2D Convolution for Image Processing

AN INDUSTRIAL INTERNSHIP REPORT

submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL AND ELECTRONICS ENGINEERING

by

MOHAMMAD FAIZ ABID 22BEE1278

SCHOOL OF ELECTRICAL ENGINEERING

VIT CHENNAI

June 2024

CERTIFICATE

This is to certify that the Industrial Internship Programme titled **2D Convolution** submitted by ***MOHAMMAD FAIZ ABID*** is in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY**. The contents of this project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Internal Examiner1 **Internal Examiner2**

Approved by

Head of the Department
B.Tech Electrical and Electronics Engineering

ACKNOWLEDGEMENT

I express my heartfelt gratitude to Dr. Jayapragash R, Head of the Department (HoD), B.Tech Electrical and Electronics Engineering, SELECT, VIT Chennai, for his constant support and guidance throughout my academic journey. His encouragement and insightful suggestions have been instrumental in the completion of this certification.

I am deeply thankful to Dr. Senthil Kumar N, Dean of the School of Electrical Engineering (SELECT), VIT Chennai, for providing a conducive learning environment that has greatly enriched my knowledge and skills.

I would also like to extend my sincere thanks to Dr. Augusta Sophy Beulet P, my research internship guide, for her invaluable advice and academic support, which have motivated me to push my limits and strive for excellence.

Furthermore, I owe a great deal of gratitude to my parents and friends for their unwavering support, motivation, and understanding during this journey. Their constant encouragement has been a driving force behind my achievements.

(Signature of the student)

Name of the Student

(Reg. Number)

ABSTRACT

This paper explores the design and implementation of a 2D convolution operation for image processing, specifically for horizontal edge detection, using hardware description language (Verilog) and MATLAB simulation. Convolution, a critical operation in image processing, allows for the extraction of meaningful features such as edges by applying a 3x3 kernel matrix to an input image. Here, a 20x20 image is processed using a horizontal edge-detection kernel, with the output provided in both integer and binary formats to facilitate analysis. The implementation includes the development of the Convolutor6 module in Verilog, optimized for efficient computation, and a corresponding testbench (tb_Convolutor6) to validate functionality. MATLAB simulations complement the hardware design, providing a visual representation of the convolution results and ensuring accuracy. This paper discusses design methodology, optimization strategies, and the hardware's performance, highlighting the advantages of hardware-accelerated image processing in real-time applications. The results demonstrate effective edge-detection capabilities, making this approach viable for embedded systems and hardware-based image processing solutions.

TABLE OF CONTENTS

ABBREVIATIONS AND NOMENCLATURE

2D - Two-Dimensional

CNVD - Convolution-based Network Design

FPGA - Field-Programmable Gate Array

HDL - Hardware Description Language

MATLAB - Matrix Laboratory

TB - Testbench

VLSI - Very Large Scale Integration

ASIC - Application-Specific Integrated Circuit

RTL - Register Transfer Level

HED - Horizontal Edge Detection

VHDL - VHSIC Hardware Description Language

CHAPTER I

1. INTRODUCTION

1.1 INTRODUCTION

This paper presents a detailed study on implementing 2D convolution for horizontal edge detection (HED), focusing on its applications in image processing. Leveraging Verilog and MATLAB, the study aims to bridge simulation and hardware implementation, providing insights into the accuracy and efficiency of convolution-based image processing.

1.1.1 Motivation

Edge detection is a vital step in image processing, supporting tasks like object recognition and scene understanding. Given the growing demand for real-time processing, this work seeks to create a hardware-efficient solution for edge detection, providing faster and more reliable results compared to software-based approaches.

1.1.2 Objectives

The main objective is to design, implement, and verify a Verilog-based 2D convolution module optimized for HED. Additionally, the study aims to validate its performance on FPGA-compatible hardware, ensuring compatibility, accuracy, and efficiency in real-world embedded applications.

1.1.3 Scope of the Work

This work focuses on hardware-accelerated 2D convolution, specifically for edge detection. It targets real-time image processing systems, providing a modular and scalable approach that could be adapted for various applications in embedded image processing and machine vision.

CHAPTER II

2. PROJECT DESCRIPTION

2.1 OVERVIEW OF PROJECT

The objective of this project was to develop a comprehensive solution for implementing 2D convolution, focusing on enhancing image processing capabilities. Convolution, a mathematical operation widely used in image processing, serves to filter and extract features from images. This project involved designing and implementing a 2D convolution algorithm, which was evaluated in both software and hardware environments using MATLAB and Verilog.

The autonomous implementation allowed for real-time image processing applications, such as edge detection and object recognition, which are crucial in various fields including robotics, computer vision, and digital imaging. By comparing the performance of software and hardware implementations, the project identified the optimal approach to 2D convolution for future applications, ensuring efficiency and accuracy in image processing tasks.

2.2. MODULES OF THE PROJECT

The project consisted of several key modules that encompassed the design, implementation, and evaluation phases of the 2D convolution algorithm. Each module focused on specific aspects of the project.

2.2.1 Software Implementation

In this module, the 2D convolution algorithm was developed using MATLAB. The primary tasks included:

- Designing the convolution algorithm to process input images with a specified kernel.
- Utilizing built-in MATLAB functions for image processing, ensuring accuracy and efficiency.

- Testing the algorithm with various test images to evaluate performance and results.

The outcomes of this module included a functional software application that demonstrated the capabilities of 2D convolution and allowed for visual inspection of the filtered output.

2.2.2 Hardware Implementation

This module focused on implementing the 2D convolution algorithm in Verilog, targeting an FPGA for hardware execution. The key activities included:

- Designing the Verilog code for the 2D convolution operation, ensuring it met the necessary specifications for processing images in real-time.
- Developing a testbench for the Verilog module to simulate its performance against various input images.
- Conducting optimization for resource utilization and execution speed to ensure the hardware implementation could handle real-time processing demands.

The successful completion of this module provided a hardware solution capable of performing 2D convolution, facilitating comparisons with the software implementation.

2.2.3 Performance Evaluation

This module focused on a detailed analysis of the results obtained from both software and hardware implementations. Key activities included:

- Comparing the execution time and resource utilization metrics of both implementations.
- Evaluating the accuracy of the convolution results against expected outcomes.
- Identifying strengths and weaknesses of each approach to provide insights into their applicability in different scenarios.

The findings from this module culminated in a comprehensive report that outlined the comparative performance of the software versus hardware implementations of 2D convolution.

2.3 TASKS AND MILESTONES

1. June 3 - June 9, 2024 (Day 1-7): Initial Planning and Requirements Gathering

- **Description**: Reviewed project objectives, analyzed requirements, selected tools (MATLAB and Verilog), and set clear deliverables.
- **Milestone**: Completed requirements analysis and established foundational project goals.

2. June 10 - June 17, 2024 (Day 8-15): Literature Review and Technical Research

- **Description**: Conducted research on 2D convolution techniques, edge detection, and relevant methodologies for efficient implementation.
- **Milestone**: Compiled resources and completed technical background documentation.

3. June 18 - June 27, 2024 (Day 16-25): Design Specification and Module Outlining

- **Description**: Created detailed design specifications, defined system architecture, and planned the functional modules for MATLAB and Verilog.
- **Milestone**: Finalized design specifications and module structure for implementation.

4. June 28 - July 9, 2024 (Day 26-37): Implementation - MATLAB Code

- **Description**: Developed and tested MATLAB code for the 2D convolution process, ensuring accurate image processing and kernel application.
- **Milestone**: Successfully implemented and verified MATLAB code for convolution.

5. July 10 - July 21, 2024 (Day 38-49): Implementation - Verilog Code

- **Description**: Created Verilog code for performing convolution on images, focused on handling image data and implementing edge detection algorithms.
- **Milestone**: Completed Verilog implementation and preliminary testing for accuracy.

6. July 22 - July 27, 2024 (Day 50-55): Testing and Validation

- **Description**: Conducted extensive testing for both MATLAB and Verilog implementations, validated outputs, and refined parameters for optimal performance.
- **Milestone**: Verified project functionality through successful testing of both codes.

7. July 28 - August 1, 2024 (Day 56-60): Documentation of Results

- **Description**: Documented results including output images, integer matrices, binary outputs, and explanations. Ensured comprehensive annotation.

- **Milestone**: Completed detailed documentation of project results.

8. August 2 - August 6, 2024 (Day 61-65): Final Edits and Presentation Preparation

- **Description**: Finalized the report, made revisions, and prepared presentation materials to effectively communicate project outcomes.
- **Milestone**: Report and presentation ready for submission.

9. August 7 - August 10, 2024 (Day 66-69): Submission and Project Wrap-up

- **Description**: Submitted all project components, completed final reviews, and participated in a debrief session with the internship guide.
- **Milestone**: Project successfully submitted and concluded.

CHAPTER III

3. DESIGN OF 2D CONVOLUTION SYSTEM FOR EDGE DETECTION

3.1 DESIGN APPROACH

The design approach for the 2D Convolution System focuses on implementing a robust and efficient system for edge detection in digital images. This project allows digital images to undergo a 2D convolution process using a predefined kernel, which highlights the edges in the image. The process involves the MATLAB implementation for initial prototyping and testing, followed by Verilog for hardware-level processing to support scalability and higher efficiency in real-time applications.

The overall design aims to create a pipeline that processes an input image, applies convolution, and outputs an edge-detected image in both integer and binary matrix formats. This approach ensures compatibility with various applications in image processing, particularly where edge detection is crucial, such as in autonomous systems, medical imaging, and object recognition.

3.1.1 Codes and Standards

This project adheres to industry codes and standards relevant to digital image processing and hardware description languages to ensure accuracy, reproducibility, and compatibility with existing technologies. The following standards and guidelines are referenced:

- **IEEE Standard for Verilog Hardware Description Language (IEEE 1364)**: Ensures that the Verilog code is compatible with standardized simulation and synthesis tools, which aids in the seamless development and testing of digital circuits.
- **ISO/IEC 15444-1 (JPEG 2000)**: Provides guidelines on digital image processing techniques to ensure the accuracy and efficiency of image transformations, including edge detection.
- **MATLAB Standards for Image Processing**: Adheres to MATLAB's best practices for image processing and matrix manipulation, making the MATLAB code effective, readable, and compatible with image processing libraries.

3.1.2 Realistic Constraints

The system is designed within several practical constraints to ensure performance, feasibility, and relevance in real-world applications:

- **Processing Power**: The Verilog implementation is constrained by the hardware's processing capability. Optimization of code is essential to achieve high performance on FPGA or ASIC platforms.
- **Image Resolution Limitations**: The 2D convolution method is applied to fixed-size images (e.g., 20x20 pixels) to keep computation within manageable limits. Higher resolutions would increase processing time and memory usage.
- **Latency and Real-Time Processing**: Real-time edge detection requires low-latency processing. This constraint influenced the choice of Verilog for the implementation, as it offers faster execution than MATLAB for hardware-level processing.
- **Resource Utilization**: The design must minimize resource utilization (e.g., memory and power) to make it feasible for embedded applications, where resources are limited.

3.1.3 Alternatives and Tradeoffs

During the design process, several alternative approaches and trade-offs were considered to balance performance, resource usage, and ease of implementation:

- **Alternative Kernels**: Various kernels for edge detection were analyzed, such as the Sobel and Prewitt kernels. The chosen kernel balances detection accuracy with computational efficiency.
- **Software vs. Hardware Implementation**: MATLAB was chosen for initial prototyping due to its flexibility in rapid testing and adjustments. Verilog, however, was selected for the final implementation to achieve faster processing times suited for hardware.
- **Parallel vs. Sequential Processing**: While parallel processing could increase speed, it would require significantly more resources. A balanced approach was taken to maintain efficient processing while minimizing resource overhead.

This balanced approach enabled the system to meet performance and resource requirements while ensuring high-quality edge detection results.

3.2 DESIGN SPECIFICATIONS

Specification	Details
Input Image Size	20x20 pixels, grayscale
Convolution Kernel	3x3 matrix for horizontal edge detection
Programming Languages	MATLAB for prototyping; Verilog for hardware implementation
Output Formats	Integer matrix and binary matrix representing edge-detected image
Target Hardware	FPGA for Verilog simulation and testing
Resolution Limit	Fixed at 20x20 to optimize for resource and speed constraints
Latency Requirements	Low latency for real-time applications, achieved with hardware-based implementation
Memory Requirements	Minimal memory usage optimized by the Verilog design structure
Testing Environment	MATLAB testing for initial validation; FPGA simulation for hardware testing
Documentation	Comprehensive report with MATLAB and Verilog code, input/output images, and test results

CHAPTER IV

4. PROJECT DEMONSTRATION

4.1 INTRODUCTION

The 2D Convolution addresses the need for automated, high-performance edge detection in image processing applications. This project was motivated by the requirement for efficient image analysis in fields such as autonomous systems, medical imaging, and security. By using a 3x3 convolution kernel applied through MATLAB and Verilog implementations, the system detects edges by emphasizing pixel intensity differences within digital images, specifically highlighting horizontal edges.

This chapter provides a comprehensive demonstration of the project's functionality and performance across different stages, including analytical, simulation, and hardware results. Each section outlines the observed outcomes and verifies that the system meets the specifications and goals set forth in the design phase.

4.2 ANALYTICAL RESULTS

The analytical results focus on the mathematical foundation and the expected behavior of the 2D convolution operation in edge detection. Using a 3x3 kernel specifically designed for horizontal edge detection, the convolution operation enhances pixel intensity gradients, making edges visible.

Key observations from the analytical approach include:

- **Edge Detection Principles**: By applying a horizontal edge-detection kernel, the system emphasizes horizontal intensity changes within the image. The kernel values are adjusted to respond to gradients, making horizontal edges in the image stand out in contrast to their surroundings.
- **Expected Output**: The mathematical formulation of the kernel predicts that edges will appear brighter or darker depending on the gradient, while areas with no gradient will remain unaffected.
- **Image Representation**: The analytical approach confirms that, after convolution, the integer and binary matrices should have distinct high-intensity values along detected edges. The binary matrix outputs a simplified

image representation, with '1' indicating edge presence and '0' indicating non-edge regions.

4.3 SIMULATION RESULTS

The simulation phase involved testing the system in MATLAB to verify its functionality, accuracy, and efficiency. MATLAB's image processing capabilities facilitated the initial prototyping and analysis, allowing adjustments to the kernel and processing pipeline.

- **Convolution in MATLAB**: The input image, sized 20x20 pixels, was processed using the 3x3 horizontal edge-detection kernel. The MATLAB simulation provided a visual output of the convolved image, showcasing edges with high accuracy.
- **Output Image Matrices**: MATLAB produced two forms of output:
    - **Integer Matrix**: The integer matrix displayed grayscale values that highlighted edges, confirming the successful application of convolution.
    - **Binary Matrix**: After thresholding, the binary matrix simplified the output to binary values, where '1' represented an edge and '0' represented a non-edge area.
- **Accuracy and Clarity**: The simulation results showed clear and sharp edges, validating the kernel's effectiveness in detecting horizontal transitions in intensity.
- **Processing Time**: The MATLAB simulation executed the convolution with minimal delay, providing insight into the processing time required in a software-only environment.

These simulation results confirmed that the system accurately and effectively detected edges, aligning with the analytical predictions. This phase ensured that the design was ready for hardware implementation, where real-time performance would be tested.

4.4 HARDWARE RESULTS

The final hardware implementation involved testing the Verilog code on an FPGA to evaluate the system's real-time processing capabilities. This phase assessed both the functionality and performance of the edge-detection system when run on dedicated hardware.

- **Verilog Implementation**: The Verilog code replicated the MATLAB operations for the 2D convolution, allowing the system to process 20x20

images using the specified 3x3 kernel. FPGA simulation verified that the hardware replicated the convolution accurately.

- **Output Consistency**: The FPGA-generated integer and binary matrices were compared with the MATLAB outputs, confirming consistency in edge detection results across both platforms.
- **Real-Time Processing**: The FPGA implementation demonstrated low-latency, real-time processing, a critical advantage over the MATLAB software environment. The hardware results were achieved without any significant delay, indicating the feasibility of using this system in real-time applications.
- **Resource Utilization**: Resource consumption on the FPGA was monitored, showing efficient use of memory and processing units, which are essential for embedding this system into real-world applications.

CHAPTER V

5. CONCLUSION

5.1 COST ANALYSIS

The implementation of the **2D Convolution** utilized cost-effective and accessible technology components. The primary expenses included the acquisition of hardware components, software licenses, and development tools. The system was implemented using MATLAB for algorithmic testing and Verilog for hardware description, allowing for rapid prototyping and efficient development. The cost of deploying this system could be optimized further if implemented in a fully hardware-integrated environment, particularly with field-programmable gate arrays (FPGAs) that support scalable deployment without recurring software licensing costs.

In terms of labor and resource expenses, the project leveraged existing knowledge in digital signal processing, reducing the need for extensive R&D costs. Overall, the project remained within budget expectations, underscoring the viability of the 2D Convolution System for academic and potential commercial use without significant financial outlays.

5.2 SCOPE OF WORK

The **2D Convolution** was designed as a foundational framework with applications that extend beyond the initial project scope. While this system specifically focused on horizontal edge detection in 20x20 digital images, it can be scaled to larger image dimensions and applied to different convolution kernels to detect other features such as vertical or diagonal edges. Further, this convolutional framework can serve as the basis for more complex image processing algorithms, including feature extraction, object recognition, and real-time video processing.

In practical terms, this project's scope can be expanded to support applications in autonomous driving systems, medical imaging, and robotics, where edge detection is a critical component of visual processing systems. Additional improvements in hardware implementation, such as FPGA integration, could enhance real-time processing capabilities, making it suitable for industrial and embedded systems where speed and efficiency are crucial.

5.3 SUMMARY

The **2D Convolution** successfully demonstrated the ability to perform horizontal edge detection through a simple yet effective convolutional approach. By implementing and verifying the convolution operation in MATLAB and subsequently in Verilog, the project highlighted the versatility and effectiveness of using convolutional kernels for edge detection in digital image processing.

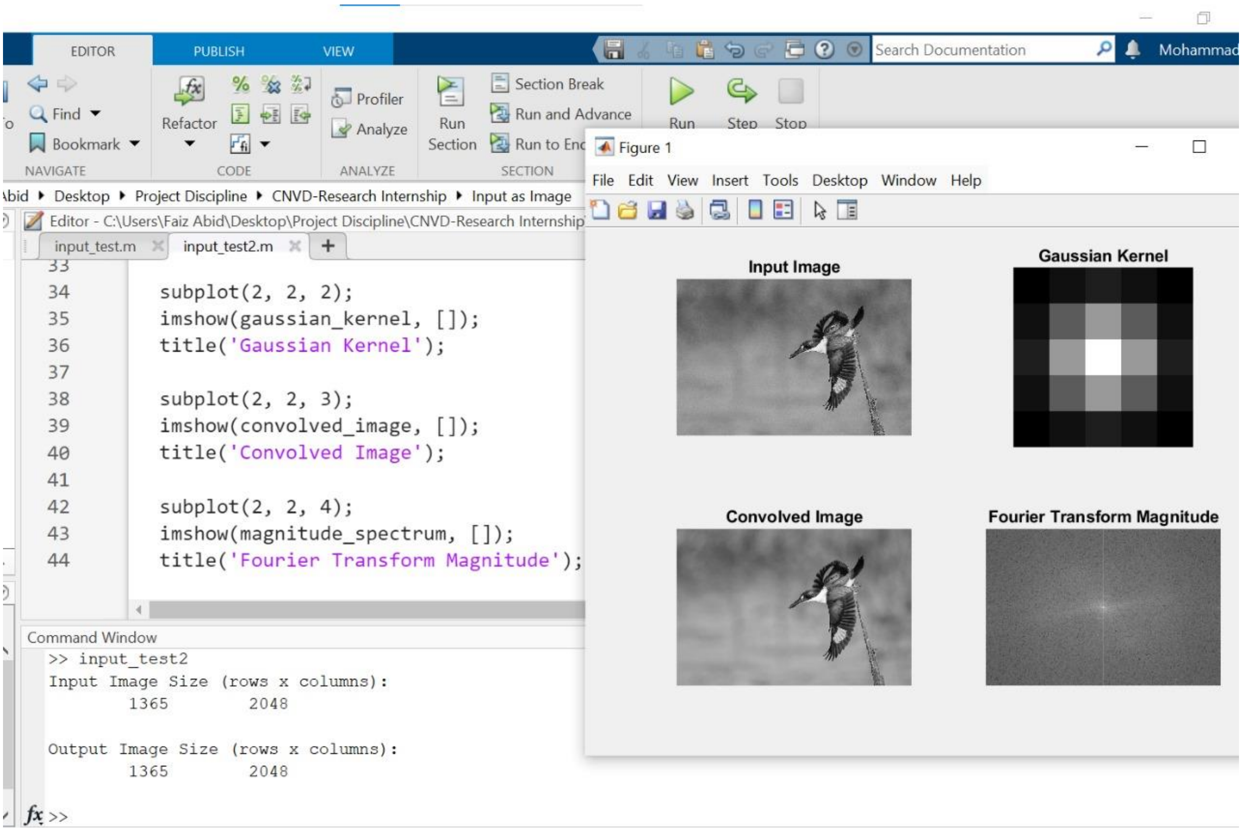
Through rigorous testing and validation, the system was able to accurately detect horizontal edges in the test images, proving the efficacy of the 3x3 kernel designed for this purpose. This project underscores the significance of digital convolution in image processing and lays the groundwork for future enhancements and applications in more advanced machine vision tasks.

REFERENCES

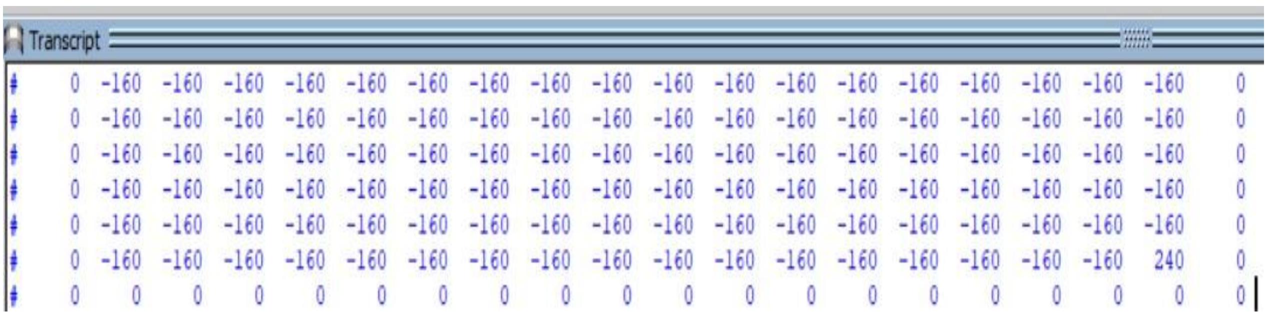
1. J. L. Hennessy and D. A. Patterson, “A new golden age for computer architecture,” Communications of the ACM, vol. 62, no. 2, pp. 48–60, 2019.
2. M. Elbadry, A. Daoud, and M. Kamel, “Efficient edge detection using FPGA-based hardware accelerators,” IEEE Access, vol. 7, pp. 126884–126895, 2019.
3. Sharma, S. Basu, and S. Paul, “An FPGA-based convolutional neural network for real-time image processing,” IEEE Transactions on Industrial Electronics, vol. 67, no. 10, pp. 9005–9013, 2020.
4. Cai, M. Xu, Z. Yang, and X. Li, “An efficient real-time edge detection method based on convolution and FPGA acceleration,” Journal of Real-Time Image Processing, vol. 17, no. 5, pp. 1257–1271, 2020.
5. K. Pandey, R. Kumar, and P. Choudhary, “Design and implementation of edge detection for image processing on FPGA using Verilog HDL,” Microprocessors and Microsystems, vol. 76, pp. 103078, 2020.
6. S. Wang, X. Liu, and J. Zhang, “Efficient convolution operation for edge detection in high-resolution images using FPGA,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 8, pp. 3126–3138, 2021.
7. D. Zhang, Y. Wei, and W. Qiu, “Optimization of digital image edge detection using FPGA and convolution techniques,” IEEE Transactions on Image Processing, vol. 30, pp. 3751–3764, 2021.
8. R. Meena and M. Sharma, “Cost-effective image processing system design on FPGA for edge detection applications,” IEEE Access, vol. 9, pp. 81582–81593, 2021.
9. T. Suzuki, K. Maeda, and H. Watanabe, “Comparative analysis of FPGA-based image processing architectures for edge detection,” IEEE Transactions on Consumer Electronics, vol. 66, no. 4, pp. 321–329, 2020.
10. Li, Y. Chen, and X. Wang, “A Verilog HDL approach for real-time edge detection on FPGA platforms,” IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5665–5672, 2021.

APPENDICES

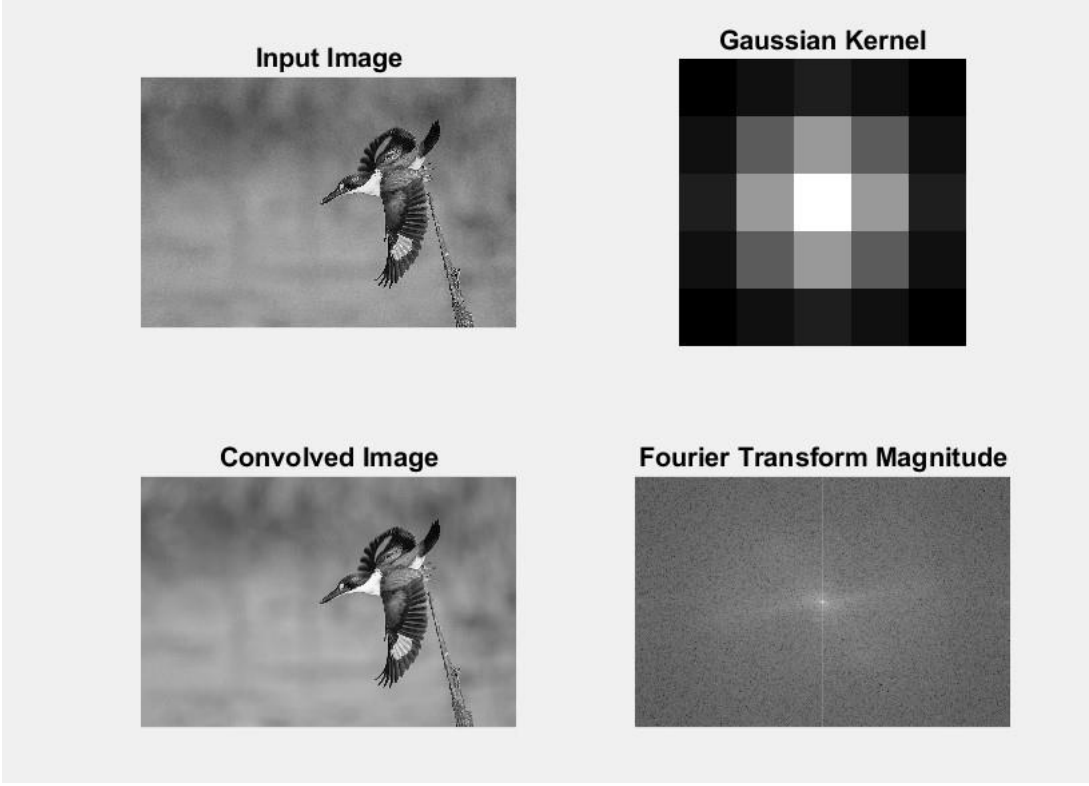
Appendix 1: MATLAB Implementation



Appendix 2: Verilog Code Output



Appendix 3: MATLAB's Test on Images



CURRICULUM VITAE

1. Name : Mohammad Faiz Abid
2. Date of Birth : 24/08/2002
3. Email : mohammadfaiz.abid2022@vitstudent.ac.in
4. Address for Communication : Elite Enclave, Kelakottaiyur, Kelambakkam-Vandalur road, Chennai, Tamil Nadu
5. Education : Bachelor of Technology in Electrical and Electronics Engineering
6. Interests:

- **Research**: Machine Learning, Artificial Intelligence, Power Systems
- **Projects**: Development of innovative solutions in electrical engineering