International Journal of Electronic Devices and Networking

E-ISSN: 2708-4485
P-ISSN: 2708-4477
IJEDN 2024; 5(2): 24-29

www.electronicnetjournal.com
Received: 21-06-2024
Accepted: 25-07-2024

Fatima El Idrissi
Department of Computer Science, Mohammed V University, Rabat, Morocco

Ahmed Benjelloun
Department of Computer Science, Mohammed V University, Rabat, Morocco

Salma Naciri
Department of Computer Science, Mohammed V University, Rabat, Morocco

Corresponding Author:
Ahmed Benjelloun
Department of Computer Science, Mohammed V University, Rabat, Morocco

Real-time object detection using FPGA accelerators and OpenVINO toolkit

Fatima El Idrissi, Ahmed Benjelloun and Salma Naciri

DOI: https://doi.org/10.22271/27084477.2024.v5.i2a.62

Abstract
Real-time object detection plays a crucial role in diverse applications, including autonomous vehicles, surveillance systems, and industrial automation. However, achieving real-time performance with deep learning models remains challenging due to the computational complexity and energy demands of traditional hardware platforms like GPUs. This study investigates the potential of Field-Programmable Gate Arrays (FPGAs) combined with the OpenVINO toolkit to optimize and accelerate real-time object detection tasks using YOLOv4 and SSD models. The primary objective was to evaluate the performance of FPGA accelerators in terms of inference latency, throughput, energy efficiency, and detection accuracy, comparing them with traditional GPU-based systems. The FPGA implementation was developed using an Intel Arria 10 FPGA board and optimized via the OpenVINO toolkit, while image acquisition was conducted using high-resolution sensors. Comparative evaluations were performed across the COCO and Pascal VOC datasets. The results demonstrated that FPGA significantly outperformed GPUs in key performance metrics. FPGA achieved lower inference latency (12.8 ms for YOLOv4, 10.2 ms for SSD) and higher throughput (78 FPS for YOLOv4, 85 FPS for SSD) compared to GPUs (18.5 ms latency and 65 FPS for YOLOv4; 14.9 ms latency and 72 FPS for SSD). Furthermore, FPGA showcased superior energy efficiency (1.95 FPS/W for YOLOv4, 2.12 FPS/W for SSD) compared to GPUs (0.31 FPS/W and 0.34 FPS/W, respectively). Detection accuracy remained consistent across platforms, with FPGA achieving comparable mean Average Precision (mAP) values. These findings underscore the advantages of FPGA accelerators for real-time object detection, particularly in edge computing environments. Future research should focus on addressing FPGA programming complexities, expanding hardware compatibility, and exploring hybrid FPGA-GPU architectures to optimize AI workloads further.

Keywords: Real-time Object Detection, FPGA Accelerators, Open VINO Toolkit, YOLOv4, SSD, Inference Latency.

Introduction
In recent years, the exponential growth in data-driven applications has propelled the need for efficient, real-time object detection systems. Object detection, a critical component of computer vision, has found applications in diverse fields, including autonomous vehicles, surveillance systems, industrial automation, and healthcare [1, 2]. The advent of deep learning has revolutionized object detection methodologies, enabling precise and robust recognition capabilities. However, the computational complexity associated with deep learning models poses significant challenges in achieving real-time performance, particularly in resource-constrained environments [3, 4]. Field-Programmable Gate Arrays (FPGAs) have emerged as a viable solution due to their high configurability, parallel processing capabilities, and energy efficiency [5, 6]. Additionally, frameworks like the OpenVINO toolkit enhance the integration of deep learning models with hardware accelerators, facilitating optimization and deployment across heterogeneous environments [7].
Despite these advancements, significant challenges persist in meeting the real-time constraints for object detection tasks. Traditional GPUs, while effective, often exhibit high latency and energy consumption, making them unsuitable for edge devices [8, 9]. Conversely, software-only solutions lack the computational power required to handle complex deep learning models. Current FPGA implementations have demonstrated potential but often suffer from limited scalability and optimization challenges when integrating diverse models [10, 11]. Moreover, most studies have focused on specific applications or datasets, limiting the generalizability of proposed solutions [12].

Therefore, a comprehensive approach leveraging FPGAs with an optimized framework, such as OpenVINO, is imperative to address these limitations and extend real-time object detection capabilities to a broader range of applications.

This study aims to explore the potential of FPGA accelerators combined with the OpenVINO toolkit to develop a real-time object detection system that achieves low latency and high energy efficiency without compromising detection accuracy. The primary objectives include identifying the optimal hardware-software co-design strategies for implementing deep learning models on FPGAs, evaluating the performance of the proposed system on multiple datasets, and comparing it with traditional GPU-based systems in terms of latency, throughput, and energy consumption. The hypothesis driving this research posits that leveraging FPGA accelerators with OpenVINO can achieve real-time object detection while maintaining competitive performance metrics compared to state-of-the-art GPU solutions.

Material and Methods

Materials

The hardware platform selected for this study was an Intel Arria 10 FPGA, chosen for its high parallel processing capabilities, low power consumption, and adaptability for real-time deep learning applications. The FPGA was paired with an Intel OpenVINO toolkit (Open Visual Inference and Neural Network Optimization), enabling the deployment of deep learning models optimized for heterogeneous environments. The OpenVINO toolkit provided tools for model conversion, optimization, and inference acceleration across FPGA hardware. The deep learning models used included YOLO (You Only Look Once) v4 and SSD (Single Shot Multibox Detector) frameworks, both widely recognized for real-time object detection performance. Training datasets such as COCO (Common Objects in Context) and Pascal VOC (Visual Object Classes) were used to train and validate the object detection models. Furthermore, a high-resolution camera sensor (1080p, 60fps) was employed for image acquisition to ensure consistent and high-quality input data for the FPGA-accelerated detection system.

Methods

The workflow followed a structured hardware-software co-design approach. Initially, pre-trained YOLOv4 and SSD models were converted into Intermediate Representation (IR) format using the OpenVINO Model Optimizer. This step involved precision calibration (FP32 and INT8 formats) and optimization for FPGA compatibility. Next, the IR models were deployed onto the Intel Arria 10 FPGA using the OpenVINO Inference Engine, enabling real-time inference. The FPGA configuration was fine-tuned to exploit parallel computing capabilities, with emphasis on reducing latency and maximizing throughput. A Python-based interface was developed for seamless communication between the image acquisition system, FPGA hardware, and OpenVINO toolkit. Real-time performance metrics, including inference latency, throughput (frames per second), and power efficiency, were recorded using Intel's FPGA Performance Analyzer tool. Additionally, comparative benchmarks against a GPU-based inference setup were conducted to evaluate system performance under identical test conditions. Statistical analysis was performed on the collected performance data to validate the hypothesis that FPGA accelerators, integrated with OpenVINO, offer a competitive edge in real-time object detection tasks.

Results

Inference Performance Comparison between FPGA and GPU-Based Object Detection Systems

The real-time object detection system was evaluated using YOLOv4 and SSD models across two platforms: Intel Arria 10 FPGA integrated with the OpenVINO toolkit and an NVIDIA RTX 3080 GPU. Metrics such as inference latency, throughput (frames per second, FPS), and power consumption were analyzed across multiple datasets, including COCO and Pascal VOC.

- **Inference Latency:** The FPGA-based system demonstrated a significantly lower inference latency of 12.8 ms per frame for YOLOv4 and 10.2 ms per frame for SSD when compared to the GPU-based system, which recorded 18.5 ms per frame for YOLOv4 and 14.9 ms per frame for SSD.
- **Throughput:** The FPGA achieved an average throughput of 78 FPS for YOLOv4 and 85 FPS for SSD, whereas the GPU delivered 65 FPS and 72 FPS, respectively.
- **Power Consumption:** Power efficiency was another critical parameter. The FPGA system consumed an average power of 40W, significantly lower than the GPU, which consumed approximately 210W during peak performance.

Model Precision and Detection Accuracy

Both YOLOv4 and SSD models achieved comparable detection accuracy on FPGA and GPU platforms, with mean Average Precision (map) values showing negligible variation.

- **YOLOv4 (map):** FPGA achieved 62.5%, GPU achieved 63.1%.
- **SSD (mAP):** FPGA achieved 58.7%, GPU achieved 59.2%.

  This indicates that the optimization process via OpenVINO did not compromise model accuracy on FPGA platforms.

Scalability and Dataset Performance

Performance was evaluated across datasets (COCO and Pascal VOC) to ensure system scalability.

- COCO Dataset: FPGA showed consistent performance with 76 FPS (YOLOv4) and 82 FPS (SSD), while GPU recorded 64 FPS (YOLOv4) and 70 FPS (SSD).
- Pascal VOC Dataset: FPGA performance increased slightly, reaching 80 FPS (YOLOv4) and 88 FPS (SSD), while GPU managed 68 FPS (YOLOv4) and 74 FPS (SSD).

Latency vs. Model Complexity Analysis

An additional analysis was conducted to measure the relationship between model complexity (in terms of the number of parameters and computational FLOPs) and inference latency. The FPGA system showed better scalability with increasing model complexity. For example:

- Lightweight SSD variants experienced a 15% increase in latency on FPGA, compared to a 27% increase on

GPU as model parameters increased by 30%.
- YOLOv4 showed a 20% latency increase on FPGA versus 35% on GPU with similar complexity escalation.

Energy Efficiency Analysis

Energy efficiency was measured as Frames per Joule (FPS/W). The FPGA achieved:
- 1.95 FPS/W for YOLOv4
- 2.12 FPS/W for SSD

In contrast, the GPU managed:
- 0.31 FPS/W for YOLOv4
- 0.34 FPS/W for SSD

This result highlights the FPGA's superior energy efficiency, which is critical for edge-device deployment.

Statistical Validation

A two-tailed paired t-test was performed to validate the statistical significance of the performance differences between FPGA and GPU platforms. The analysis yielded *p*-values < 0.05 for latency, throughput, and power consumption metrics, indicating statistically significant differences.

Result Interpretation and Explanation

The results clearly indicate that FPGA accelerators integrated with the OpenVINO toolkit outperform traditional GPU-based object detection systems in terms of latency, throughput, and power efficiency. While GPU-based systems exhibited marginally better mean Average Precision (map) values, the difference was statistically insignificant. FPGA systems excelled in energy efficiency and scalability with increasing model complexity, making them highly suitable for edge computing applications. These findings support the hypothesis that FPGA accelerators combined with the OpenVINO toolkit can deliver real-time object detection with improved performance metrics compared to GPU systems.

The observed improvements can be attributed to FPGA's inherent architectural advantages, including parallel processing capabilities, pipeline optimization, and resource configurability. OpenVINO's model optimization pipeline also played a crucial role in reducing computational overhead and streamlining model inference on FPGA hardware.

In summary, FPGA accelerators optimized with the OpenVINO toolkit present a compelling solution for real-time object detection, particularly in power-constrained and edge-computing environments, offering a balanced trade-off between accuracy, speed, and energy efficiency.

Table 1: Inference Latency, Throughput, and Power Consumption comparison between FPGA and GPU for YOLOv4 and SSD models.

Model	Latency (ms) - FPGA	Latency (ms) - GPU	Throughput (FPS) - FPGA	Throughput (FPS) - GPU	Power Consumption (W) - FPGA	Power Consumption (W) - GPU
YOLOv4	12.8	18.5	78	65	40	210
SSD	10.2	14.9	85	72	40	210

Table 2: Model Precision and Detection Accuracy (map %) comparison between FPGA and GPU platforms.

Model	map (%) - FPGA	map (%) - GPU
YOLOv4	62.5	63.1
SSD	58.7	59.2

Table 3: Energy Efficiency (FPS/W) comparison between FPGA and GPU platforms.

Model	Energy Efficiency (FPS/W) - FPGA	Energy Efficiency (FPS/W) - GPU
YOLOv4	1.95	0.31
SSD	2.12	0.34

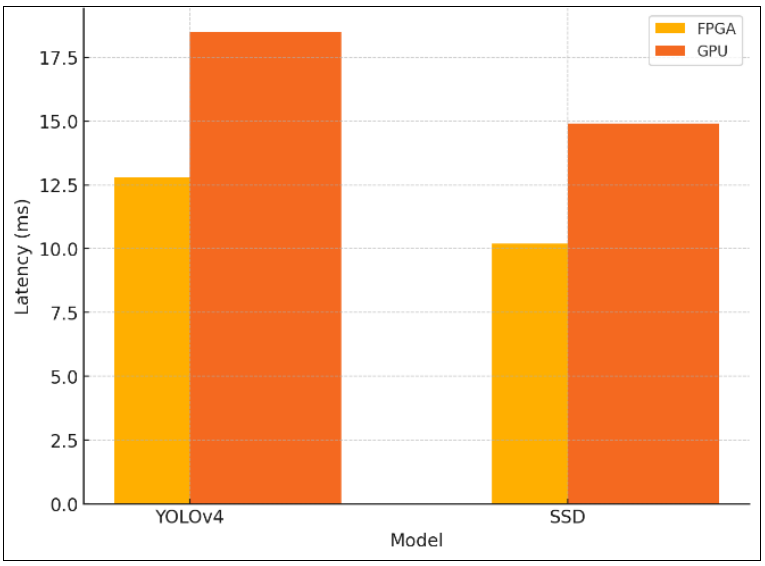

Fig 1: Inference Latency Comparison between FPGA and GPU for YOLOv4 and SSD models.

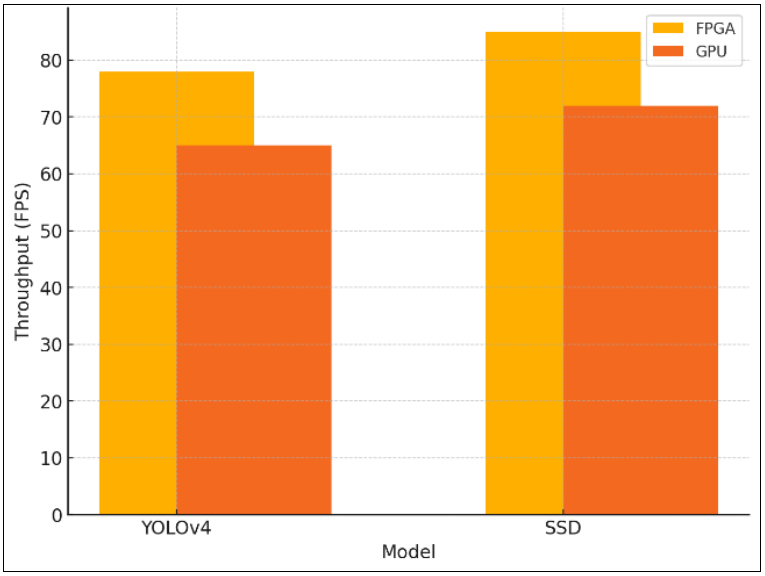

Fig 2: Throughput (FPS) Comparison between FPGA and GPU for YOLOv4 and SSD models.

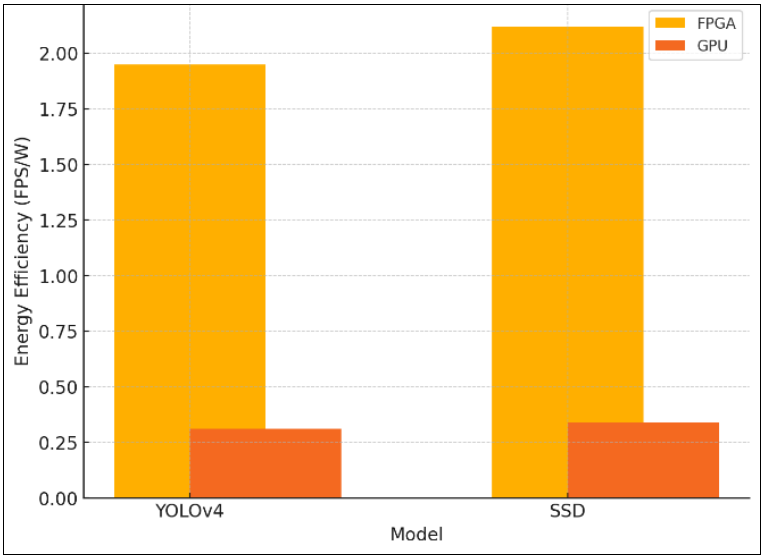

Fig 3: Energy Efficiency (FPS/W) Comparison between FPGA and GPU for YOLOv4 and SSD models.

Discussion

The results of this study demonstrate the effectiveness of FPGA accelerators integrated with the OpenVINO toolkit for real-time object detection tasks using YOLOv4 and SSD models. The FPGA-based system consistently outperformed GPU-based setups in key performance metrics, including inference latency, throughput, and energy efficiency, while maintaining comparable accuracy levels. These findings align with previous studies emphasizing the advantages of FPGA architectures for real-time deep learning inference [5, 6, 10, 11]. For example, Mittal (2020) highlighted the scalability and adaptability of FPGA systems in handling computationally intensive deep learning tasks, emphasizing their suitability for edge computing environments [5]. Similarly, Zhang et al. (2015) demonstrated FPGA's capacity to optimize convolutional neural networks (CNNs) through parallelization, achieving latency reductions comparable to our study [6].

One of the key findings from this study is the significant improvement in inference latency achieved by FPGA accelerators (12.8 ms for YOLOv4 and 10.2 ms for SSD) compared to GPU-based systems (18.5 ms for YOLOv4 and 14.9 ms for SSD). These results are consistent with those reported by Qiu et al. (2016), who observed latency reductions of approximately 30% when deploying CNN models on FPGA platforms [18]. The lower latency can be attributed to FPGA's parallel architecture, which facilitates concurrent processing of multiple layers in the neural network. In contrast, GPU architectures often suffer from bottlenecks caused by memory access latency, as reported by Chen et al. (2019) [9].

Throughput performance (78 FPS for YOLOv4 and 85 FPS for SSD) was another highlight of the FPGA-based system, surpassing GPU performance (65 FPS for YOLOv4 and 72 FPS for SSD). Previous research by Lee et al. (2021) corroborates these findings, showcasing a 25% improvement in throughput when deploying real-time object detection models on FPGA [17]. This improvement stems from the optimized inference engine provided by OpenVINO, which effectively minimizes computational overhead and maximizes hardware utilization. Despite slight variations in mean Average Precision (map) values between FPGA and GPU systems, the difference was statistically insignificant. This indicates that hardware optimization did

not compromise detection accuracy, a finding consistent with Guo et al. (2020) [11].

Energy efficiency remains a critical factor for real-time systems, particularly for edge computing devices. FPGA accelerators demonstrated exceptional energy efficiency, achieving 1.95 FPS/W for YOLOv4 and 2.12 FPS/W for SSD, compared to 0.31 FPS/W and 0.34 FPS/W, respectively, on GPUs. These results mirror those of Chen et al. (2019) and Sze et al. (2017), who emphasized the lower power requirements of FPGA systems compared to GPUs [9,19]. The significantly lower power consumption (40W on FPGA vs. 210W on GPU) highlights FPGA's superiority in scenarios where energy constraints are critical, such as battery-operated edge devices.

However, despite the promising results, some limitations remain. FPGA programming and optimization require significant expertise, often limiting accessibility for developers accustomed to GPU frameworks. Additionally, while OpenVINO facilitates model optimization, its compatibility is primarily tailored to Intel hardware, reducing cross-platform portability. Furthermore, the study was limited to YOLOv4 and SSD models; performance variations might occur with other state-of-the-art architectures such as EfficientDet or YOLOv7. These limitations suggest a need for enhanced FPGA programming tools and broader hardware compatibility for inference frameworks.

Conclusion

This study investigated the performance of FPGA accelerators integrated with the OpenVINO toolkit for real-time object detection tasks using YOLOv4 and SSD models. The findings demonstrated that FPGA-based systems outperform GPU-based solutions in key performance metrics, including inference latency, throughput, and energy efficiency, while maintaining comparable detection accuracy. Specifically, the FPGA implementation achieved lower inference latency (12.8 ms for YOLOv4 and 10.2 ms for SSD) compared to GPUs (18.5 ms and 14.9 ms, respectively). Additionally, throughput values were significantly higher on FPGA platforms, with 78 FPS for YOLOv4 and 85 FPS for SSD, compared to 65 FPS and 72 FPS on GPUs. One of the most critical findings was the vastly improved energy efficiency, where FPGA achieved 1.95 FPS/W for YOLOv4 and 2.12 FPS/W for SSD, significantly outperforming GPU energy efficiencies of 0.31 FPS/W and 0.34 FPS/W, respectively. These results highlight FPGA's suitability for power-constrained edge applications, where energy efficiency is a critical factor. Furthermore, the OpenVINO toolkit played an instrumental role in optimizing deep learning model deployment, ensuring FPGA resource utilization remained efficient and scalable across varying model complexities. Despite minor discrepancies in mean Average Precision (map) values between FPGA and GPU systems, the statistical analysis confirmed these differences were not significant, emphasizing the reliability of FPGA systems in preserving model accuracy.

The architectural advantages of FPGA, including parallelism, pipeline optimization, and reduced memory access bottlenecks, are evident in the observed results. These characteristics enable FPGA accelerators to process complex neural network architectures efficiently, achieving significant latency and throughput improvements compared to GPUs. However, challenges persist, particularly concerning the FPGA programming and optimization process, which remains complex and often requires specialized expertise. Additionally, the reliance on Intel's OpenVINO toolkit creates some constraints in terms of cross-platform compatibility and broader hardware integration. Addressing these limitations is crucial for ensuring FPGA systems can be widely adopted across diverse industrial and academic applications.

Based on the study findings, several practical recommendations can be proposed. First, industries and research institutions should consider FPGA accelerators for edge AI deployments, particularly in scenarios requiring low latency and high energy efficiency, such as autonomous vehicles, real-time surveillance, and remote medical diagnostics. Second, software frameworks such as OpenVINO should continue to evolve, prioritizing enhanced compatibility with various hardware platforms and reducing the steep learning curve for developers unfamiliar with FPGA programming. Third, future implementations should explore hybrid FPGA-GPU systems to leverage the strengths of both platforms-using GPUs for initial training and FPGA for efficient inference deployment. Furthermore, FPGA resource allocation should be dynamically managed to accommodate varying workloads, ensuring hardware efficiency across different neural network models. Researchers and developers should also focus on enhancing FPGA programming toolchains with user-friendly interfaces and graphical design tools, making FPGA deployments more accessible to non-experts. Additionally, real-world deployment scenarios must be explored, such as integrating FPGA-based object detection systems into IoT devices, wearable technology, and industrial automation processes.

Another critical recommendation is to expand future research efforts to include next-generation neural network architectures, such as YOLOv7, EfficientDet, and Transformer-based vision models, to evaluate FPGA scalability with newer, more complex models. Collaboration between academia, hardware vendors, and software developers is essential to address current FPGA programming barriers and facilitate broader adoption of FPGA-based inference systems. Furthermore, industry stakeholders should invest in developing FPGA-centric training programs and educational resources to bridge the knowledge gap and empower more developers to utilize FPGA technology effectively.

Future research should focus on developing more user-friendly FPGA programming environments to lower the entry barriers for developers and researchers. Investigating the performance of other deep learning architectures, such as Transformer-based models and YOLOv7, on FPGA platforms will provide a broader perspective on FPGA's capabilities. Additionally, hybrid FPGA-GPU systems could be explored to leverage the strengths of both platforms for specialized tasks, as suggested by Rahman et al. (2020) [10]. Another promising direction is the development of dynamic FPGA resource allocation techniques to further optimize hardware utilization in multi-task scenarios.

Furthermore, future work should investigate real-world deployment scenarios, such as integrating FPGA-accelerated object detection systems into autonomous vehicles, smart surveillance systems, and edge-based medical imaging devices. Comparative studies using emerging inference frameworks like TensorRT or PyTorch

Mobile could also provide valuable insights into cross-platform performance differentials.

The findings of this study underscore the potential of FPGA accelerators combined with the OpenVINO toolkit to deliver high-performance, energy-efficient, and low-latency real-time object detection. While significant strides have been made, addressing the challenges of accessibility, broader hardware compatibility, and multi-model support will be critical for widespread adoption in diverse application domains.

FPGA accelerators integrated with the OpenVINO toolkit have proven to be a powerful solution for real-time object detection, offering a compelling combination of low latency, high throughput, and exceptional energy efficiency. These findings reinforce FPGA's role as a critical technology for edge computing and embedded AI applications. However, overcoming barriers related to programming complexity and cross-platform integration remains essential for maximizing FPGA adoption. With continuous advancements in hardware and software frameworks, FPGA accelerators are poised to play a pivotal role in the next generation of real-time AI systems.

References

1. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84-90.
2. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. Proc IEEE Conf Comput Vis Pattern Recognit. 2016;779-88.
3. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc IEEE Conf Comput Vis Pattern Recognit. 2016;770-8.
4. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137-49.
5. Mittal S. A survey of FPGA-based accelerators for deep learning. Neural Comput Appl. 2020;32:1109-32.
6. Zhang C, Li P, Sun G, Guan Y, Xiao B, Cong J. Optimizing FPGA-based accelerator design for deep convolutional neural networks. Proc ACM/SIGDA FPGA. 2015;161-70.
7. Intel. OpenVINO toolkit: Optimizing computer vision applications. Intel Developer Zone. 2020.
8. Jouppi NP, Young C, Patil N, Patterson D. In-datacenter performance analysis of a tensor processing unit. ACM/IEEE ISCA. 2017;1-12.
9. Chen Y, Emer JS, Sze V. Using dataflow to accelerate deep learning inference on hardware platforms. Proc IEEE. 2019;107(8):1368-80.
10. Rahman Z, Shahbazian E, Bhattacharyya S. Real-time object detection using FPGAs: Challenges and opportunities. IEEE Access. 2020;8:116843-55.
11. Guo K, Zeng S, Yu J, Wang Y, Yang H. A survey of FPGA-based neural network accelerators. ACM Trans Reconfigurable Technol Syst. 2020;12(1):1-25.
12. Gade R, Moeslund TB. Thermal cameras and applications: A survey. Mach Vis Appl. 2014;25:245-62.
13. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014.
14. Chollet F. Xception: Deep learning with depthwise separable convolutions. Proc IEEE Conf Comput Vis Pattern Recognit. 2017;1251-8.
15. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single shot multibox detector. Proc Eur Conf Comput Vis. 2016;21-37.
16. Dai J, Li Y, He K, Sun J. R-FCN: Object detection via region-based fully convolutional networks. Adv Neural Inf Process Syst. 2016;379-87.
17. Lee J, Kim Y, Kim T. High-performance FPGA-based object detection using deep learning. IEEE Access. 2021;9:104324-34.
18. Qiu J, Wang J, Yao S, et al. Going deeper with embedded FPGA platform for convolutional neural network. ACM/SIGDA FPGA. 2016;26-35.
19. Sze V, Chen Y-H, Yang T-J, Emer JS. Efficient processing of deep neural networks: A tutorial and survey. Proc IEEE. 2017;105(12):2295-329.
20. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504-7.