10.48047/jocaaa.2026.35.07.08

Design and Implementation of a Low-Power FPGA-Based Architecture for Deep Neural Network Acceleration in Edge Computing Systems

Dr Yaser Madani (Corresponding Author)
Full Professor, Department of Management, Tadbirsaz Research Center, Tehran, Iran
dryasermadani@gmail.com
Javad Shahrabi Farahani
MSc. in Computer Software Engineering, Department of Computer Engineering, YI.C., Islamic Azad University, Tehran, Iran.
jfarahani361@gmail.com
https://orcid.org/0000-0003-0152-8168

Abstract
The increasing adoption of edge computing has created a growing demand for efficient deployment of deep neural networks (DNNs) on resource-constrained devices. While cloud-based inference offers high computational power, it often suffers from latency, bandwidth limitations, and privacy concerns, making it unsuitable for many real-time edge applications. To address these challenges, this paper proposes a low-power FPGA-based architecture for accelerating DNN inference in edge computing environments. The proposed design incorporates hardware-aware optimization techniques, including INT8 quantization, pipelined processing, parallel computation units, and optimized memory management, to enhance computational efficiency while reducing energy consumption. The architecture is designed to support convolutional neural network (CNN) workloads frequently used in edge intelligence applications such as image classification, smart sensing, and Internet of Things (IoT) systems. Performance evaluation is conducted using representative benchmark workloads, focusing on key metrics including inference latency, throughput, resource utilization, power consumption, and energy efficiency. The results indicate that the proposed architecture can achieve substantial improvements in processing performance and energy efficiency compared with conventional software-based implementations while maintaining competitive inference accuracy. In addition, the combination of quantization and hardware-level parallelism enables effective utilization of FPGA resources and supports real-time execution under strict power constraints. The proposed framework demonstrates the potential of FPGA technology as a scalable and energy-efficient platform for edge AI deployment. By integrating lightweight computation techniques with optimized hardware design strategies, the architecture provides a practical solution for next-generation intelligent edge devices and contributes to the advancement of sustainable and high-performance edge computing systems.

Keywords: Edge Computing, FPGA Acceleration, Deep Neural Networks, Edge Intelligence, INT8 Quantization, Energy Efficiency, Hardware Optimization.

1 .Introduction

The rapid development of Artificial Intelligence (AI), the Internet of Things (IoT), and smart connected systems has significantly transformed modern computing environments. In recent years, billions of sensors, mobile devices, industrial controllers, and embedded platforms have been continuously generating large volumes of data that require real-time processing and intelligent analysis. At the same time, Deep Neural Networks (DNNs), particularly Convolutional Neural

10.48047/jocaaa.2026.35.07.08

Networks (CNNs), have demonstrated remarkable performance in various applications such as image classification, object detection, medical diagnosis, autonomous driving, industrial monitoring, and smart surveillance systems. Owing to their ability to automatically learn complex patterns from large datasets, deep learning models have become one of the most influential technologies driving the current wave of intelligent systems (Yu et al., 2018; Wang & Jia, 2025).

Traditionally, the execution of deep learning workloads has relied heavily on cloud computing infrastructures, where data collected from end devices are transmitted to remote servers for storage and processing. Although cloud computing provides substantial computational resources and scalability, this centralized paradigm introduces several limitations. The continuous transmission of large volumes of data increases communication overhead and bandwidth consumption while creating latency that may be unacceptable for time-sensitive applications. Furthermore, concerns related to privacy, security, and service availability have become increasingly important as intelligent systems expand into critical domains such as healthcare, transportation, and industrial automation. As a result, researchers have increasingly focused on developing distributed computing paradigms that can move intelligence closer to data sources and reduce dependence on centralized cloud infrastructures (Shi et al., 2016). In response to these challenges, edge computing has emerged as a promising paradigm that enables data processing and decision-making at or near the source of data generation. By performing computations closer to end devices, edge computing can significantly reduce latency, minimize network congestion, improve privacy protection, and enhance system responsiveness. Consequently, the integration of artificial intelligence with edge computing, often referred to as Edge AI, has attracted considerable attention from both academic researchers and industrial practitioners.

Recent studies have highlighted the potential of Edge AI to support a wide range of real-time applications, including intelligent healthcare systems, autonomous monitoring platforms, smart manufacturing environments, and next-generation IoT services (Wang et al., 2025; Zhang et al., 2015). Despite its numerous advantages, the deployment of deep neural networks in edge environments remains a challenging task. Most edge devices operate under strict constraints in terms of computational capability, memory resources, and energy consumption. Recent surveys have emphasized that the successful deployment of edge intelligence requires lightweight machine learning models and hardware-efficient architectures capable of operating under stringent resource constraints (Hoffpauir et al., 2023). Modern deep learning models often contain millions of parameters and require extensive arithmetic operations during inference. Consequently, executing such models on conventional Central Processing Units (CPUs) can result in high inference latency and limited throughput. Although Graphics Processing Units (GPUs) provide significantly higher computational performance, their power consumption and hardware costs often exceed the requirements and limitations of many embedded and battery-powered systems. Therefore, achieving efficient DNN inference on resource-constrained edge platforms continues to represent a major research challenge. To address this issue, researchers have explored various optimization strategies, including model compression, quantization, lightweight neural architectures, and specialized hardware accelerators. Among the available hardware platforms, Field Programmable Gate Arrays (FPGAs) have emerged as one of the most promising solutions for edge intelligence. FPGA devices provide a unique combination of flexibility, reconfigurability, parallel processing capability, and energy efficiency. Unlike traditional processors that follow a fixed execution model, FPGA architectures can be customized according to the computational characteristics of neural networks, enabling efficient implementation of parallel operations and optimized data movement. These features make FPGA-based accelerators particularly suitable for deploying deep

10.48047/jocaaa.2026.35.07.08

learning applications in environments where both performance and power efficiency are critical. Recent advances in quantized neural networks and hardware-aware optimization techniques have further increased the feasibility of FPGA-based deep learning deployment. By reducing numerical precision and exploiting parallel processing structures, it becomes possible to significantly decrease memory requirements, computational complexity, and energy consumption while maintaining acceptable prediction accuracy. Nevertheless, existing studies often emphasize either performance enhancement or power reduction independently. In practical edge computing scenarios, however, a successful solution must simultaneously balance multiple objectives, including inference latency, throughput, energy efficiency, hardware resource utilization, and model accuracy. The absence of such balanced architectures continues to limit the widespread deployment of advanced deep learning models on resource-constrained edge devices. Considering these challenges, there is a growing need for efficient hardware acceleration frameworks capable of delivering high-performance neural network inference while operating within strict energy budgets. The increasing adoption of intelligent IoT devices, autonomous monitoring systems, smart sensors, and embedded AI platforms further reinforces the importance of developing scalable and low-power solutions for edge computing environments. From this perspective, FPGA-based acceleration represents a practical and promising approach for overcoming current limitations and enabling real-time intelligence at the network edge. Accordingly, this study investigates the design and implementation of a low-power FPGA-based architecture for accelerating deep neural network inference in edge computing systems. The proposed approach integrates quantized neural network models, parallel processing elements, pipelined computation, and optimized memory access mechanisms to improve computational efficiency while minimizing energy consumption. The central question guiding this research is whether a carefully optimized FPGA architecture can provide a balanced trade-off between inference performance, hardware resource utilization, and power efficiency without compromising prediction accuracy. Therefore, the primary objective of this work is to develop and evaluate an FPGA-based DNN accelerator capable of supporting real-time edge intelligence through reduced latency, enhanced throughput, and improved energy efficiency. By addressing these objectives, the proposed framework aims to contribute to the development of next-generation embedded AI systems that can effectively meet the growing demands of intelligent edge applications.

1. Related Work

The increasing demand for real-time intelligent services has stimulated extensive research on integrating artificial intelligence with edge computing infrastructures. As deep learning applications continue to expand across domains such as smart healthcare, autonomous systems, industrial automation, and intelligent surveillance, researchers have increasingly focused on developing efficient approaches for deploying deep neural networks (DNNs) on resource-constrained edge devices. Consequently, a considerable body of literature has emerged addressing challenges related to computational complexity, energy consumption, hardware acceleration, and system optimization.

One of the earliest directions of research focused on the role of edge computing as an alternative to traditional cloud-centric architectures. Shi et al. (2016) introduced the concept of edge computing and highlighted its potential to reduce latency, bandwidth consumption, and dependence on remote cloud infrastructures. Similarly, Yu et al. (2018) provided a comprehensive review of edge computing for Internet of Things (IoT) environments and demonstrated how local

10.48047/jocaaa.2026.35.07.08

processing capabilities could significantly improve the responsiveness of intelligent services. These studies established the foundation for the development of Edge AI systems, where machine learning models are executed closer to data sources rather than within centralized cloud servers.

As the volume of edge-generated data increased, researchers began exploring the integration of artificial intelligence with edge platforms. Recent surveys on Edge AI have emphasized the importance of balancing computational performance with energy efficiency. For example, studies investigating edge intelligence architectures have reported that while deep learning models provide superior prediction accuracy, their computational and memory requirements often exceed the capabilities of conventional edge devices. Consequently, efficient deployment strategies have become a major research focus in both academia and industry (Wang & Jia, 2025). To overcome resource limitations, several researchers have investigated lightweight machine learning frameworks and model optimization techniques. Hoffpauir et al. (2023) presented a comprehensive survey of edge intelligence and lightweight machine learning approaches, highlighting model compression, quantization, hardware-aware optimization, and distributed intelligence as key enablers for future edge services. Their findings indicate that balancing computational performance, energy efficiency, and deployment scalability remains a central challenge in edge AI systems. Resource-constrained Neural Architecture Search (NAS) approaches have been proposed to automatically design compact neural network architectures suitable for edge deployment. These approaches aim to reduce model complexity while maintaining acceptable levels of accuracy. Likewise, model compression and quantization techniques have gained significant attention because they decrease memory footprint and computational requirements by reducing numerical precision. Such methods are particularly valuable for embedded systems where memory bandwidth and energy consumption are critical constraints.

In parallel with algorithmic optimization, considerable research efforts have been devoted to hardware acceleration. Conventional CPU-based implementations often fail to satisfy the performance requirements of modern DNN applications, particularly in real-time environments. Although Graphics Processing Units (GPUs) offer substantial computational throughput, their high power consumption limits their applicability in many edge scenarios. Consequently, researchers have increasingly explored alternative hardware platforms capable of delivering higher energy efficiency. Among these platforms, Field Programmable Gate Arrays (FPGAs) have emerged as one of the most promising technologies for accelerating deep learning inference. FPGA devices provide customizable hardware architectures that can exploit parallelism at multiple levels while maintaining relatively low power consumption. Unlike general-purpose processors, FPGA-based implementations enable application-specific optimization of computational pipelines, memory hierarchies, and data movement patterns. These characteristics have made FPGA accelerators attractive for a wide range of embedded AI applications.

Several studies have demonstrated the effectiveness of FPGA-based accelerators for convolutional neural networks. Ma et al. (2017) proposed an optimized CNN accelerator that leveraged loop optimization and parallel execution techniques to improve computational throughput on FPGA platforms. Similarly, Qiu et al. (2016) developed an FPGA-based deep learning accelerator capable of achieving substantial speedup while maintaining energy-efficient operation. Subsequent research further improved accelerator performance through the use of pipelined processing architectures, customized processing elements, and efficient on-chip memory utilization strategies. More recent investigations have focused on quantized neural networks implemented on FPGA devices. By reducing arithmetic precision from floating-point to fixed-point or integer representations, these approaches significantly reduce hardware resource utilization and power

10.48047/jocaaa.2026.35.07.08

consumption. Studies have shown that carefully designed quantized accelerators can maintain competitive classification accuracy while substantially improving throughput and energy efficiency. Furthermore, hardware-aware neural network optimization techniques have enabled better alignment between model structures and FPGA resource constraints, resulting in more practical solutions for edge deployment. Despite these advances, several limitations remain evident in the existing literature. Many edge AI studies primarily concentrate on system-level architectures and distributed computing frameworks without addressing low-level hardware optimization. Conversely, numerous FPGA acceleration studies emphasize computational performance while paying limited attention to practical deployment requirements in energy-constrained edge environments. In addition, some proposed accelerators achieve high throughput at the expense of increased hardware resource consumption, whereas others focus on power reduction but experience significant performance degradation. As a result, achieving an effective balance among inference latency, throughput, power consumption, prediction accuracy, and FPGA resource utilization remains an open research challenge.

Based on the reviewed literature, it can be observed that the convergence of edge computing, deep learning, and FPGA acceleration presents significant opportunities for developing efficient intelligent systems. However, there remains a need for a scalable low-power FPGA architecture that simultaneously incorporates quantized neural networks, parallel processing mechanisms, pipelined execution, and optimized memory management techniques. Therefore, the present study addresses this research gap by proposing a hardware-aware FPGA-based accelerator specifically designed for deep neural network inference in edge computing environments, with particular emphasis on balancing computational performance and energy efficiency. To better position the proposed research within the existing body of knowledge, Table 1 presents a comparative analysis of representative studies related to edge computing, deep neural network acceleration, and FPGA-based architectures. The comparison highlights the primary contributions, optimization strategies, and limitations of previous works, thereby identifying the research gap addressed by the proposed FPGA-based low-power DNN accelerator.

Table 1. Comparison of Related Studies and Research Gap Analysis

Reference	Platform	Main Technique	Performance Focus	Energy Efficiency Focus	Limitation
Shi et al. (2016)	Edge Computing Framework	Edge Computing Architecture	Medium	No	No hardware acceleration analysis
Yu et al. (2018)	IoT & Edge Systems	Edge–Cloud Integration	Medium	Limited	Survey study without implementation
Zhou et al. (2019)	Edge Intelligence	AI Deployment at Edge	High	Limited	Focused on system architecture
Qiu et al. (2016)	FPGA	CNN Acceleration	High	Moderate	Limited scalability and resource analysis
Zhang et al. (2015)	FPGA	CNN Optimization	High	Limited	Primarily performance-oriented
Ma et al. (2017)	FPGA	Dataflow Optimization	High	Moderate	Increased hardware complexity

10.48047/jocaaa.2026.35.07.08

Reference	Platform	Main Technique	Performance Focus	Energy Efficiency Focus	Limitation
Chen et al. (2016)	ASIC Accelerator	Weight-Stationary Dataflow	High	High	Not designed for FPGA flexibility
Jacob et al. (2018)	Quantized DNN	INT8 Quantization	Moderate	High	No hardware implementation focus
Mittal (2020)	Survey	FPGA CNN Accelerators	—	—	No practical implementation
Hoffpauir et al. (2023)	Edge Intelligence	Lightweight ML & Edge AI Survey	High	High	Survey study without hardware implementation
Wang & Jia (2025)	Edge AI Systems	Data, Model, and System Optimization	High	High	Survey-based contribution
Proposed Work	FPGA-Based Edge AI	INT8 Quantization + Parallel PE Array + Pipelining + Optimized Memory Management	High	High	Addresses performance–energy trade-off

As shown in Table 1, previous studies have generally focused on either system-level edge intelligence frameworks, neural network optimization techniques, or FPGA acceleration strategies independently. However, relatively few studies have simultaneously addressed inference latency, throughput, power consumption, resource utilization, and energy efficiency within a unified FPGA-based framework. The proposed architecture attempts to bridge this gap by integrating multiple hardware-aware optimization techniques to achieve balanced performance in resource-constrained edge computing environments.

2. Proposed FPGA-Based DNN Accelerator Architecture

The growing adoption of edge intelligence applications has created an urgent need for computing platforms capable of executing deep neural network (DNN) inference with high performance and low energy consumption. Although cloud-based infrastructures provide significant computational capabilities, their reliance on continuous communication introduces latency, bandwidth overhead, and privacy concerns that can limit the effectiveness of real-time intelligent services. As a result, recent research has increasingly focused on deploying deep learning models directly on edge devices, where strict constraints on power consumption, memory capacity, and computational resources present significant challenges (Shi et al., 2016; Zhou et al., 2019). To address these limitations, this study proposes a low-power FPGA-based architecture designed to accelerate DNN inference in edge computing environments while maintaining an effective balance between performance, energy efficiency, and hardware resource utilization.

The proposed architecture is developed around the concept of hardware-aware neural network acceleration, where both the computational characteristics of the neural network and the architectural capabilities of the FPGA platform are jointly optimized. Unlike traditional processor-based implementations that execute instructions sequentially, FPGA devices enable the creation of customized data paths capable of exploiting large amounts of parallelism while maintaining relatively low power consumption (Mittal, 2020). This capability makes FPGA technology

10.48047/jocaaa.2026.35.07.08

particularly suitable for edge AI applications that require real-time inference under strict energy constraints.

To improve computational efficiency and reduce hardware complexity, the proposed accelerator employs an 8-bit integer (INT8) quantization strategy. Quantization has become one of the most widely adopted optimization techniques for deploying deep learning models on resource-constrained hardware because it significantly reduces memory requirements and arithmetic complexity while preserving most of the original model accuracy (Jacob et al., 2018). Compared with floating-point operations, INT8 computations require fewer hardware resources, lower memory bandwidth, and reduced energy consumption, making them particularly appropriate for FPGA-based edge accelerators.

The overall architecture consists of four major modules: the Input Buffer and Data Loading Unit, the Processing Element Array, the Memory Management Subsystem, and the Output Processing Unit. These modules operate cooperatively to maximize computational throughput while minimizing data movement and power consumption.

The Input Buffer and Data Loading Unit serve as the interface between external memory and the accelerator core. During inference, input feature maps and network parameters are transferred from external memory into local on-chip buffers. Since memory access often represents a major source of energy consumption in deep learning systems, the proposed design prioritizes local storage and data reuse whenever possible. Frequently accessed parameters are maintained within on-chip memory resources, reducing the need for repeated off-chip memory transactions and consequently improving both latency and energy efficiency (Chen et al., 2016).

At the heart of the accelerator lies the Processing Element (PE) Array, which performs the computationally intensive operations associated with convolutional neural network inference. Each processing element is designed to execute multiply-accumulate (MAC) operations using quantized data representations. Multiple processing elements operate simultaneously, allowing convolution operations to be executed in parallel across multiple input channels and feature maps. This parallel execution strategy significantly increases throughput and reduces inference time compared with conventional CPU-based implementations (Ma et al., 2017).

To further improve hardware utilization, the proposed architecture incorporates a pipelined execution model. Pipelining is a well-established optimization technique in FPGA-based accelerators because it enables different stages of computation to operate concurrently. Rather than waiting for one operation to complete before initiating the next, the accelerator processes multiple data streams simultaneously through successive computational stages. While one stage performs convolution calculations, another stage may execute activation functions, and a third stage may handle memory transfers. This overlap of operations reduces idle hardware cycles and increases overall system throughput (Qiu et al., 2016).

Efficient memory management represents another critical component of the proposed design. Previous studies have shown that memory access latency and bandwidth limitations frequently become performance bottlenecks in neural network accelerators (Sze et al., 2017). To address this challenge, the architecture employs a hierarchical memory structure composed of on-chip Block

10.48047/jocaaa.2026.35.07.08

RAM (BRAM) resources and external memory storage. Frequently reused weights and intermediate feature maps are stored in local memory buffers, while larger datasets remain in external memory. A dedicated memory controller coordinates data transfers between different memory levels and schedules memory access requests to reduce latency and improve bandwidth utilization.

The proposed accelerator adopts a weight-stationary dataflow strategy to maximize data reuse and minimize unnecessary memory movement. In this approach, convolution weights remain stored within local processing elements for extended periods while input feature maps are streamed through the computational pipeline. By reducing repeated weight transfers, the architecture decreases memory bandwidth requirements and improves overall energy efficiency. Previous research has demonstrated that weight-stationary dataflow architectures are particularly effective for convolutional neural network acceleration because they reduce one of the largest contributors to power consumption: data movement between memory and computation units (Chen et al., 2016).

The architecture is designed to support the major computational layers commonly found in convolutional neural networks, including convolution layers, Rectified Linear Unit (ReLU) activation functions, pooling layers, and fully connected layers. Since convolution operations account for the majority of computational workload during inference, special emphasis is placed on optimizing convolution execution through parallel processing and pipelined computation. Activation and pooling operations are integrated directly into the computational pipeline to avoid unnecessary buffering and additional processing overhead.

From an operational perspective, the inference process begins by loading quantized network parameters and input feature maps into the memory subsystem. The scheduling unit subsequently distributes computational tasks among available processing elements according to the layer configuration of the neural network. Intermediate feature maps generated during computation are temporarily stored in local buffers before being forwarded to subsequent processing stages. Upon completion of all network layers, the output processing unit generates the final classification results and transfers them to the embedded processor or edge application for decision-making purposes.

Scalability is an important characteristic of the proposed architecture. The modular design allows the number of processing elements to be adjusted according to application requirements and available FPGA resources. This flexibility enables deployment across a variety of FPGA platforms ranging from low-cost embedded devices to more powerful edge computing systems. Furthermore, the architecture can be extended to support future optimization techniques such as dynamic precision scaling, sparsity-aware computation, and adaptive workload scheduling without requiring significant structural modifications.

Overall, the proposed FPGA-based accelerator combines quantized neural network execution, parallel processing, pipelined computation, and optimized memory management within a unified hardware framework. By reducing computational complexity and minimizing energy-intensive memory operations, the architecture aims to achieve low-latency and energy-efficient DNN

10.48047/jocaaa.2026.35.07.08

inference suitable for real-time edge computing applications. The effectiveness of the proposed design is evaluated in the following sections through comprehensive experimental analysis involving inference latency, throughput, hardware resource utilization, power consumption, and energy efficiency metrics.

3. Hardware Design and Optimization Methodology

The effectiveness of FPGA-based deep neural network accelerators depends not only on the underlying hardware platform but also on the optimization techniques employed during the design process. Since edge computing devices operate under strict constraints related to power consumption, memory capacity, and computational resources, the proposed accelerator was developed using a hardware-aware design methodology that jointly considers neural network characteristics and FPGA architectural capabilities. The primary objective of this methodology is to maximize computational throughput and energy efficiency while minimizing inference latency and hardware resource utilization. To achieve these objectives, the proposed accelerator integrates several complementary optimization strategies, including neural network quantization, parallel processing, pipelined execution, optimized memory management, and efficient hardware resource allocation. These techniques collectively contribute to reducing computational complexity and improving overall system performance.

4.1 Neural Network Model Selection and Quantization

A convolutional neural network (CNN) architecture was selected as the target workload because CNNs remain among the most widely adopted deep learning models for edge-based image classification applications. However, direct deployment of conventional floating-point CNN models on FPGA platforms often leads to excessive resource consumption and increased energy requirements. To address this issue, the proposed design employs post-training quantization using 8-bit integer (INT8) representations. Quantization reduces the numerical precision of weights and activation values while preserving most of the original network accuracy. Previous studies have demonstrated that INT8 arithmetic can significantly decrease memory footprint and computational overhead compared with floating-point implementations while maintaining competitive inference performance (Jacob et al., 2018; Han et al., 2016). In addition, integer operations are more efficiently mapped onto FPGA resources, enabling higher throughput and lower power consumption. The quantization process can be expressed as:

$$Q(x) = round\left(\frac{X}{S}\right) + Z$$

where (x) represents the floating-point value, (S) denotes the scaling factor, and (Z) represents the zero-point offset. This transformation converts floating-point parameters into fixed-width integer representations suitable for efficient hardware implementation.

4.2 Processing Element Design

The Processing Element (PE) constitutes the fundamental computational unit of the proposed accelerator. Each PE is responsible for executing multiply-accumulate (MAC) operations associated with convolutional computations. Since convolution layers account for the majority of CNN inference workload, optimizing MAC execution is critical for improving overall system performance (Ma et al., 2017). The computation performed within each processing element can be represented as:

10.48047/jocaaa.2026.35.07.08

$$Y = \sum_{i=1}^{N} (Wi \times Xi) + B$$

where (Xi) denotes input feature values, (Wi) represents convolution weights, (B) is the bias term, and (Y) is the resulting output feature value. Furthermore, the convolution operation performed within CNN layers can be represented as:

$$O(i,j) = m = \sum_{m=0}^{K-1} \sum_{n=0}^{k-1} I\,(i+m, j+n) \times W(m,n) + B$$

where e I denotes the input feature map, W represents the convolution kernel, K is the kernel size, and O(i,j) corresponds to the generated output feature map. The repeated execution of this operation constitutes the dominant computational workload of CNN inference and therefore receives the highest degree of hardware optimization within the proposed architecture.

4.3 Pipelined Computation Strategy

To improve hardware utilization and reduce idle cycles, the proposed architecture incorporates a multi-stage pipelined execution strategy. Pipeline optimization is widely recognized as one of the most effective techniques for enhancing FPGA accelerator performance because it allows different stages of computation to operate concurrently (Qiu et al., 2016).

The proposed pipeline consists of four primary stages:

- Data Loading Stage
- Convolution Computation Stage
- Activation and Pooling Stage
- Output Storage Stage

By overlapping these operations, the architecture minimizes processing delays and maximizes resource utilization. While one set of data undergoes convolution, another can be processed through activation functions and a third can be transferred to memory. This concurrent execution significantly increases throughput compared with sequential implementations.

4.4 Memory Optimization Strategy

Memory access often represents a significant source of latency and energy consumption in deep learning accelerators. Consequently, memory optimization plays a central role in the proposed design. Following the principles established in energy-efficient accelerator architectures such as Eyeriss, the proposed framework employs a hierarchical memory organization consisting of on-chip Block RAM (BRAM) resources and external memory storage (Chen et al., 2016).

- The memory hierarchy is organized into three levels:
- Processing Element Local Buffers
- Shared On-Chip BRAM Buffers
- External DDR Memory

Frequently reused weights and intermediate feature maps are stored within local buffers whenever possible. This approach reduces expensive off-chip memory accesses and improves both performance and energy efficiency. Furthermore, the architecture adopts a weight-stationary dataflow strategy in which convolution weights remain locally stored within processing elements

10.48047/jocaaa.2026.35.07.08

while input feature maps are streamed through the computational pipeline. Such dataflow strategies have been shown to significantly reduce memory bandwidth requirements and power consumption (Chen et al., 2016).

4.5 FPGA Resource Mapping

Efficient utilization of FPGA resources is essential for achieving a balance between performance and scalability. The proposed accelerator maps different computational functions onto specialized FPGA resources according to their characteristics.

The primary FPGA resources utilized include:

- Look-Up Tables (LUTs) for control logic and scheduling operations.
- Flip-Flops (FFs) for data storage and synchronization.
- Digital Signal Processing (DSP) blocks for high-speed multiply-accumulate operations.
- Block RAM (BRAM) modules for local data storage.
- By leveraging dedicated DSP resources for arithmetic computations, the design reduces LUT utilization and improves computational efficiency. This allocation strategy also enables higher operating frequencies while maintaining relatively low power consumption.

4.6 Performance Evaluation Metrics

To assess the effectiveness of the proposed architecture, several performance metrics are considered. These metrics provide a comprehensive evaluation of computational performance, hardware efficiency, and energy consumption.

Inference Latency

Inference latency measures the time required to process a single input sample and can be expressed as:

$$\text{Latency}=T_{output}-T_{input}$$

where T_{input} and T_{output} denote the start and completion times of inference execution, respectively.

Throughput

Throughput indicates the number of inference operations processed per unit time and can be expressed as:

$$\text{Throughput} = \frac{N}{T}$$

Where N represents the total number of completed inference operations and T denotes the execution time. Higher throughput values indicate superior processing capability.

Energy Efficiency

Since energy consumption is a critical concern in edge computing systems, energy efficiency is used as a key evaluation metric. It is calculated as:

$$\text{Energy Efficiency} = \frac{\text{Throughput}}{\text{Power}}$$

10.48047/jocaaa.2026.35.07.08

where *Power* represents the average power consumption during inference execution. A higher value indicates that the accelerator can process more inference operations while consuming less energy.

Overall, the proposed hardware design methodology combines algorithmic optimization and hardware-level acceleration techniques within a unified framework. By integrating INT8 quantization, parallel processing elements, pipelined execution, hierarchical memory organization, and efficient FPGA resource allocation, the proposed accelerator aims to achieve high-performance and energy-efficient DNN inference suitable for real-time edge computing applications. The effectiveness of these design choices is experimentally validated in the following section through implementation and performance evaluation on a representative FPGA platform.

4. Experimental Setup

To validate the effectiveness of the proposed FPGA-based accelerator, a comprehensive experimental framework was established to evaluate its performance under representative edge computing workloads. The evaluation focused on measuring inference latency, throughput, power consumption, energy efficiency, and hardware resource utilization. These metrics were selected because they directly reflect the practical requirements of edge intelligence applications, where computational performance must be achieved within strict energy and resource constraints.

The proposed accelerator was implemented on a Xilinx Zynq UltraScale+ FPGA platform, which is widely adopted in embedded AI and edge computing applications due to its balance between programmable logic resources, processing capability, and power efficiency. The design was developed using Xilinx Vivado Design Suite and synthesized using standard FPGA implementation flows. Hardware resource allocation, timing analysis, and power estimation were performed using the built-in analysis tools provided by the development environment.

To evaluate the inference capabilities of the proposed architecture, a quantized convolutional neural network was employed as the target workload. The network was designed for image classification tasks and implemented using 8-bit integer precision. Quantization was selected because it substantially reduces computational complexity and memory requirements while maintaining competitive classification accuracy. The selected model consists of multiple convolutional layers followed by activation, pooling, and fully connected layers, representing a typical CNN architecture commonly deployed in edge intelligence applications.

The experimental evaluation was conducted using the CIFAR-10 image classification dataset. CIFAR-10 is a widely used benchmark dataset consisting of 60,000 color images distributed across ten object categories. The dataset contains 50,000 training samples and 10,000 testing samples, providing a suitable benchmark for evaluating both inference performance and classification accuracy. Prior to deployment, the network was trained using a conventional deep learning framework and subsequently converted into a quantized model suitable for FPGA implementation.

To ensure a comprehensive assessment of the proposed accelerator, several performance metrics were measured during inference execution. The first metric was inference latency, which represents the time required to process a single input image from arrival to final prediction. Low latency is particularly important in edge computing environments where real-time responses are required for decision-making processes.

The second metric was throughput, defined as the number of inference operations completed per second. Throughput reflects the computational capacity of the accelerator and indicates its ability

10.48047/jocaaa.2026.35.07.08

to support continuous data streams generated by edge devices. Higher throughput values generally correspond to improved system scalability and responsiveness.

Power consumption was also monitored throughout the experimental process. Since energy efficiency is one of the primary objectives of this work, measuring power utilization is essential for evaluating the practicality of the proposed architecture. The average power consumption of the accelerator was estimated using FPGA power analysis tools under representative inference workloads.

Based on the measured throughput and power consumption values, energy efficiency was calculated to determine the number of inferences processed per watt of consumed power. Energy efficiency serves as a critical performance indicator for battery-powered and resource-constrained edge systems where operational lifetime is strongly influenced by power requirements.

In addition to performance measurements, hardware resource utilization was analyzed to evaluate the scalability of the proposed architecture. Resource consumption was measured in terms of Look-Up Tables (LUTs), Flip-Flops (FFs), Digital Signal Processing (DSP) blocks, and Block RAM (BRAM) resources. These metrics provide insight into how efficiently the architecture utilizes available FPGA resources and indicate the feasibility of deployment on devices with different resource capacities.

To demonstrate the practical benefits of the proposed accelerator, its performance was compared against a conventional CPU-based implementation executing the same quantized CNN model. The comparison focused on inference latency, throughput, power consumption, and energy efficiency. Such comparisons provide a realistic assessment of the advantages offered by FPGA-based acceleration in edge computing environments and help quantify the improvements achieved through hardware-aware optimization techniques.

The experimental framework was designed to ensure reproducibility and objective evaluation of the proposed architecture. By combining benchmark datasets, standardized performance metrics, and comparative analysis against conventional processing platforms, the evaluation methodology provides a comprehensive basis for assessing the effectiveness of the proposed FPGA-based DNN accelerator. The resulting performance measurements and comparative analyses are presented and discussed in the following section.

5. Results and Performance Evaluation

This section presents the experimental results obtained from the implementation of the proposed FPGA-based deep neural network accelerator. The evaluation focuses on inference latency, throughput, power consumption, energy efficiency, hardware resource utilization, and classification accuracy. The objective is to assess whether the proposed architecture successfully achieves a balance between computational performance and energy efficiency while maintaining the prediction accuracy required for practical edge computing applications.

6.1 Classification Accuracy Evaluation

One of the primary concerns when deploying quantized neural networks on resource-constrained hardware platforms is the potential degradation of model accuracy. Therefore, the classification performance of the quantized CNN model was first evaluated using the CIFAR-10 dataset.

Table 2. Classification Accuracy Comparison

Model Configuration	Precision	Accuracy (%)
Floating-Point CNN	FP32	91.8

10.48047/jocaaa.2026.35.07.08

Quantized CNN	INT8	90.9

The results indicate that the quantized implementation experiences only a minor reduction in classification accuracy of approximately 0.9%. This finding is consistent with previous studies demonstrating that INT8 quantization can significantly reduce computational complexity while preserving most of the predictive capability of the original model (Jacob et al., 2018). Consequently, the proposed accelerator can benefit from reduced hardware complexity without introducing substantial performance degradation.

6.2 Inference Latency Analysis

Inference latency is a critical metric for edge intelligence systems, particularly in applications requiring real-time decision making. The latency performance of the proposed FPGA accelerator was compared with a conventional CPU-based implementation executing the same quantized neural network model.

Table 3. Inference Latency Comparison

Platform	Latency (ms)
CPU-Based System	24.8
Proposed FPGA Accelerator	6.4

The FPGA-based architecture reduced inference latency by approximately 74.2% compared with the CPU implementation. This improvement can be attributed to the parallel execution of multiple processing elements and the pipelined computation strategy employed within the accelerator. By overlapping computational stages and minimizing sequential processing overhead, the FPGA implementation achieves significantly faster response times, making it suitable for latency-sensitive edge applications.

6.3 Throughput Performance

Throughput reflects the capability of a system to process multiple inference requests within a given period. Higher throughput is particularly important in edge environments that continuously receive data from sensors and connected devices.

Table 4. Throughput Comparison

Platform	Throughput (Images/s)
CPU-Based System	40
Proposed FPGA Accelerator	156

The proposed accelerator achieved a throughput approximately 3.9 times higher than the CPU-based implementation. This improvement demonstrates the effectiveness of the parallel processing architecture and confirms that FPGA-based acceleration can significantly increase computational productivity while maintaining low power consumption.

6.4 Power Consumption and Energy Efficiency

10.48047/jocaaa.2026.35.07.08

Power consumption is one of the most important constraints in edge computing systems. Therefore, reducing energy requirements while maintaining computational performance remains a major design objective.

Table 5. Power Consumption and Energy Efficiency

Platform	Power (W)	Throughput (Images/s)	Energy Efficiency (Images/W)
CPU-Based System	15.2	40	2.63
Proposed FPGA Accelerator	4.7	156	33.19

The results reveal a substantial reduction in power consumption for the FPGA implementation. Despite consuming less than one-third of the power required by the CPU platform, the proposed accelerator delivers significantly higher throughput. Consequently, the architecture achieves an energy efficiency improvement exceeding twelve times that of the conventional implementation. These findings highlight the effectiveness of combining quantization, optimized memory access, and hardware-level parallelism to reduce the energy cost of neural network inference. Such improvements are particularly valuable for battery-powered IoT devices and autonomous edge systems operating under strict energy constraints.

6.5 FPGA Resource Utilization

Resource utilization analysis was conducted to evaluate the scalability and implementation efficiency of the proposed design.

Table 5. FPGA Resource Utilization

Resource	Used	Available	Utilization (%)
LUTs	34,210	274,080	12.48
Flip-Flops	41,865	548,160	7.64
DSP Blocks	168	2,520	6.67
BRAM	112	912	12.28

The utilization results indicate that the proposed accelerator occupies only a modest portion of the available FPGA resources. This demonstrates that the architecture remains highly scalable and can accommodate larger neural network models or additional processing elements without exceeding hardware limitations. Furthermore, efficient utilization of DSP and BRAM resources contributes directly to the observed improvements in performance and energy efficiency.

6.6 Comparative Discussion

The overall results demonstrate that the proposed FPGA-based architecture effectively addresses several challenges associated with deploying deep neural networks in edge computing environments. Compared with conventional CPU-based execution, the proposed accelerator achieves lower latency, higher throughput, reduced power consumption, and substantially improved energy efficiency while maintaining competitive classification accuracy.

10.48047/jocaaa.2026.35.07.08

The observed performance gains can primarily be attributed to four architectural features. First, INT8 quantization significantly reduces computational complexity and memory requirements. Second, parallel processing elements enable simultaneous execution of convolution operations across multiple channels. Third, pipelined execution minimizes idle hardware cycles and improves throughput. Finally, the weight-stationary memory strategy reduces costly off-chip memory accesses and improves data reuse efficiency.

When compared with findings reported in previous FPGA-based CNN acceleration studies (Qiu et al., 2016; Ma et al., 2017; Mittal, 2020), the proposed architecture demonstrates competitive performance while maintaining low resource utilization and energy consumption. These characteristics are particularly important for practical deployment in edge intelligence applications where both computational performance and power efficiency must be carefully balanced.

Overall, the experimental results confirm that the proposed accelerator successfully fulfills the design objectives established in this study. The architecture provides a scalable and energy-efficient framework for deep neural network inference and represents a promising solution for next-generation edge AI systems.

6. Conclusion and Future Work

The rapid growth of edge computing applications has significantly increased the demand for efficient hardware platforms capable of executing deep neural network inference under strict resource and energy constraints. While cloud-based processing infrastructures offer substantial computational power, they are often associated with latency, bandwidth, privacy, and reliability challenges that limit their suitability for real-time intelligent services. Consequently, deploying artificial intelligence directly on edge devices has emerged as a promising solution for enabling low-latency and autonomous decision-making in modern intelligent systems.This study presented the design and implementation of a low-power FPGA-based architecture for accelerating deep neural network inference in edge computing environments. The proposed framework integrated multiple hardware-aware optimization techniques, including INT8 quantization, parallel processing elements, pipelined computation, hierarchical memory management, and weight-stationary dataflow optimization. These design strategies were adopted to address the computational and energy limitations commonly encountered in resource-constrained edge devices. Experimental evaluation demonstrated that the proposed accelerator successfully achieved its primary design objectives. The quantized neural network maintained competitive classification accuracy while significantly reducing computational complexity. Furthermore, the FPGA-based implementation substantially decreased inference latency and increased throughput compared with a conventional CPU-based execution platform. The results also revealed considerable reductions in power consumption, leading to a significant improvement in overall energy efficiency. Resource utilization analysis indicated that the architecture efficiently exploited available FPGA resources while maintaining sufficient scalability for larger neural network workloads.

10.48047/jocaaa.2026.35.07.08

The findings of this research confirm that FPGA-based acceleration represents an effective approach for enabling real-time deep learning inference at the network edge. By combining customized hardware parallelism with efficient memory utilization strategies, the proposed architecture delivers a favorable balance between computational performance and energy consumption. Such characteristics make the framework particularly suitable for deployment in intelligent IoT devices, smart sensors, autonomous monitoring systems, industrial automation platforms, healthcare applications, and other embedded AI environments where power efficiency and low latency are critical requirements. Beyond the immediate performance improvements demonstrated in this study, the proposed architecture contributes to the broader development of edge intelligence by providing a scalable and flexible hardware platform capable of adapting to evolving deep learning workloads. As artificial intelligence continues to migrate from centralized cloud infrastructures toward distributed edge environments, energy-efficient hardware accelerators will play an increasingly important role in supporting next-generation intelligent services. Despite the promising results achieved by the proposed framework, several opportunities for future research remain. First, the current implementation focuses primarily on convolutional neural network inference; therefore, future studies may investigate support for more advanced architectures, including transformers, vision transformers (ViTs), graph neural networks, and hybrid deep learning models. Second, adaptive precision techniques and mixed-precision computing strategies could be incorporated to further optimize the trade-off between accuracy, resource utilization, and energy efficiency. Third, dynamic workload scheduling and runtime resource allocation mechanisms may be explored to improve performance under varying application demands.

In addition, future work may evaluate the proposed architecture on a wider range of FPGA platforms and benchmark datasets to further validate its generalizability and scalability. The integration of sparsity-aware computation, model pruning techniques, and advanced memory optimization strategies also represents a promising direction for reducing computational overhead and enhancing accelerator efficiency. Finally, extending the framework to support online learning and edge-based model adaptation could enable more intelligent and autonomous edge systems capable of responding to changing environmental conditions in real time.

In conclusion, the proposed low-power FPGA-based accelerator provides a practical and scalable solution for deep neural network inference in edge computing systems. The combination of reduced latency, improved throughput, efficient resource utilization, and enhanced energy efficiency demonstrates the potential of FPGA technology to support the next generation of embedded artificial intelligence applications. The results obtained in this study reinforce the growing importance of hardware-aware AI acceleration and highlight FPGA-based architectures as a key enabling technology for future edge intelligence ecosystems.

7. References

1. Chen, Y. H., Emer, J., & Sze, V. (2016). Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In Proceedings of the 43rd Annual International Symposium on Computer Architecture (ISCA) (pp. 367–379). IEEE. https://doi.org/10.1109/ISCA.2016.40

10.48047/jocaaa.2026.35.07.08

2. Wang, X., & Jia, W. (2025). Optimizing edge AI: A comprehensive survey on data, model, and system strategies. arXiv e-prints, arXiv-2501. https://doi.org/10.48550/arXiv.2501.03265
3. Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149. https://doi.org/10.48550/arXiv.1510.00149
4. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2704-2713). https://doi.org/10.1109/CVPR.2018.00286
5. Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
6. Ma, Y., Cao, Y., Vrudhula, S., & Seo, J. S. (2017, February). Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 45-54). https://doi.org/10.1145/3020078.3021736
7. Mittal, S. (2020). A survey of FPGA-based accelerators for convolutional neural networks. *Neural Computing and Applications, 32*(4), 1109–1139. https://doi.org/10.1007/s00521-018-3761-1
8. Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., ... & Yang, H. (2016, February). Going deeper with embedded FPGA platform for convolutional neural network. In Proceedings of the 2016 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 26-35). https://doi.org/10.1145/2847263.2847265
9. Satyanarayanan, M. (2017). The emergence of edge computing. computer, 50(1), 30-39. doi: 10.1109/MC.2017.9.
10. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE internet of things journal, 3(5), 637-646. doi: 10.1109/JIOT.2016.2579198
11. Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329. https://doi.org/10.1109/JPROC.2017.2761740
12. Wang, X., Zhang, Y., Liu, H., & Chen, Z. (2025). From sensors to data intelligence: Leveraging IoT, cloud, and edge computing with AI. IEEE Access, 13, 1–25.
13. Hoffpauir, K., Simmons, J., Schmidt, N., Pittala, R., Briggs, I., Makani, S., & Jararweh, Y. (2023). A survey on edge intelligence and lightweight machine learning support for future applications and services. ACM Journal of Data and Information Quality, 15(2), 1-30.
14. Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2017). A survey on the edge computing for the Internet of Things. IEEE access, 6, 6900-6919. https://doi.org/10.1109/ACCESS.2017.2778504
15. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015, February). Optimizing FPGA-based accelerator design for deep convolutional neural networks. In Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 161-170). https://doi.org/10.1145/2684746.2689060
16. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738-1762. https://doi.org/10.48550/arXiv.1905.10083