THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Image Detection with RISC-V Processor on FPGA

Project Proposal

Joshua Wallace

45809978

2024

The University of Queensland

School of Information Technology and Electrical Engineering

Table of Abbreviations

Abbreviations	
FPGA	Field Programmable Gate Array
ASIC	Application Specific Integrated Circuit
CPU	Central Processing Unit
GPU	Graphical Processing Unit
SoC	System on Chip
ISA	Instruction Set Architecture
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer
IO	Input Output
RTOS	Real Time Operating System
CNN	Convolutional Neural Network
YOLO	You Only Look Once
VGA	Video Graphics Array

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, the demand for efficient and versatile image processing systems has surged across various domains, including surveillance, medical imaging, autonomous vehicles, and more. Field Programmable Gate Arrays (FPGAs) provide a reconfigurable, low-power embedded platform with parallel processing capabilities akin to graphic processing units (GPUs) commonly used to accelerate demanding computing tasks. Image processing is a good candidate for such application of parallel processing, due to the increasing algorithm complexity and large volume of data involved. [3] Hence, FPGAs could provide a suitable platform for real-time image processing tasks, offering a balance of performance, power efficiency, and flexibility.

One significant aspect of image processing systems is the choice of the underlying processor architecture. Traditional approaches often rely on general-purpose CPUs or GPUs to execute image processing tasks. However, these architectures may not always offer the best balance of performance, power efficiency, and flexibility for image processing applications. In recent years, the RISC-V instruction set architecture (ISA) has gained traction as an open, customizable, and energy-efficient alternative to proprietary processor designs. These soft processors can be implemented on FPGAs, providing a controllable and scalable platform for real-time image processing tasks whilst enabling the processing tasks to be performed in parallel.

This project will focus on the exploration and implementation of image detection algorithms using the RISC-V processor architecture deployed on an FPGA platform. By leveraging the configurability of FPGAs and the energy efficiency of the RISC-V ISA, this research aims to develop a scalable and adaptable solution for image processing using FPGA-accelerated hardware. The data processing capacity is large that the processing speed must be strict to meet the demands of real-time time image transmission [4]. Hence, an FPGA system-on-chip (SoC) offers a means for a low power consumption, low latency but high throughput platform - all of which are essential for the increasing computational demands of image processing tasks [5].

The work will produce a hardware implementation of an image processing system that can be controlled by a RISC-V processor, and demonstrate the benefits of FPGA-based hardware acceleration for image detection tasks.

Chapter 2

Background

2.1 Background

A review of existing literature relevant to the topic has been conducted to provide a basis for the work. The related works to the research were selected zx: Field-Programmable Gate Arrays (FPGAs), RISC-V processors, image detection methods, and convolutional neural networks (CNNs). These are broadly discussed below, with a focus on the application of these technologies to image detection and hardware implementations.

2.1.1 Image Detection and Analysis

Image analysis is the process of extracting features from an image, to simplify it into a form that is more easily interpreted [6]. This often includes edge detection, object recognition, and image classification as parts of the processing pipeline. Edge detection forms the basis of many image analysis techniques, as it is used to localize the variations in the image and to identify the physical phenomena which produce them. These are generally classed as gradient-based or Laplacian based, though gradient-based algorithms are more widely used [2].

The Robert, Prewitt and Sobel techniques are the most prominent of the gradient-based algorithms, applying kernels to the image to detect edges [7]. [4, 8–11] all provide a hardware implementation of the these filters, utilising the parallel processing capabilities of FPGAs to accelerate the gradient operations. These designs realized improved efficiency in performance defined as increased throughput, and resource utilisation. Hence, FPGAs can provide a platform for processing real time algorithms on application-specific hardware with substantially higher performance than programmable digital signal processors (DSPs) [1]. This strongly indicates that FPGA-based implementations of image detection algorithms can offer significant improvements in running time for a range of digital signal processing methods, but were benchmarked only against microcontroller-based CPU platforms and not parallelised platforms.

The general pipeline for image detection and analysis is shown in Figure 2.1. This covers the available techniques which can be abstracted into hardware, to provide a high-speed, real-time image detection system.

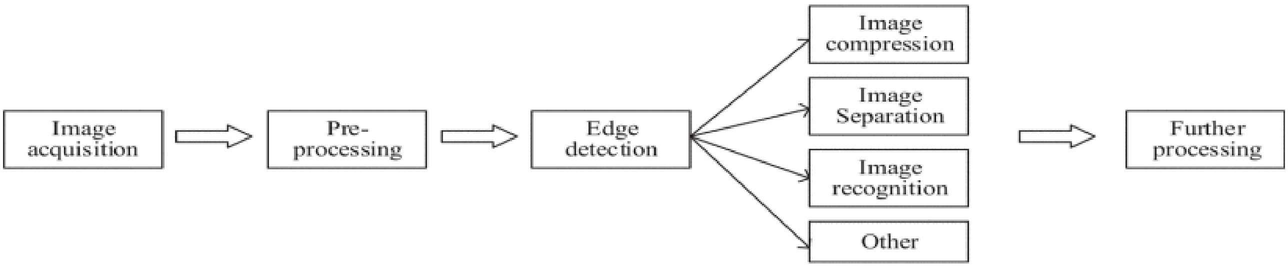

Figure 2.1: Pipeline for image analysis techniques [1]

[11, 12] follow a similar pipeline to Figure 2.1, with the image being captured by a camera module, and then processed by an FPGA though using differing image techniques. These found that FPGAs can accelerate the required multiplication and addition operators separately - required for classical, and deep learning, based techniques [13].

2.1.2 Convolutional Neural Networks and Deep Learning

Convolutional neural networks (CNNs) are a type of artificial neural network that is primarily used for image detection, as it uses the convolution kernel to detect features in the image. Due to the lack of memory in low-end FPGA models, the CNN is optimal for an image processing task on SoC, as it has a low number of weights and biases relative to other neural networks [14] The convolution operation requires a large number of multiply-accumulate (MAC) operations, and is the primary bottleneck in the performance of CNNs. The operation must window over the entire image, and the number of MAC operations is proportional to the size of the image and the number of filters in the layer. Hence, the CNN is computationally intensive, demanding a significant quantity of operations for high accuracy [15]. However, the proceeding [14] paper found that the use of an embedded co-processor makes the implementation of CNN less challenging on the FPGA platform, as the network can be designed in a systems language such as C. This can then be compiled and translated into machine code, as there exists a toolchain for RISC-V.

Similar works for image-analysis using different neural network architectures have also been implemented using FPGA hardware. [16, 17] demonstrates a hardware implementation of the You Only Look Once (YOLO) algorithm using a Xilinx Virtex-7 FPGA. Like the CNN hardware implementations, the architecture is focused on accelerating the convolution operation - and is a common denominator in the performance neural networks in hardware. This acceleration is not specific to the convolution operator, and can be applied to any intensive operation in the network as the FPGA has little overhead on each operator when compared to traditional platforms [18]. This is corroborated by [5], which found that the implementation of hybrid neural networks on a Xilinix Virtex-7 690T FPGA achieved 4.13 times higher throughput than state of the accelerators in 2019.

2.1.3 RISC-V

Instruction set architectures (ISAs) define the operations a processor can execute, however, the majority of ISAs are proprietary and require licensing to use. Reduced Instruction Set Computing (RISC) is a form of ISA which offers a simpler, more efficient design than the traditional Complex Instruction Set Computing (CISC) ISAs - such as the prevalent x86 instruction set [19]. Arm and RISC-V are the two most prominent RISC ISAs, with Arm being the most widely used in embedded systems but with RISC-V offering royalty-free licensing and an open-source nature. The lack of licensing fees and the open-source nature of RISC-V has led to its increasing popularity in the embedded systems domain [20]. It is estimated that the number of chips utilising RISC-V technology ”will grow 73.6 percent per year to 2027, when there will be some 25 billion AI chips produced, accounting for US $291 billion in revenue” [14].

There exists a number of FPGA implementations to create softcore processors using the RISC-V ISA [21]. This 2023 paper [20] demonstrates a RISC-V implementation of the NEORV32 core, using a Wishbone bus interface. The authors selected the NEORV32 core due to it being vendor-agnostic and platform independent, with the project being highly documented. The softcore nature allows for the implementation details to be customised to the specific application, such that the core can be adapted to the specific use case [22]. The NEORV32 processors offers a system-on-chip (SoC) Harvard architecture, with a 32-bit RISC-V processor, and a range of peripherals. It supports UART, SPI, standard GPIO and the Wishbone b4 external bus interface for SoC connectivity [23].

As it is an open-source ISA, RISC-V has the added benefit of being extensible and modular [24], allowing for instructions to be added to the processor. [25] utilises this to create custom instructions within the RISC-V ISA to accelerate the expensive convolution operation for a CNN. Other applications have also extended the architecture to include custom instructions for frequently used and computationally-expensive operations.

2.1.4 Field-Programmable Gate Arrays

Field Programmable gate arrays (FPGAs) provide flexible compute resources that can be reconfigured to suit a wide range of applications. Similar to graphics processing units (GPUs), FPGAs offer parallel processing capabilities, but with the added benefit of reconfigurability [26]. The ability to develop custom logic designs is not unique to FPGAs, as application-specific integrated circuits (ASICs) also offer this capability. ASICs offer better optimization in many aspects than FPGAs, but cannot be reconfigured and require a large upfront cost for design and fabrication [27]. This makes FPGAs a more suitable platform for low-volume production and prototyping, as they offer a faster development cycle and lower cost than ASICs.

These reconfigurable platforms are optimal for edge applications on images, due to the parallelised pipeline structure, which enables high-speed processing of large amounts of image data, and it’s high processing speed ensures real-time image data processing [4]. FPGAs in particular have been studied extensively in the field of image detection, as replacements to existing hardware infrastructures due to the increasing complexity of algorithms [28]. They offer low latency and low power consumption due to their computing characteristics. However, the ability to configure oftens mean that resource constraints and utilisation must be considered when designing the system, such as with many embedded devices.

Figure 2.2 shows an implemention of a hardware-accelerated edge-detection algorithm for video images, using the Intel Cyclone IV FPGA platform direct from a camera module. This requires a camera module, FPGA and uses a VGA monitor to display the output - but does not use a softcore processor.

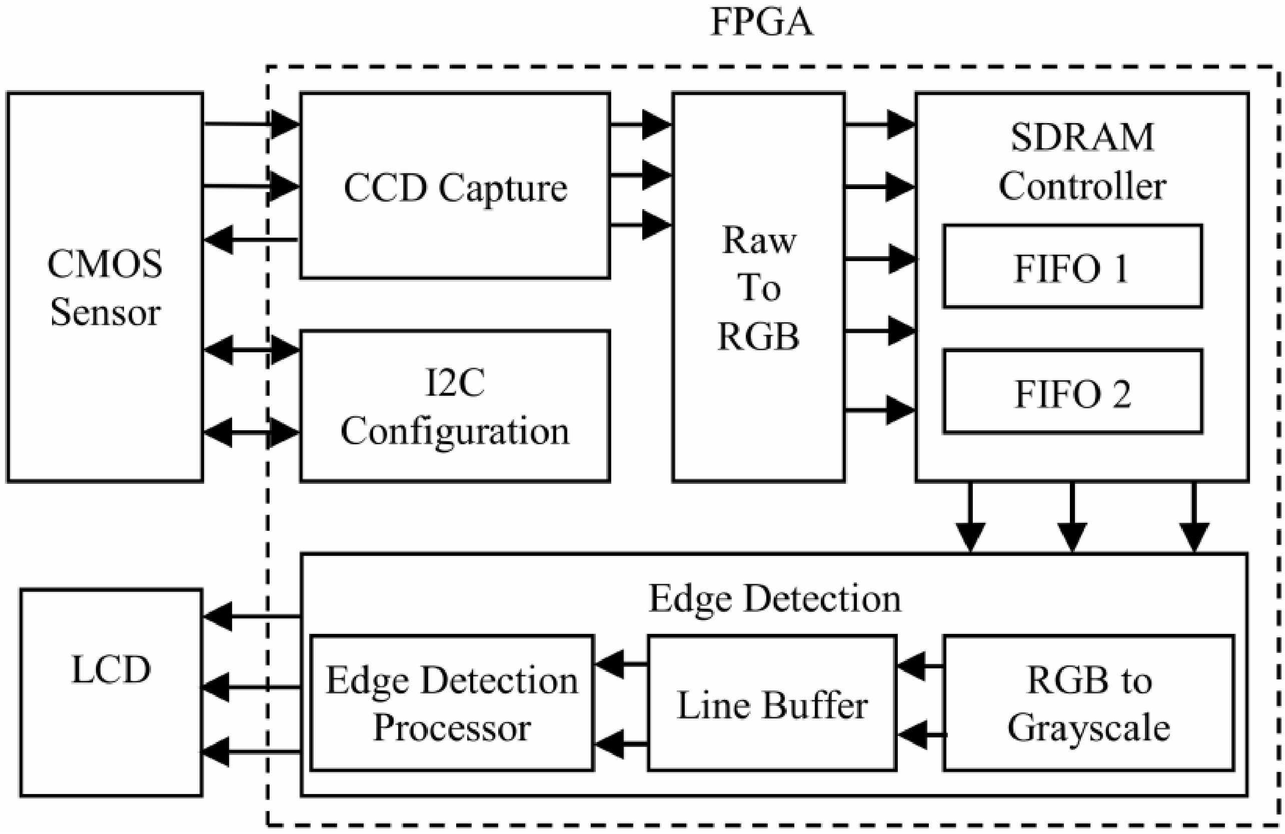

Figure 2.2: Video detection FPGA pipeline [2]

This pipeline has the core advantage over traditional CPU architectures which are serially structured, as it can perform the operations in hardware and in parallel. Hence, an image processing system employing an FPGA as the primary control chip is better suited to the low-latency and high speed demands of real-time image processing [1]. Applying a softcore processor to such a pipeline would allow for improved control on a real-time system. Furthermore, custom instructions can be added to optimise repeat operations at the systems level whilst maintaining the advantage of performing the processing using the hardware.

Chapter 3

Topic

3.1 Topic

Image detection, a critical task in computer vision, involves identifying and locating objects within images. Traditional software-based approaches for image detection often face challenges related to processing speed and resource utilization, particularly when dealing with large datasets or real-time applications. In response to these challenges, I propose leveraging Field-Programmable Gate Arrays (FPGAs) to accelerate image detection algorithms in hardware. By offloading computationally intensive tasks to FPGA-based hardware accelerators, we aim to significantly enhance the performance and efficiency of image detection systems.

The integration of FPGA-based hardware acceleration offers several compelling advantages. Firstly, FPGAs provide inherent parallelism and customizable hardware architectures, enabling efficient execution of image processing algorithms. This parallelism can lead to substantial improvements in processing speed, making real-time image detection feasible for applications such as surveillance, autonomous vehicles, and industrial automation. Additionally, FPGA-based solutions offer flexibility and scalability, allowing for the optimization of hardware resources to suit specific image detection requirements. Their adaptability is particularly advantageous in scenarios where the detection algorithms need to be customized or updated frequently.

Furthermore, hardware-accelerated image detection using FPGAs can lead to significant reductions in power consumption compared to traditional CPU-based approaches, making it suitable for resource-constrained environments and embedded systems. By harnessing the computational efficiency of FPGA-based acceleration, we can achieve higher throughput and lower latency, thereby improving the responsiveness and effectiveness of image detection systems. Overall, this research aims to demonstrate the practical significance and utility of FPGA-based hardware acceleration in advancing the capabilities of image detection technology, paving the way for more efficient and scalable solutions in various domains.

3.2 Aims

This projects aim to:

- Demonstrate the feasibility and benefits of FPGA-based hardware acceleration for image detection algorithms.
- Demonstrate the control of the image detection algorithm using a RISC-V softcore processor.
- Evaluate the performance, efficiency, and scalability of a FPGA-based image detection system.
- Investigate the potential applications and use cases of FPGAs for image detection tasks in real-world scenarios.

It will not include the development of a new image detection algorithm or method, but rather the implementation of existing algorithm(s) for hardware-acceleration, and the control through a RISC-V softcore processor.

3.3 Overview

The work can be divided into subcomponents to provide a timeline and structure to the project, as dependencies on previous work are required to progress.

3.3.1 Hardware

The work will require the Xilinx Artix 7 XC7A100T FPGA, and a corresponding board to access the FPGA resources and I/O ports. A Digilent Basys 3 board is available initially for development, however, a Digilent Nexys A7-100T board may be required depending on resource limits. Both these boards use the Artix 7 XC7A100T FPGA, and have VGA output ports to allow for processed images to be displayed to an external monitor. A camera module will be connected to the FPGA, and the NEORV32 processor will be used to control the image detection algorithm. This module has currently not been selected in it's entirety, but initial trials will be conducted with an available VGA OV7670 camera module. The module has been selected due to it's availability, compatability and previous research [12] conducted with it.

Figure 3.1 below provides an overview of the hardware components and their connections.

3.3.2 Softcore Processor

The NEORV32 RISC-V softcore processor has been selected due to it's highly documented ecosystem, and personal familiarity. This processor will be implemented as a SoC on the FPGA, and will be used to control the image detection algorithm and interface with the camera module. A Wishbone b4 interface will be used to connect the processor to the FPGA.

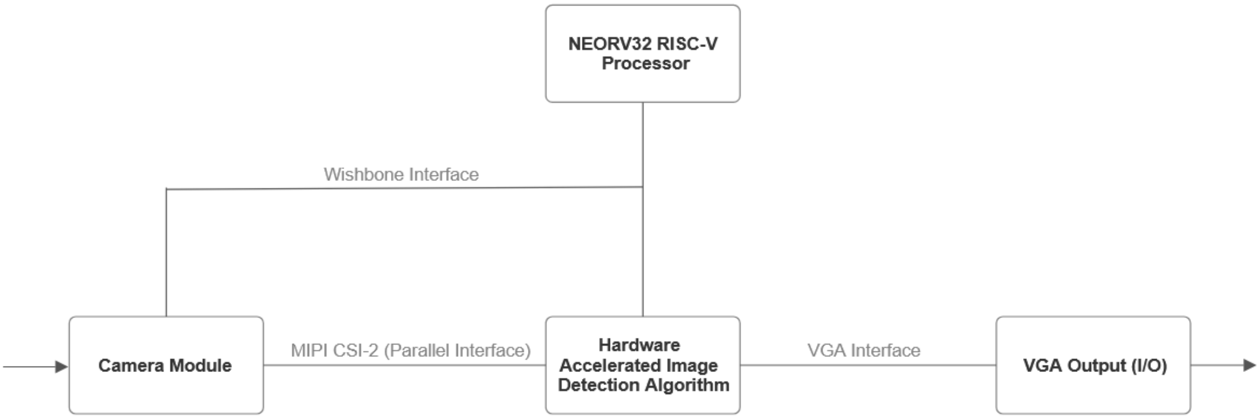

Figure 3.1: Overview of proposed FPGA-based system.

3.3.3 Image Detection Algorithm

A convolutional neural network has been selected, such as in [2], to be implemented on the FPGA for image detection. Existing research has demonstrated that is not as resource intensive as other neural networks, and has noticeable advantages over traditional image detection algorithms. Furthermore, image analysis is moving in the direction of deep learning algorithms for more complex image detection tasks, such as classification [29]. By employing a CNN as the algorithm, the project will be able to demonstrate the capabilities of FPGA-based hardware acceleration for deep learning tasks and can be extended to a variety of domains.

3.4 Performance Indicators

The performance indicators of the system to establish the success of the project are provided in Table 3.1 below.

Table 3.1: Table of performance indicators.

Indicator	Description
Algorithm accuracy	The ability of the system to correctly detect features in an image.
Scalability	A measure of the limits of the system in terms of image size, image complexity and algorithm complexity
Throughput	Measured as the number of images processed per time unit
Latency	The time delay between acquisition of image and output of results relative to the clock-speed of the Xilinx Artix 7 XC7A100T FPGA

Continued on next page

Table 3.1: Performance indicators to assess the success of the project. (continued)

Indicator	Description
Resource Utilization	The amount of FPGA resources used to implement the system specified by number of LUTs and BRAM required for implementation

3.5 Required Resources

The hardware will be designed using both a Digilent Basys 3 development board for initial prototyping, before ideally being migrated to a Digilent Nexys A7-100T development board, to access more I/O and resources. This prototyping step is done to demonstrate the implementation on severely resource constrained platforms, as per the scalability performance indicator. Both the selected boards have a VGA output, which can be used to display the processed image directly from the FPGA board. The specific camera module to be used has not yet been selected, however, a VGA OV7670 camera module which uses an I2C communication protocol is being considered for initial trials. A VGA monitor will also be required to display the output of the processed image.

Benchmarking materials are dependent on the completed image detection algorithm selected, and cannot be provided at this time. Thus, the following resources are required for the project at this stage:

- Xilinx Artix 7 XC7A100T FPGA
- Digilent Basys 3 FPGA Board
- Digilent Nexys A7 100T FPGA Board
- Camera Module
- VGA OV7670 Camera Module I2C
- VGA Monitor
- USB Power Supply
- VGA to VGA Cable / VGA to HDMI Cable

Chapter 4

Research Plan

4.1 Milestones

To track the development of the project, a set of milestones have been established with consideration of assessable dates. Table 4.1 below outlines the milestones for the work and what is required for each phase. Emboldened text indicates an assessable milestone.

Table 4.1: Timeline of milestones.

Milestone	Description	Date
Project Proposal	Complete the project proposal and literature review.	21/3
Configure hardware	Implement NEORV32 processor on FPGA, connect to external components through wishbone interface.	15/4
Seminar	Present the current project status.	6/5
Image detection algorithm	Select and implement image detection algorithm in hardware.	12/8
VGA driver	Develop driver and interface to display processed image on VGA monitor.	9/9
Benchmark	Benchmark system performance against related works.	4/10
Demo	Demonstrate the complete project.	18/10
Thesis	Document and write project results.	4/11

The timeline was developed based on the expected time to completion for each task. A more detailed overview of the non-assessable tasks required for each milestone is provided in the following sections.

4.1.1 Configure Hardware

This task has been allocated three weeks of work, estimated at 20 hours of work. It entails the implementation of the NEORV32 processor on the FPGA, and the connection of the processor to the external components through the wishbone interface.

4.1.2 Image Detection Algorithm

This task has been allocated three months of work, estimated at 100 hours of work. It requires the selection, implementation, and testing of the image detection algorithm in hardware. Furthermore, it must additionally interface with the NEORV32 processor, and be able to be controlled by the processor. As it is the core of the project, it has been allocated the most time.

4.1.3 VGA Driver

This task has been allocated four weeks of work, estimated at 30 hours of work. The task requires the development of a driver and interface to display the processed image on a VGA monitor for the selected algorithm. It is expected to be a relatively simple task, but has been allocated additional time for any unforeseen issues.

4.1.4 Benchmark

This task has been allocated four weeks of work, estimated at 30 hours of work. It requires the benchmarking of the system performance against related works, and the documentation of the results. The task is required to provide an assessment of the project's success against the performance indicators.

4.2 Risk Assessment

This project is conducted in the low-risk laboratory covered by general OHS laboratory rules, and in a home setting. There are no hazardous materials or dangerous equipment used in the project, and the risk of injury is negligible. The only risk to the project is hardware failure or proprietary software issues, which can be mitigated by using open-source software and hardware, and regular backups of the project files. Redundancies are also in place for hardware failure, as two FPGA developments boards are designated for use.

Bibliography

[1] C. Ye, “Real-time image edge detection system design and algorithms for artificial intelligence fpgas,” in *2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs)*, pp. 476–481, 2022.

[2] B. K. Upadhyaya and D. Chakraborty, “Fpga implementation of gradient based edge detection algorithms for real time image,” in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1227–1233, 2018.

[3] R. Khosla and B. Singh, “Implementation of efficient image processing algorithm on fpga,” in *2013 International Conference on Machine Intelligence and Research Advancement*, pp. 335–339, 2013.

[4] X. Wei, G.-M. Du, X. Wang, H. Cao, S. Hu, D. Zhang, and Z. Li, “Fpga implementation of hardware accelerator for real-time video image edge detection,” in *2021 IEEE 15th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pp. 16–20, 2021.

[5] S. Yin, S. Tang, X. Lin, P. Ouyang, F. Tu, L. Liu, and S. Wei, “A high throughput acceleration for hybrid neural networks with efficient resource management on fpga,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 4, pp. 678–691, 2019.

[6] “Image analysis.” `https://www.mathworks.com/discovery/image-analysis.html`.

[7] A. B. N, M. Afsar, K. K. Khaitan, Rahul, and C. Gururaj, “Optimized fpga implementation and synthesis of image segmentation techniques,” in *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, pp. 191–196, 2021.

[8] G. B. Reddy and K. Anusudha, “Implementation of image edge detection on fpga using xsg,” in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–5, 2016.

[9] S. Yaman, B. Karakaya, and Y. Erol, “Real time edge detection via ip-core based sobel filter on fpga,” in *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*, vol. 1, pp. 1–4, 2019.

[10] D. Sangeetha and P. Deepa, “An efficient hardware implementation of canny edge detection algorithm,” in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, pp. 457–462, 2016.

[11] R. Harinarayan, R. Pannerselvam, M. Mubarak Ali, and D. Kumar Tripathi, “Feature extraction of digital aerial images by fpga based implementation of edge detection algorithms,” in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, pp. 631–635, 2011.

[12] E. Gholizadehazari, T. Ayhan, and B. Ors, “An fpga implementation of a risc-v based soc system for image processing applications,” in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2021.

[13] Y. Ma, Q. Xu, and Z. Song, “Resource-efficient optimization for fpga-based convolution accelerator,” *Electronics*, vol. 12, p. 4333, 10 2023.

[14] S. K. Mousavikia, E. Gholizadehazari, M. Mousazadeh, and S. B. O. Yalcin, “Instruction set extension of a riscv based soc for driver drowsiness detection,” *IEEE Access*, vol. 10, pp. 58151–58162, 2022.

[15] M. F. Tolba, H. Saleh, M. Al-Qutayri, A. Hroub, and T. Stouraitis, “Efficient cnn hardware architecture based on linear approximation and computation reuse technique,” in *2023 International Conference on Microelectronics (ICM)*, pp. 7–10, 2023.

[16] G. Zhang, K. Zhao, B. Wu, Y. Sun, L. Sun, and F. Liang, “A risc-v based hardware accelerator designed for yolo object detection system,” in *2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE)*, pp. 9–11, 2019.

[17] Z. Wang, K. Xu, S. Wu, L. Liu, L. Liu, and D. Wang, “Sparse-yolo: Hardware/software co-design of an fpga accelerator for yolov2,” *IEEE Access*, vol. 8, pp. 116569–116585, 2020.

[18] J. He, W. Li, J. Xu, M. Zhang, and H. Yang, “Research on convolution decomposition and hardware acceleration based on fpga,” in *2023 4th International Conference on Computer Engineering and Application (ICCEA)*, pp. 285–291, 2023.

[19] ARM, “Risc (reduced instruction set computing).” ARM Glossary.

[20] K. Böhmer, B. Forlin, C. Cazzaniga, P. Rech, G. Furano, N. Alachiotis, and M. Ottavi, “Neutron radiation tests of the neorv32 risc-v soc on flash-based fpgas,” in *2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp. 1–6, 2023.

[21] L. Poli, S. Saha, X. Zhai, and K. D. Mcdonald-Maier, “Design and implementation of a risc v processor on fpga,” in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 161–166, 2021.

[22] T. Harmina, D. Hofman, and J. Benjak, "Dct implementation on a custom fpga risc-v processor," in *2023 International Symposium ELMAR*, pp. 163–167, 2023.

[23] S. Nolting, "The neorv32 risc-v processor." `https://stnolting.github.io/neorv32/`, 2024.

[24] D.-T. Nguyen-Hoang, K.-M. Ma, D.-L. Le, H.-H. Thai, T.-B.-T. Cao, and D.-H. Le, "Implementation of a 32-bit risc-v processor with cryptography accelerators on fpga and asic," in *2022 IEEE Ninth International Conference on Communications and Electronics (ICCE)*, pp. 219–224, 2022.

[25] N. Wu, T. Jiang, L. Zhang, F. Zhou, and F. Ge, "A reconfigurable convolutional neural network-accelerated coprocessor based on risc-v instruction set," *Electronics*, vol. 9, no. 6, p. 1005, 2020.

[26] N. R, "Fpgas vs gpus - what's the difference?," 2023.

[27] A. S. Hussein and H. Mostafa, "Asic-fpga gap for a risc-v core implementation for dnn applications," in *2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 385–388, 2021.

[28] T. Xiao and M. Tao, "Research on fpga based convolutional neural network acceleration method," in *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 289–292, 2021.

[29] A. Rastogi, R. Singh, R. Sharma, and S. D. Kalony, "The survey of digital image analysis with artificial intelligence- dcnn technique," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, pp. 209–211, 2020.