STM JOURNALS

Journal of
Semiconductor Devices and Circuits

ISSN: 2455-3379
Volume 11, Issue 3, 2024
September–December
DOI (Journal): 10.37591/JoSDC

https://journals.stmjournals.com/josdc

Review JoSDC

Optimizing Image Processing with Verilog on FPGA: Techniques and Performance Enhancements

Parul H. Panchal[1], Smit Hingarajiya[2,*], Malay Tarapara[3]

Abstract

The integration of image-processing algorithms into hardware platforms, particularly FPGAs, presents a compelling opportunity for achieving high performance, low latency, and power efficiency in real-time applications. This study focuses on designing and implementing optimal image-processing methods using Verilog hardware description language (HDL) for field-programmable gate array (FPGA)-based systems. The study explores the development of core algorithms, including edge detection, image enhancement, and adaptive filtering, to maximize resource utilization and processing speed on hardware platforms. Key contributions include the parallel processing of image data streams and the implementation of custom hardware accelerators to improve throughput and efficiency. The proposed system architecture was tested on a Xilinx FPGA platform, demonstrating significant improvements in processing speed and power efficiency compared to traditional Central Processing Unit (CPU) and Graphics processing unit (GPU)-based implementations. Performance evaluations reveal that the Verilog-based FPGA designs outperform software solutions in real-time processing tasks, making them ideal for embedded systems with stringent power and performance constraints. This work underscores the potential of hardware-driven approaches to revolutionize image-processing tasks by leveraging the reconfigurability and parallelism of FPGAs.

Keywords: Integrated chips, image processing, HDL, resource optimization, parallel processing

INTRODUCTION

Image processing plays a pivotal role in various applications, including medical imaging, security surveillance, autonomous vehicles, and industrial automation. Traditional software-based implementations of image-processing algorithms often struggle to meet real-time performance requirements due to their high computational complexity and resource demands. Field-programmable gate arrays (FPGAs) with their inherent parallelism and reconfigurability, offer a promising hardware platform for accelerating image-processing tasks. Leveraging hardware description languages (HDLs) such as Verilog allows designers to implement optimized, custom-tailored algorithms directly on the hardware, enabling substantial improvements in processing speed, power efficiency, and system latency [1–15].

*Author for Correspondence
Smit Hingarajiya
E-mail: 21el042@bvmengineering.ac.in

[1]Professor, Department of Electronics and Electrical Engineering, Birla Vishvakarma Mahavidyalaya, Vallabh Vidyanagar, Anand, Gujarat, India
[2,3]Students, Department of Electronics and Electrical Engineering, Birla Vishvakarma Mahavidyalaya, Vallabh Vidyanagar, Anand, Gujarat, India

Received Date: November 13, 2024
Accepted Date: November 18, 2024
Published Date: December 02, 2024

Citation: Parul H. Panchal, Smit Hingarajiya, Malay Tarapara. Optimizing Image Processing with Verilog on FPGA: Techniques and Performance Enhancements. Journal of Semiconductor Devices and Circuits. 2024; 11(3): 47–55p.

The purpose of this study is to investigate how different image-processing methods can be designed and implemented in FPGA systems using Verilog HDL. This study focuses on key algorithms, such as edge detection, image enhancement, and adaptive filtering, which are essential for many real-time applications. By

harnessing the parallel processing capabilities of FPGAs, the proposed approach seeks to address the limitations of conventional software methods, thereby offering a significant performance boost. This work not only demonstrates the advantages of hardware acceleration in image processing but also provides insights into the trade-offs between hardware resources and processing efficiency. The results of this study underscore the potential of FPGA-based designs as a robust solution for modern image-processing challenges, particularly in embedded systems with stringent performance and power constraints [16–25].

LITERATURE SURVEY

The field of image processing has seen significant advancements owing to the increasing demand for real-time applications in various domains, such as medical imaging, security surveillance, autonomous vehicles, and industrial automation. Traditional software-based implementations, typically run on CPUs and GPUs, often fall short in meeting stringent real-time processing requirements owing to their high computational demands. With their capacity for parallel processing and reconfigurability, FPGAs have become a potential hardware platform for speeding up image-processing workloads. Verilog, HDL, plays a key role in designing FPGA-based solutions, allowing precise control over hardware resources. With an emphasis on important approaches, performance measures, and research needs, this literature review examines the state of the art in the implementation of image-processing algorithms on FPGAs using Verilog. FPGA implementation of image-processing algorithms.

The potential of FPGAs to improve the performance of image-processing algorithms has been investigated in several studies. Researchers have focused on implementing core image-processing operations, such as edge detection, filtering, and enhancement, directly in hardware. These operations are critical for pre-processing in many vision-based systems.

- *Edge Detection:* Edge detection algorithms, such as Sobel, Prewitt, and Canny, are fundamental for highlighting object boundaries within images. Kumar et al. (2020) implemented the Sobel edge detection algorithm on an FPGA using Verilog and demonstrated significant improvements in processing speed compared with software implementations on CPUs. Their results indicated that the FPGA-based implementation was approximately 10 times faster and consumed less power, making it suitable for embedded applications [26].
- *Image Enhancement:* Image enhancement techniques, including contrast adjustment, brightness control, and noise reduction, are essential for improving image quality. In a notable study, Patel et al. (2019) implemented various enhancement techniques using Verilog on a Xilinx Spartan 6 FPGA, achieving real-time performance with minimal resource utilization [27]. This approach enabled high-speed processing without the need for external memory, highlighting the efficiency of direct hardware implementation.
- *Filtering:* Images are smoothed, and noise is eliminated using filtering techniques, such as Gaussian and median filtering. A research effort by Sharma et al. (2021) demonstrated FPGA implementation of Gaussian filtering using parallel processing techniques in Verilog. The study reported that the FPGA system achieved a processing speed of 200 fps, outperforming GPU-based implementations with lower power consumption [28].

Optimization Techniques for FPGA-based Image Processing

The optimization of resource utilization and processing speed is a major focus of FPGA-based image processing. Researchers have developed several strategies to enhance the performance of Verilog-based implementation.

Parallel Processing

Leveraging the parallel nature of FPGAs is a key strategy for accelerating image-processing tasks. Wong and Lee (2020) introduced a parallel processing approach for implementing k-means clustering in FPGAs using Verilog. Compared to a single-core solution, their design achieved an 8x speedup by

using multiple processing cores to handle pixel data concurrently. Parallelism is essential for real-time applications where processing speed is essential [29].

Pipeline Architecture

Pipeline architecture is another optimization technique that enhances data throughput. Gupta et al. (2022) implemented a pipelined image enhancement system on an FPGA using Verilog. Compared to non-pipelined architecture, the system throughput increased by 30% by segmenting the algorithm into smaller steps and running them concurrently. This architecture is particularly beneficial in streaming applications, where continuous data flow is required [30].

Resource Optimization

Efficient use of FPGA resources, such as Look-Up Tables (LUTs) and Block RAMs, is essential for achieving high-performance designs.

Maheshwari et al. (2021) focused on optimizing the hardware implementation of the Canny edge detection algorithm by reducing the use of multipliers and adders through algorithmic modifications. The optimized Verilog design reduces resource usage by 40% while maintaining the accuracy and speed of the original implementation [31].

Performance Comparison: FPGA vs. CPU/GPU Implementations

The ability of FPGAs to perform better than conventional CPU and GPU implementations in terms of speed and power efficiency is a primary driver of their use. Several studies have conducted comparative analyses to quantify these benefits.

Speed and Latency

An FPGA-based image-processing system designed by Khan et al. (2024) demonstrated a drastic reduction in processing latency compared to CPU-based systems. Their Verilog implementation of a morphological filter achieved a processing speed of 300 frames per second compared to 50 frames per second on a high-end CPU. This speed improvement is largely attributed to the custom-tailored hardware architecture that minimizes the data movement delays [32].

Power Efficiency

In embedded systems, power consumption is crucial, particularly for portable and battery-powered devices. Zhang et al. (2023) compared the power efficiency of FPGA, CPU, and GPU implementations of a real-time object-detection algorithm. The Verilog-based FPGA design was found to be five times more power efficient than the CPU implementation and three times more efficient than the GPU implementation, highlighting the energy-saving benefits of hardware acceleration [33].

Identified Gaps and Future Directions

Despite these encouraging outcomes, there are still a number of obstacles to overcome in the field of FPGA-based image processing. The difficulty of Verilog coding and the high learning curve involved in the FPGA design are two of the main obstacles. In addition, scalability issues arise when dealing with high-resolution images or complex algorithms, where resource limitations on FPGAs can become a bottleneck.

OPTIMIZING IMAGE PROCESSING WITH VERILOG ON FPGA

In recent years, the demand for efficient image processing has surged because of the proliferation of applications in fields such as computer vision, medical imaging, video surveillance, and autonomous vehicles. These applications frequently call for processing high-resolution photos in real time, which calls for advancements in the hardware architecture. One of the most effective ways to meet these demands is by leveraging FPGAs combined with high-level HDLs such as Verilog. This article delves into the optimization of image-processing tasks on FPGAs using Verilog [34–44].

FPGAs are flexible integrated circuits that can be configured to perform specific tasks. Their substantial parallel processing capabilities are enabled by their numerous customizable logic blocks (CLBs), input/output blocks, and interconnects. This parallelism makes FPGAs exceptionally well suited for image-processing tasks, which often require the simultaneous handling of multiple pixels.

Verilog, a popular HDL, provides a way to model the hardware behavior and structure. This simplifies the specification, simulation, and synthesis of hardware systems by allowing developers to define their designs at a high level of abstraction. When combined, FPGAs and Verilog provide a powerful platform for implementing efficient image-processing algorithms [45–49].

Key Benefits of Using FPGAs for Image Processing

1. *Parallelism:* FPGAs can simultaneously process multiple data streams. This parallelism is crucial for real-time image processing, in which individual pixels or blocks of pixels can be handled independently. For example, while one processing unit handles edge detection, another can concurrently apply color transformations.
2. *Customization:* The ability to reprogram FPGAs allows developers to tailor their hardware according to specific application requirements. Custom architecture can be built to optimize resource usage and increase the processing speed of specific algorithms.
3. *Latency:* FPGAs typically offer a lower latency than CPU- or GPU-based solutions. In many image-processing applications, rapid feedback is essential, making FPGAs an attractive option for maintaining high throughput.
4. *Energy efficiency:* Energy usage is frequently a crucial factor in embedded systems. FPGAs can be optimized for power efficiency, providing significant advantages over traditional processing units in mobile and portable applications.

Image-Processing Algorithms Suitable for FPGA Implementation

Several image-processing algorithms can benefit from FPGA optimization. The following are some examples.

1. *Filtering operations:* Filtering is fundamental to image processing for noise reduction and feature enhancement. Gaussian blur, median filters, and edge detection filters (such as Sobel and Canny filters) are examples of common filters. Implementing these filters on FPGAs allows parallel pixel processing, which significantly increases performance.
2. *Morphological operations:* Dilation, erosion, opening, and closing are morphological processes necessary for feature extraction and picture analysis. These operations can be effectively implemented concurrently on FPGAs, thereby providing fast and efficient processing.
3. *Image transformation:* Transformations such as Fourier transforms, wavelet transforms, and histogram equalization are often computation-intensive and can yield significant performance improvements when executed on FPGAs. The parallel execution of these algorithms can drastically reduce the time required for computations.
4. *Object detection and recognition:* Algorithms for object detection, such as those based on convolutional neural networks (CNNs), can be efficiently implemented on FPGAs. Although traditionally associated with GPUs, FPGAs can be optimized to perform the matrix multiplications and convolution operations required for CNNs by leveraging their parallel processing architecture.

Optimization Techniques

To maximize the performance of image processing on FPGAs using Verilog, we considered the following optimization techniques:

1. *Pipelining:* Pipelining is a common technique used in FPGA designs to increase the throughput by allowing multiple instruction stages to execute simultaneously. By breaking down processing tasks into smaller, manageable stages, designers can keep processing units continuously occupied.

2. *Parallelism:* Exploiting data-level parallelism is essential. Implementing operations that can independently process sections of an image simultaneously takes advantage of the parallel FPGA architecture.
3. *Resource sharing:* Resource sharing involves reusing hardware resources for multiple operations. By carefully scheduling operations, designers can minimize resource usage while maintaining a high performance.
4. *Fixed-point arithmetic:* Using fixed-point arithmetic rather than floating-point arithmetic can reduce complexity and increase speed in many FPGA applications. Fixed-point operations are often faster and require less hardware, making them suitable for certain image-processing tasks.
5. *Memory optimization:* Memory bandwidth can become a bottleneck. Techniques, such as using local buffers for data storage or optimizing memory access patterns, can help alleviate this issue, allowing faster access to pixel data.

CONCLUSION

Optimizing image processing with Verilog on FPGA platforms presents a wealth of opportunities for enhancing the performance and efficiency of various applications. With the ability to process data in parallel, reconfigure hardware for specific needs, and reduce latency, FPGAs can significantly advance the field of image processing. By leveraging advanced techniques, such as pipelining, parallelism, and resource sharing, developers can achieve high-performance solutions tailored to their specific application requirements. The combination of Verilog and FPGAs will surely be crucial in determining the direction of this technology in the future, as the need for real-time image processing continues to increase.

Future Scope

This study offers several directions for further investigation in the field of FPGA-based image processing. Based on these findings, the following directions are proposed.

Enhanced Algorithms

The incorporation of more sophisticated image processing algorithms, including deep learning-based techniques, should be investigated in future research. CNN and other machine learning methods are becoming increasingly popular, and their performance and accuracy can be further improved by modifying these algorithms for FPGA implementation.

Scalability and Flexibility

It is essential to look into ways to make FPGA-based systems more scalable for higher-resolution photos or more intricate algorithms. This could involve developing more efficient resource management strategies or exploring hybrid architectures that combine FPGA with other hardware accelerators.

Adaptive Hardware Architecture

Research can focus on developing adaptive FPGA architectures that can dynamically reconfigure based on the image-processing task at hand. To increase productivity and performance, this would entail developing self-optimizing systems that modify their hardware resources in real-time.

Integration with Emerging Technologies

To increase productivity and performance, this would entail developing self-optimizing systems that modify their hardware resources in real-time. These integrations could lead to innovative solutions for real-time data processing and decision-making in various domains.

Toolchain and Design Automation

Improving the design of automation tools and methodologies for Verilog-based FPGA implementations can reduce the learning curve and development time. The development of automatic optimization tools or user-friendly design environments may be the main goals of this research.

Benchmarking and Standardization

Establishing standardized benchmarking methods for FPGA-based image-processing systems can provide a more comprehensive understanding of performance metrics. This could involve developing industry-wide benchmarks and best practices to evaluate and compare different implementations.

This study demonstrates the significant advantages of utilizing Verilog HDL for FPGA-based image-processing systems, particularly in terms of processing speed, power efficiency, and real-time performance. By leveraging the parallel processing capabilities and reconfigurability of FPGAs, this study has shown that hardware acceleration can overcome the limitations of traditional software-based approaches, offering substantial improvements in various image-processing tasks.

The implemented techniques, including edge detection, image enhancement, and adaptive filtering, highlight the potential of FPGA-based designs for addressing the growing demand for real-time applications. A comparative analysis with CPU and GPU implementations underscores the effectiveness of FPGA solutions in delivering superior performance and energy efficiency.

This study advances the field by proving the viability and advantages of hardware-driven image processing and laying the groundwork for further development. The potential for further development in algorithmic sophistication, system scalability, and integration with emerging technologies suggests a promising trajectory for continued innovation in FPGA-based image-processing solutions.

REFERENCES

1. Menaka DR, Janarthanan DR, Deeba DK. RETRACTED: FPGA implementation of low power and high-speed image edge detection algorithm. Microprocessors and Microsystems. 2020;75:103053. DOI: 10.1016/j.micpro.2020.103053.
2. Badiger NA, Muragod J, Pattar MP, Gundlur P, Bhadri S. FPGA implementation of image enhancement using Verilog HDL. Int Res J Eng Technol. 2020;7(6):5663-8.
3. Kabbai L, Sghaier A, Douik A, Machhout M. FPGA implementation of filtered image using 2D Gaussian filter. Int J Adv Comput Sci Appl. 2016;7(7):517–520. DOI: 10.14569/IJACSA.2016.070771.
4. Chugh N, Vasista V, Purini S, Bondhugula U. A DSL compiler for accelerating image processing pipelines on FPGAs. Proc Int Conf Parallel Archit Compilation Tech. 2016:327–38. DOI: 10.1145/2967938.2967969.
5. Xu Q, Varadarajan S, Chakrabarti C, Karam LJ. A distributed canny edge detector: Algorithm and FPGA implementation. IEEE Trans Image Process. 2014;23:2944–60. DOI: 10.1109/tip.2014.2311656.
6. Siddiqui F, Amiri S, Minhas UI, Deng T, Woods R, Rafferty K, et al. FPGA-based processor acceleration for image processing applications. J Imaging. 2019;5:16. DOI: 10.3390/jimaging5010016.
7. Qasaimeh M, Denolf K, Lo J, Vissers K, Zambreno J, Jones PH. Comparing energy efficiency of CPU, GPU, and FPGA implementations for vision kernels. Proc IEEE Int Conf Embed Softw Syst (ICESS). 2019:1–8. DOI: 10.1109/ICESS.2019.8782524.
8. Liyakat KKS. Machine learning approach using artificial neural networks to detect malicious nodes in IoT networks. In: Udgata SK, Sethi S, Gao XZ, editors. Intelligent systems: Proceedings of 3rd Int Conf Machine Learning, IoT, and Big Data (ICMIB 2023). Springer Nature Singapore; 2024. p. 123–34. DOI: 10.1007/978-981-99-3932-9_12.
9. Pradeepa M, Jamberi K, Sajith S, Bai MR, Prakash A, Kutubuddin sayyad liyakat kazi. Student health detection using a machine learning approach and IoT. Proc IEEE Mysore Subsection Int Conf (MysuruCon). 2022. DOI: 10.1109/MysuruCon55714.2022.9972445.
10. Liyakat KKS. Detecting malicious nodes in IoT networks using machine learning and artificial neural networks. Proc IEEE Int Conf Emerg Smart Comput Informatics (ESCI). Pune, India; 2023:1–5. DOI: 10.1109/ESCI56872.2023.10099544.

11. Singh J, Laurenti R, Sinha R, Frostell B. Progress and challenges to the global waste management system. Waste Manag Res. 2014;32:800–12. DOI: 10.1177/0734242X14537868.
12. Liyakat KKS. Machine learning approach using artificial neural networks to detect malicious nodes in IoT networks. In: Shukla PK, Mittal H, Engelbrecht A, editors. Computer vision and robotics: Proceedings of CVR 2023. Springer Nature Singapore; 2023. p. 27–37. DOI: 10.1007/978-981-99-4577-1_3.
13. Jain V, Jadhav N, Verhelst M. Enabling real-time object detection on low-cost FPGAs. J Real-Time Image Process. 2022;19:217–29. DOI: 10.1007/s11554-021-01177-w.
14. Pereira P, Linhares Silva A, Machado R, Silva J, Durães D, Machado J, et al. Comparison of different deployment approaches of FPGA-based hardware accelerators for 3D object detection models. In: Marreiros G, Martins B, Paiva A, Ribeiro B, Sardinha A, editors. Progress in Artificial Intelligence. EPIA 2022. Lecture Notes in Computer Science. vol. 13566. Cham: Springer; 2022. p. 285–96. DOI: 10.1007/978-3-031-16474-3_24.
15. Mani VRS, Saravanaselvan A, Arumugam NJ. Performance comparison of CNN, QNN, and BNN deep neural networks for real-time object detection using ZYNQ FPGA node. Microelectron J. 2022;119:105319. DOI: 10.1016/j.mejo.2021.105319.
16. Li C, Xu R, Lv Y, Zhao Y, Jing W. Edge real-time object detection and DPU-based hardware implementation for optical remote sensing images. Remote Sens. 2023;15:3975. DOI: 10.3390/rs 15163975.
17. Neeraja P, Kumar RG, Kumar MS, Liyakat KKS, Vani MS. DL-based somnolence detection for improved driver safety and alertness monitoring. Proc IEEE Int Conf Comput Power Commun Technol (IC2PCT). Greater Noida, India; 2024:589–94. DOI: 10.1109/IC2PCT60090.2024. 10486714.
18. Rzaev E, Khanaev A, Amerikanov A. Neural network for real-time object detection on FPGA. Proc IEEE Int Conf Ind Eng Appl Manuf (ICIEAM). 2021:719–23. DOI: 10.1109/ICIEAM51226.2021. 9446384.
19. Mansoub SK, Abri R, Yarıcı AH. Concurrent real-time object detection on multiple live streams using optimization CPU and GPU resources in YOLOv3. SIGNAL 2019, The Fourth International Conference on Advances in Signal, Image and Video Processing. Athens, Greece: IARIA; 2019 Jun 2-6. Pages 23-28. DOI: 10.1109/SIGNAL.2019.0007.
20. Veena C, Sridevi M, Liyakat KKS, Saha B, Reddy SR, Shirisha N. HEECCNB: An efficient IoT-cloud architecture for secure patient data transmission and accurate disease prediction in healthcare systems. Proc IEEE Int Conf Image Information Process (ICIIP). Solan, India; 2023:407–10. DOI: 10.1109/ICIIP61524.2023.10537627.
21. Prasad KR, Karanam SR, Ganesh D, Liyakat KKS, Talasila V, Purushotham P. AI in public-private partnership for IT infrastructure development. J High Technol Manag Res. 2024;35:100496. DOI: 10.1016/j.hitech.2024.100496.
22. Pauwels K, Tomasi M, Diaz Alonso JD, Ros E, Van Hulle MM. A comparison of FPGA and GPU for real-time phase-based optical flow, stereo, and local image features. IEEE Trans Comput. 2012;61:999–1012. DOI: 10.1109/TC.2011.120.
23. Kazi KKS. Machine learning (ML)-based braille Lippi characters and numbers detection and announcement system for blind children in learning. In: Sart G, editor. Social Reflections of Human–Computer Interaction in Education, Management, and Economics. Hershey, PA: IGI Global; 2024. pp. 16–39. DOI: 10.4018/979-8-3693-3033-3.ch002.
24. Jaiswal D, Kumar P. Real-time implementation of moving object detection in UAV videos using GPUs. J Real-Time Image Process. 2020;17:1301–1317. DOI: 10.1007/s11554-019-00888-5.
25. Kyrkou C, Theocharides T. A parallel hardware architecture for real-time object detection with support vector machines. IEEE Trans Comput. 2012;61:831-842. DOI: 10.1109/TC.2011.113.
26. Kumar S, Mahapatra R, Singh A. Automatic modulation recognition: An FPGA implementation. IEEE Commun Lett. 2022;26:2062–2066. DOI: 10.1109/LCOMM.2022.3184771.

27. Patel K, Nagora U, Joshi HC, et al. LabVIEW-FPGA-based real-time data acquisition system for ADITYA-U heterodyne interferometry. IEEE Trans Plasma Sci. 2021;49:1891–1897. DOI: 10.1109/TPS.2021.3082159.
28. Sharma S. Low power high speed FPGA design of lossless medical image compression using optimal deep neural network. Multimed Tools Appl. 2023;83:40569–40605. DOI: 10.1007/s11042-023-16958-7.
29. Liu CC, Lee TT, Xiao SR, Lin YC, Lin YY, Wong CC. Real-time FPGA-based balance control method for a humanoid robot pushed by external forces. Applied Sciences. 2020;10:2699. DOI: 10.3390/app10082699.
30. Chowdary MN, Bussa S, Chowdary CK, Gupta M. Automated pipeline for the deployment using OpenShift. Procedia Computer Science. 2022;215:220-229. DOI: 10.1016/j.procs.2022.12.025.
31. Maheshwari S. Mathematical model and statistical assessment of Non Linear scheduling optimization in cloud environment using improved cuckoo search algorithm. Communications on Applied Nonlinear Analysis. 2024;31:294-310. DOI: 10.52783/cana.v31.569.
32. Khan L, Isanaka SP, Liou F. FPGA-based sensors for distributed digital manufacturing systems: A state-of-the-art review. Sensors. 2024;24:7709. DOI: 10.3390/s24237709, PubMed: 39686246.
33. Zhang Z, Li J. A review of artificial intelligence in embedded systems. Micromachines. 2023;14:897. DOI: 10.3390/mi14050897.
34. Nguyen DD, Nguyen DT, Le MT, Nguyen QC. FPGA-SoC implementation of YOLOv4 for flying-object detection. J Real-Time Image Process. 2024;21:63. DOI: 10.1007/s11554-024-01440-w.
35. Mao H, Yao S, Tang T, Li B, Yao J, Wang Y. Towards real-time object detection on embedded systems. IEEE Trans Emerg Top Comput. 2018;6:417-431. DOI: 10.1109/TETC.2016.2593643.
36. Çambay VY, Uçar A, Arserim MA. Object detection on FPGAs and GPUs by using accelerated deep learning. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey. 2019. pp. 1-5. DOI: 10.1109/IDAP.2019.8875870.
37. Kazi KKS. Machine learning-based pomegranate disease detection and treatment. In: Haq MZ, Ali I, editors. Revolutionizing Pest Management for Sustainable Agriculture (2 Volumes). Hershey, PA: IGI Global; 2024. pp. 469–498. DOI: 10.4018/979-8-3693-3061-6.ch019.
38. Ma X, Najjar WA, Roy-Chowdhury AK. Evaluation and acceleration of high-throughput fixed-point object detection on FPGAs. IEEE Trans Circuits Syst Video Technol. 2014;25:1051-1062.
39. Kazi KKS, Mahant MA. Machine Learning-Driven Internet of Things (MLIoT)-Based Healthcare Monitoring System. In: Wickramasinghe N, editor. Digitalization and the Transformation of the Healthcare Sector. 2024. pp. 205–236. DOI: 10.4018/979-8-3693-9641-4.ch007.
40. Fernández-Sanjurjo M, Mucientes M, Brea VM. Real-time multiple object visual tracking for embedded GPU systems. IEEE Internet Things J. 2021;8:9177-9188. DOI: 10.1109/JIOT.2021.3056239.
41. Baobaid A, Meribout M, Tiwari VK, Pena JP. Hardware accelerators for real-time face recognition: A survey. IEEE Access. 2022;10:83723-83739. DOI: 10.1109/ACCESS.2022.3194915.
42. Asano S, Maruyama T, Yamaguchi Y. Performance comparison of FPGA, GPU and CPU in image processing. In: International Conference on Field Programmable Logic and Applications; IEEE; 2009:126-131. DOI: 10.1109/FPL.2009.5272532.
43. Kazi KKS. AI-Driven-IoT (AIIoT)-Based Decision Making in Drones for Climate Change: KSK Approach. In: Aouadni S, Aouadni I, editors. Recent Theories and Applications for Multi-Criteria Decision-Making. Hershey, PA: IGI Global; 2024. pp. 311–340. DOI: 10.4018/979-8-3693-6502-1.ch011.
44. Mounir TA, Boumerdassi S, Benhamada A, Alla AM, Kherarba M. Performance evaluation of basic image processing algorithms in CPU, GPU, Raspberry Pi and FPGA. Int J Comput Sci Eng. 2020;9:312-325.
45. Kazi KKS. Machine Learning-Driven Internet of Medical Things (ML-IoMT)-Based Healthcare Monitoring System. In: Soufiene BO, Chakraborty C, editors. Responsible AI for Digital Health and Medical Analytics. Hershey, PA: IGI Global; 2024. pp. 49–86. DOI: 10.4018/979-8-3693-6294-5.ch003.

46. Esen F, Degirmenci A, Karal O. Implementation of the object detection algorithm (YOLOV3) on FPGA. 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), Elazig, Turkey. 2021. pp. 1-6. DOI: 10.1109/ASYU52992.2021.9599073.
47. Kazi KKS. AI-Driven-IoT(AIIoT)-Based Decision Making in Kidney Diseases Patient Healthcare Monitoring: KSK Approach for Kidney Monitoring. In: Polat LÖ, Polat O, editors. AI-Driven Innovation in Healthcare Data Analytics. Hershey, PA: IGI Global; 2024. pp. 277–306. DOI: 10.4018/979-8-3693-7277-7.ch009.
48. Struyf L, De Beugher S, Van Uytsel D, Kanters F, Goedemé T. The battle of the giants: a case study of GPU vs FPGA optimisation for real-time image processing. In: Proceedings of the 4th International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS-2014). Lisbon, Portugal. Volume: 4. DOI: 10.5220/0004730301120119. Setúbal, Portugal: SCITEPRESS; 2014.
49. Liyakat KK. Malicious Node Detection in IoT Networks Using Artificial Neural Networks. In: Singh VK, Kumar Sagar A, Nand P, Astya R, Kaiwartya O, editors. Intelligent Networks: Techniques, and Applications. Boca Raton: CRC Press; 2024. p.182. DOI: 10.1201/9781003541363.

