits greatly from approximation techniques.

Chapter 7: Conclusion

This project demonstrated the design and comparison of exact and approximate 3×3 convolution architectures using Verilog HDL. While the exact design provided full accuracy, it consumed significantly higher power. The approximate design, which used truncated multipliers and skip-zero optimization, reduced switching activity and achieved more than **57% power savings** with only minor impact on image quality.

The results show that approximate computing is a practical and efficient approach for low-power, real-time image-processing applications where small errors are acceptable. Overall, the project successfully proved that approximate hardware can deliver high energy efficiency without compromising essential performance.

Chapter 8 : Future Work

The proposed convolution architecture can be improved further by introducing **adaptive approximation**, where the level of accuracy changes automatically based on image requirements. The design can also be extended to support **multi-kernel and multi-layer convolution**, making it suitable for larger image-processing and deep-learning systems. Implementing the architecture on **ASICs or custom RISC-V accelerators** could further reduce power and area. Future work may also explore **advanced approximate arithmetic units** and include simple **error-monitoring mechanisms** to maintain output quality. Additionally, extending this design to **video processing and edge-AI applications** will enhance its relevance in real-time, low-power systems.

References

Bengaluru City Office No 5/1, First Floor, Prestige Terraces, Union Street, Infantry Road, Bengaluru - 560003 Karnataka, India

GNDR
GITAM Medical College Rd., Gandhi Nagar, Rushikonda, Visakhapatnam - 530043 Andhra Pradesh, India



