ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

FPGA-Based Real-Time Hand Gesture Recognition Using a Lightweight CNN on PYNQ-Z2

Surya Thummalapeta[1]; Dr. Narender Reddy Kampelli[2]

[1]Department of Electronics and Communication Engineering Jawaharlal Nehru Technological University, Hyderabad Hyerabad, India
[2]Department of Electronics and Communication Engineering Jawaharlal Nehru Technological University, Hyderabad Hyderabad, India

Publication Date: 2026/01/20

Abstract: Hand gesture recognition enables natural interaction between humans and machines and is widely used in vision- based embedded applications. Although convolutional neural networks provide strong recognition capability, their deployment on resource-constrained platforms presents challenges related to computation, latency, and system integration. This paper presents the design and implementation of a real-time hand gesture recognition system using a lightweight CNN accelerator deployed on a PYNQ-Z2 FPGA platform. The proposed system adopts a hardware–software co-design approach, where image acquisition and control are handled by the processing system, while CNN inference is offloaded to programmable logic for acceleration. A compact CNN architecture based on depthwise separable convolutions is employed to reduce computational complexity and resource usage. The system supports live camera input, real-time inference, and web-based visualization. Experimental observations demonstrate the feasibility of deploying CNN-based hand gesture recognition on low-cost FPGA platforms, highlighting practical design trade-offs and implementation considerations.

Keywords: *Hand Gesture Recognition, FPGA Acceleration, Convolutional Neural Networks, PYNQ-Z2, Hardware–Software Co-Design.*

How to Cite: Surya Thummalapeta; Dr. Narender Reddy Kampelli (2026) FPGA-Based Real-Time Hand Gesture Recognition Using a Lightweight CNN on PYNQ-Z2. *International Journal of Innovative Science and Research Technology*, 11(1), 1188-1196. https://doi.org/10.38124/ijisrt/26jan329

I. INTRODUCTION

Hand gesture recognition (HGR) enables natural and touch- less interaction between humans and machines and has gained attention in domains such as sign language interpretation, virtual interfaces, and robotics. Vision-based approaches using deep learning have demonstrated strong capability in learning discriminative gesture representations from images. However, deploying CNN-based models on embedded systems remains challenging due to limited computational resources, power constraints, and latency requirements.

Field-programmable gate arrays (FPGAs) provide a promis- ing solution by enabling parallel computation and hardware customization while maintaining low power consumption. Recent advances in FPGA-based system-on-chip platforms, such as Xilinx Zynq devices, allow tight integration of pro- grammable logic (PL) with embedded processors, facilitating hardware–software co-design.

This work presents an FPGA-based real-time hand gesture recognition system implemented on the PYNQ-Z2 platform. A lightweight CNN model is designed and trained offline, then synthesized into a custom accelerator using high-level synthesis (HLS). The accelerator is integrated into a complete system supporting live video capture, inference, and visualiza- tion. Recent advances in deep learning and embedded systems have enabled real-time vision-based gesture recognition across a wide range of platforms, from desktop GPUs to low-power embedded devices [6], [19].

Rather than focusing on achieving state-of-the-art accu- racy, this paper emphasizes practical system design, deploy- ment methodology, and real-time operation on a resource- constrained FPGA platform. Accordingly, this work prioritizes system-level feasibility, deterministic runtime behavior, and hardware-aware architectural design over purely accuracy- driven optimization. Accordingly, this work prioritizes system- level feasibility, deterministic runtime behavior, and hardware- aware architectural design

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

over purely accuracy-driven opti- mization.

➢ *The Main Contributions of this Work are:*

- Design of a compact CNN architecture suitable for FPGA deployment.
- Implementation of a CNN accelerator using HLS and AXI-based data movement.
- End-to-end real-time hand gesture recognition system on PYNQ-Z2 with live visualization.

II. RELATED WORK

➢ *Vision-Based Hand Gesture Recognition*

Early vision-based hand gesture recognition systems relied on hand-crafted features, color segmentation, and contour- based methods. While effective under controlled conditions, such approaches are sensitive to illumination changes, back- ground clutter, and hand shape variations. Earlier vision- based hand gesture recognition systems relied on handcrafted features such as contour descriptors, skin color segmentation, and motion cues [1], [2]. These methods were sensitive to lighting conditions, background clutter, and camera viewpoint variations [3].

Several works explored hand segmentation techniques to isolate the hand region prior to recognition, but these pipelines often require extensive preprocessing and parameter tuning. Although these methods demonstrate promising results in con- trolled environments, their dependence on handcrafted features and segmentation heuristics limits robustness under varying illumination and background conditions. These limitations motivate the adoption of learning-based approaches that can generalize across diverse scenarios.

➢ *CNN-Based Gesture Recognition*

Convolutional neural networks have become the dominant approach for hand gesture recognition due to their ability to learn hierarchical features directly from image data. CNN- based models have been applied to both static and dynamic gesture recognition tasks, achieving robust performance across varying conditions. However, deep CNN architectures typi- cally require significant computational resources, making di- rect deployment on embedded platforms challenging without optimization. Convolutional neural networks have demon- strated superior robustness and accuracy for hand gesture recognition tasks by learning hierarchical feature representa- tions directly from raw images [4], [5]. Depthwise separable convolutions further reduce model complexity while preserv- ing classification performance [6], [7]. Recent CNN-based approaches have explored both static and dynamic gesture recognition, including the use of temporal modeling and attention mechanisms.While such models improve recogni- tion accuracy, their increased computational complexity poses challenges for deployment on resource-constrained embedded systems, necessitating architectural simplification and hard- ware acceleration. Although CNN-based gesture recognition approaches achieve strong classification performance, many reported models are designed primarily for execution on GPUs or high-performance processors. When deployed on resource- constrained embedded platforms, such architectures often fail to meet real-time latency or power requirements. This limita- tion motivates the exploration of compact CNN designs that explicitly account for hardware constraints, as adopted in this work.

➢ *Dynamic and Temporal Gesture Recognition*

Beyond static hand gesture recognition, several studies have explored dynamic gesture recognition using temporal modeling techniques. Approaches based on recurrent neural networks, temporal convolution, and attention mechanisms have been proposed to capture motion information across frames [8], [9]. While these methods improve recognition of continuous gestures, they significantly increase model com- plexity and memory requirements [10].. As a result, such architectures are often unsuitable for deployment on resource- constrained embedded platforms without aggressive optimiza- tion or hardware acceleration.

➢ *FPGA-Based CNN Acceleration*

To address computational limitations, several studies have explored FPGA-based acceleration of CNN inference [11], [12].. Techniques such as model quantization, loop pipelining, and depthwise separable convolutions have been proposed to reduce hardware complexity. FPGA-based gesture recognition systems have demonstrated improved performance and en- ergy efficiency compared to software-only implementations.

Table 1 Qualitative Comparison of Hand Gesture Recognition Approaches

Work	Platform	Model Type
Vision-based methods	CPU	Hand-crafted features
CNN-based methods	CPU/GPU	Deep CNN
Dynamic gesture models	GPU	RNN / Attention
FPGA CNN accelerators	FPGA	Optimized CNN
Proposed System	**PYNQ-Z2 FPGA**	**Lightweight CNN**

Nonetheless, many existing works focus primarily on accel- erator design, with limited discussion of complete end-to- end deployment on embedded platforms. Depthwise separable convolution has emerged as an effective strategy for reducing computational cost in FPGA-based CNN accelerators [13], [14]. By decoupling spatial and channel-wise convolution, these architectures significantly reduce parameter count and arithmetic operations, making them suitable for low-cost FPGA platforms. Building on these ideas, the proposed system integrates a lightweight CNN accelerator into a complete real- time hand gesture recognition pipeline, emphasizing end-to- end deployment

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

on a low-cost FPGA platform rather than isolated accelerator evaluation.

➢ *Embedded Real-Time Gesture Recognition Systems*

Several real-time hand gesture recognition systems have been proposed for embedded platforms, focusing on latency, stability, and power efficiency. FPGA-based systems, in par- ticular, offer deterministic performance and parallel execution, making them suitable for real-time vision tasks. However, many existing implementations rely on high-end FPGA de- vices or simplified pipelines that exclude live visualization and user interaction. In contrast, the proposed work demonstrates a complete end-to-end system deployed on a low-cost FPGA platform, integrating live camera input, hardware-accelerated inference, and user-facing visualization.

➢ *Embedded and Real-Time Gesture Recognition Systems*

Several studies have investigated real-time hand gesture recognition systems targeting embedded platforms. These works typically emphasize latency reduction, system stabil- ity, and deployment feasibility rather than peak classification accuracy. FPGA-based and system-on-chip implementations have been shown to offer predictable performance and energy efficiency for real-time vision applications. However, many existing systems rely on either high-end FPGA devices or simplified pipelines that omit end-to-end integration aspects such as live visualization and user interaction. The proposed work addresses this gap by demonstrating a complete real- time system on a low-cost FPGA platform with integrated software and hardware components. The PYNQ platform en- ables rapid prototyping of hardware-accelerated vision systems by integrating Python-based control with FPGA acceleration [15], [16]. This approach simplifies system development while maintaining real-time performance.

III. SYSTEM OVERVIEW

➢ *End-to-End Processing Pipeline*

The proposed system follows a modular pipeline consist- ing of image acquisition, preprocessing, hardware-accelerated inference, and visualization. Live frames are captured from a USB camera connected to the processing system (PS). The captured images are resized and formatted into fixed-size RGB tensors before being transferred to the programmable logic (PL) using an AXI Direct Memory Access (DMA) interface. The CNN accelerator processes the input tensor and returns classification scores to the PS for postprocessing and visualization. The overall pipeline was refined iteratively based on observed data movement overheads, synchronization constraints between the processing system and programmable logic, and real-time visualization requirements.

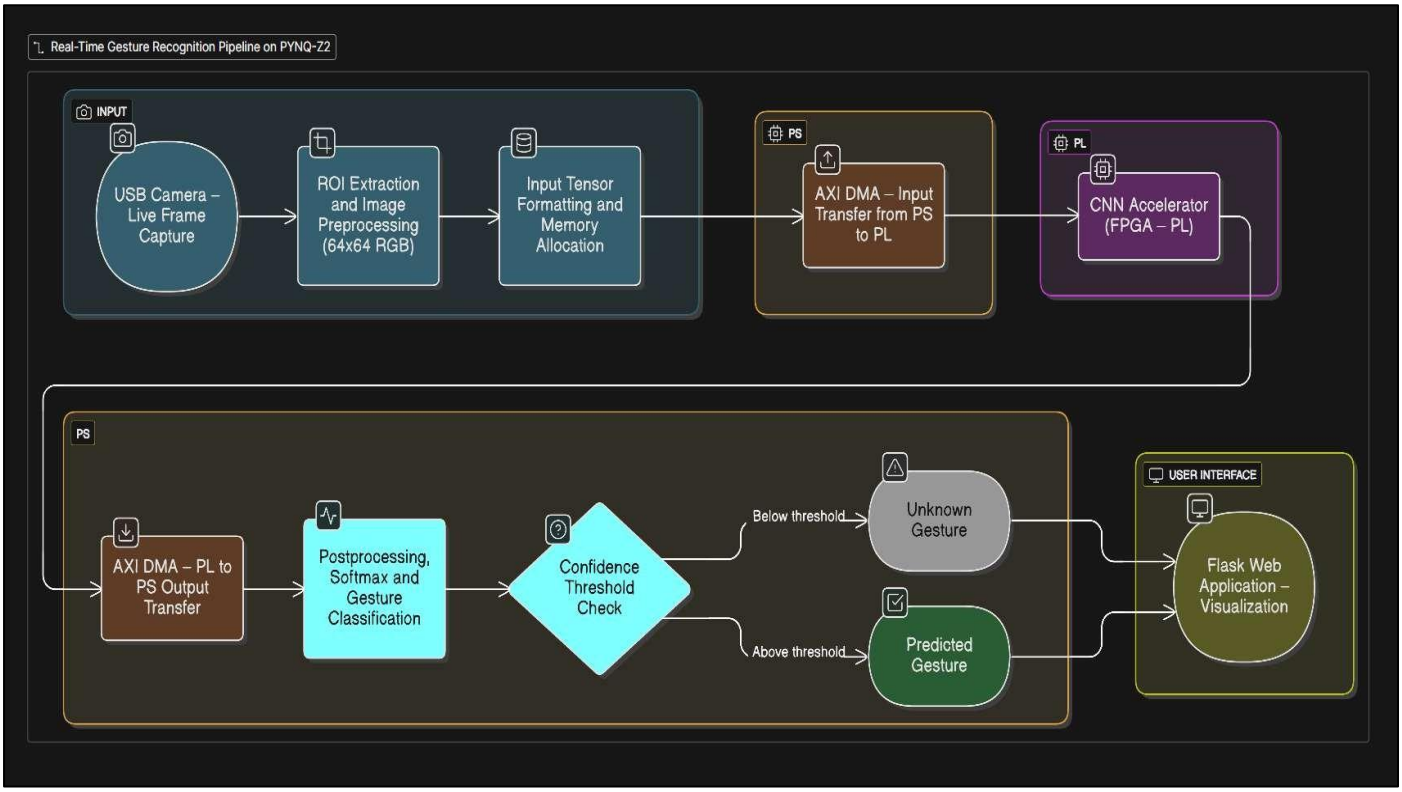

Fig 1 Overall Real-Time Hand Gesture Recognition Pipeline on the PYNQ-Z2 Platform.

➢ *Hardware–Software Co-Design Strategy*

A hardware–software co-design approach is adopted to balance flexibility and performance. The PS handles tasks that benefit from software control, such as camera interfacing, preprocessing, thresholding, and user interface management. The computationally intensive CNN inference is offloaded to the PL, where parallelism and pipelining can be exploited to reduce latency.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

IV. CNN MODEL DESIGN AND TRAINING

➢ *Dataset Description and Preprocessing*

The CNN model is trained using a combination of publicly available hand gesture datasets and curated background im- ages. Hand gesture samples are sourced from the OUHANDS dataset, which provides diverse hand pose images suitable for recognition tasks. Additional background images are included to represent non-gesture scenarios. All images are resized to 64×64 RGB format to ensure uniform input dimensions. Hand gesture samples are sourced from the OUHANDS dataset, which provides diverse hand poses suitable for detection and recognition tasks [17]. Additional background images are incorporated to improve robustness against non-gesture inputs [18].

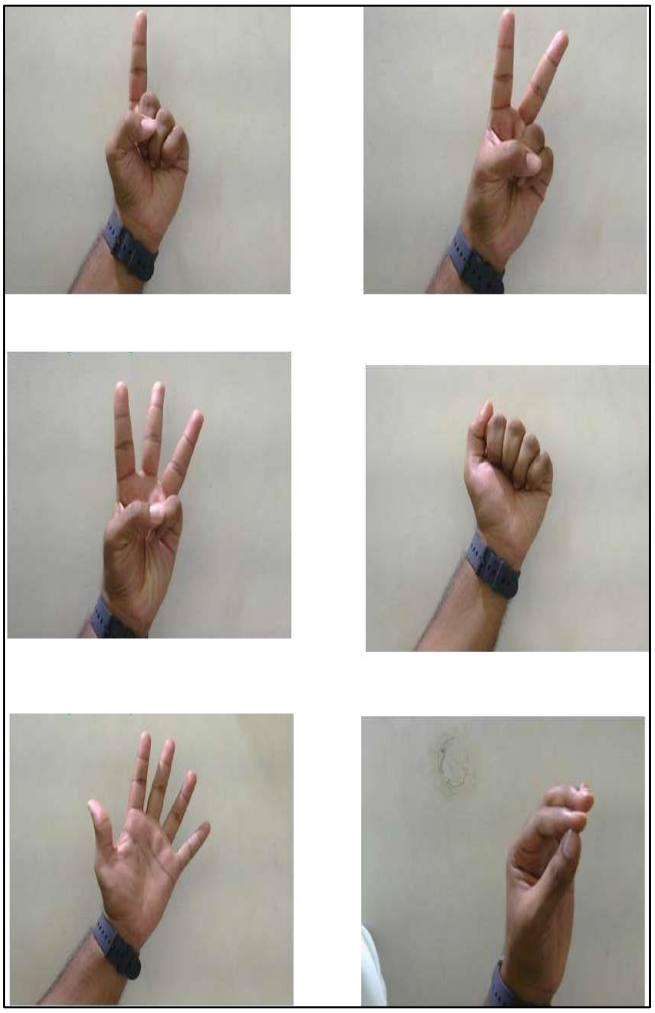

Fig 2 Sample Hand Gesture Images from the Dataset.

➢ *Lightweight CNN Architecture*

To enable efficient FPGA deployment, a compact CNN architecture is employed. The model uses depthwise sepa- rable convolutions to reduce the number of parameters and arithmetic operations. The network consists of multiple con- volutional blocks followed by pooling layers and a small fully connected classifier. This design significantly reduces computational complexity compared to standard convolutional architectures. The architectural design prioritizes a balance between representational capacity and hardware efficiency. Depthwise separable convolutions are selected to minimize parameter count while preserving spatial feature extraction ca- pability. Pooling layers progressively reduce spatial resolution, limiting memory bandwidth requirements in later layers. This design enables efficient mapping to FPGA hardware while maintaining sufficient discriminative power for static hand gesture recognition.

➢ *Training Strategy*

The CNN is trained offline using supervised learning with categorical cross-entropy loss. Training and validation splits are used to monitor convergence and prevent overfitting. After training, model weights are exported for hardware synthesis. While floating-point training is performed for convenience, the architecture is designed with quantization compatibility in mind.

V. FPGA IMPLEMENTATION AND INTEGRATION

➢ *CNN Accelerator Design*

The trained CNN model is translated into a hardware accelerator using Vitis HLS. Each layer is implemented as a sequence of pipelined operations, with careful consideration of memory access patterns and data reuse. The accelerator exposes AXI interfaces for control and streaming data transfer.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

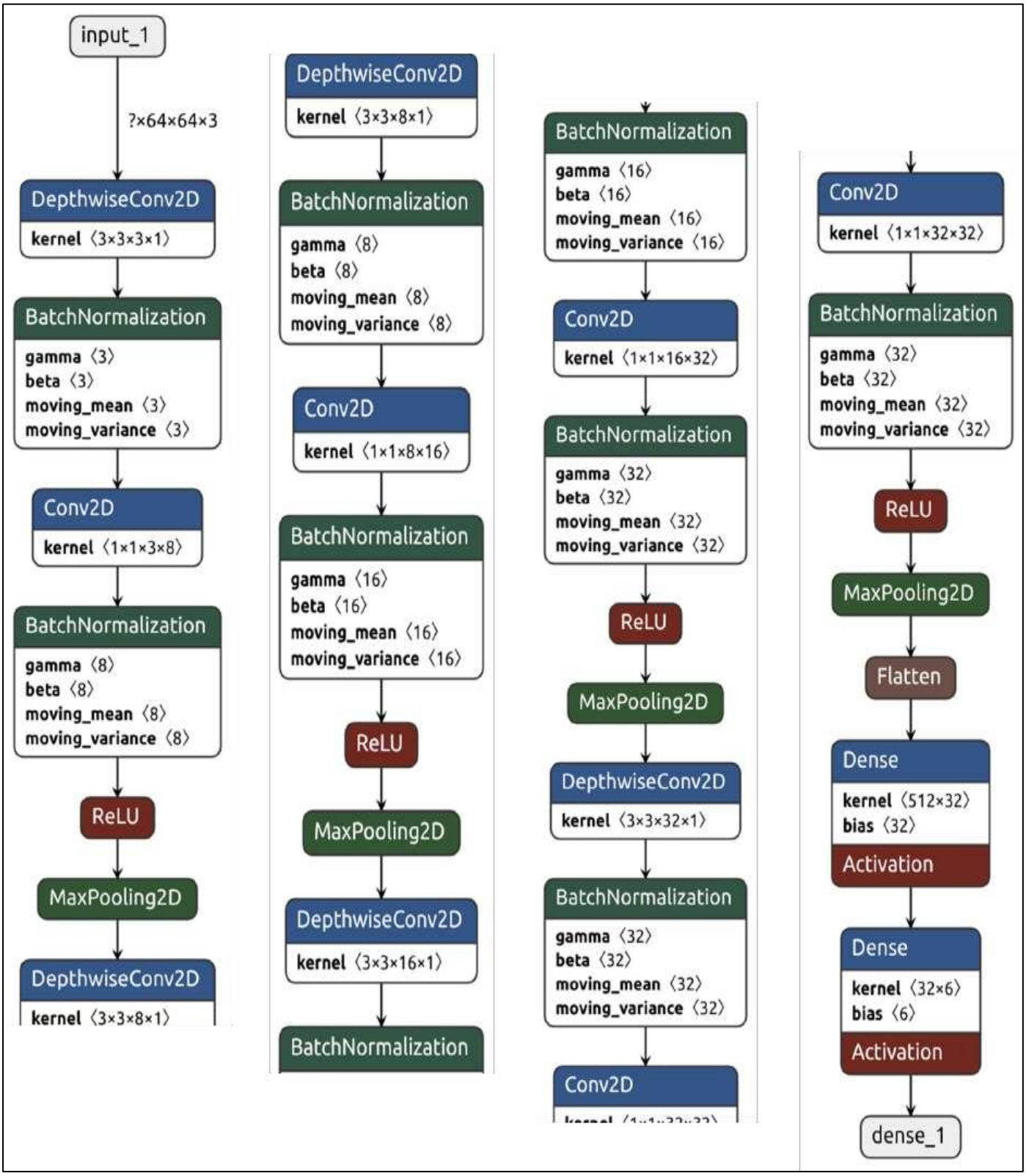

Fig 3 Lightweight CNN Architecture Based on Depth Wise Separable Convolutions.

➢ *AXI-Based Data Movement*

AXI DMA is used to transfer input tensors from PS memory to the CNN accelerator and to retrieve output results. FIFO buffers are employed to decouple data streams and improve throughput. This approach allows efficient communication between software and hardware components.

➢ *Vivado Block Design and Deployment*

The accelerator is integrated into a Vivado block design alongside the Zynq processing system, DMA engine, and in- terconnects. After synthesis and implementation, the generated bitstream and hardware description files are deployed on the PYNQ-Z2 board.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

➢ *Hardware Resource Utilization*

The FPGA resource utilization of the CNN accelerator is obtained from Vitis HLS synthesis reports. The results indicate that the lightweight CNN architecture fits comfortably within the available resources of the PYNQ-Z2 platform.

Table 2 FPGA Resource Utilization of the CNN Accelerator

Resource	Utilization	Capacity	%Utilization
LUTs	42,609	53,000	80.39
Flip-Flops	21,461	106,000	20.25
DSP Blocks	140	220	63.63
BRAM	85	140	60.71

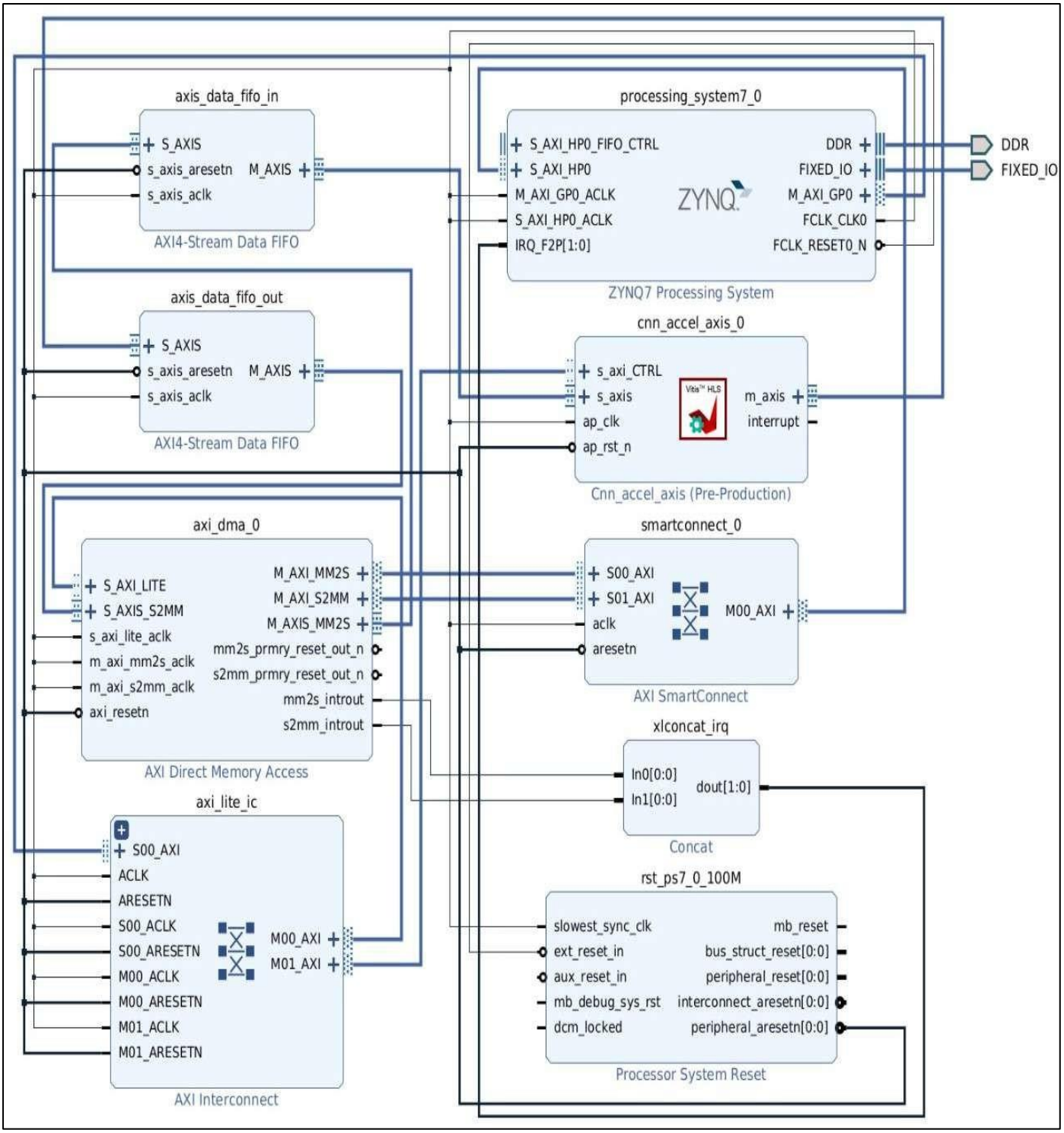

Fig 4 Vivado Block Design of the FPGA System

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

➢ *Design Trade-Offs and Implementation Considerations*

The FPGA implementation involves trade-offs between latency, resource utilization, and design complexity. Loop pipelining and data reuse are employed to improve throughput, while FIFO buffering decouples data transfer from compu- tation. Although aggressive optimization can further reduce latency, the current design prioritizes stability and functional correctness, ensuring reliable real-time operation on a low-cost FPGA platform.

VI. EXPERIMENTAL SETUP AND OBSERVATIONS

➢ *Live Inference Setup*

The system is evaluated using live video input from a USB camera. The PS executes a Python-based control application that manages frame capture, data transfer, and result display. A web-based dashboard is implemented using a lightweight server framework to visualize predictions in real time.

➢ *Runtime Observations*

The system achieves real-time inference with consistent responsiveness, as supported by the measured runtime perfor- mance metrics. The inclusion of an explicit unknown gesture class helps handle ambiguous inputs. Observed misclassifi- cations highlight the trade-off between model compactness and classification robustness, emphasizing the importance of dataset diversity and model tuning.

➢ *Runtime Performance Metrics*

To evaluate real-time feasibility, runtime performance met- rics are recorded during live system execution. These metrics include frames per second (FPS), CNN inference latency, and end-to-end (E2E) system latency, measured from frame acquisition to final visualization output. Runtime statistics are logged during continuous operation of the system.

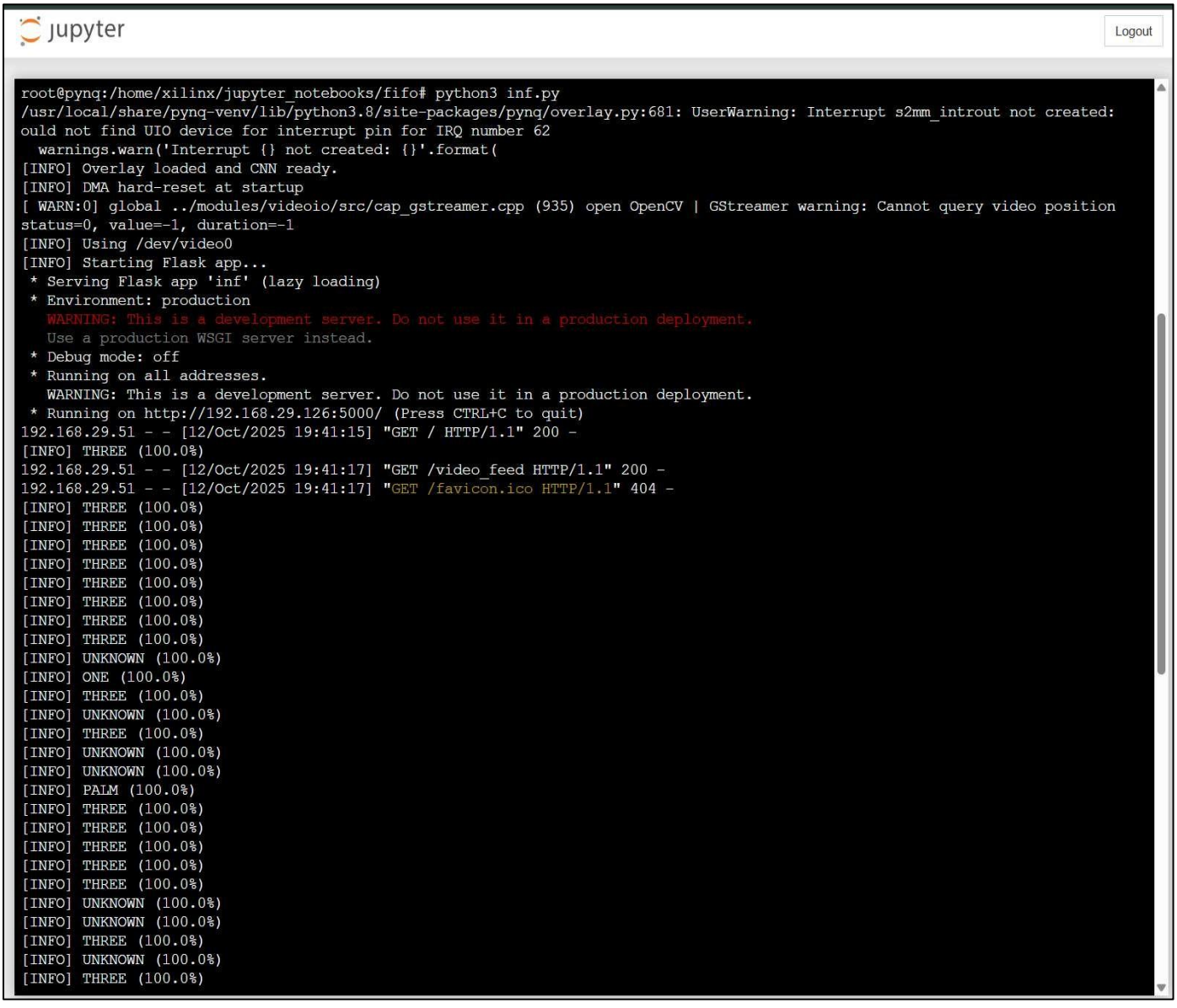

Fig 5 Real-Time Gesture Recognition Dashboard.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

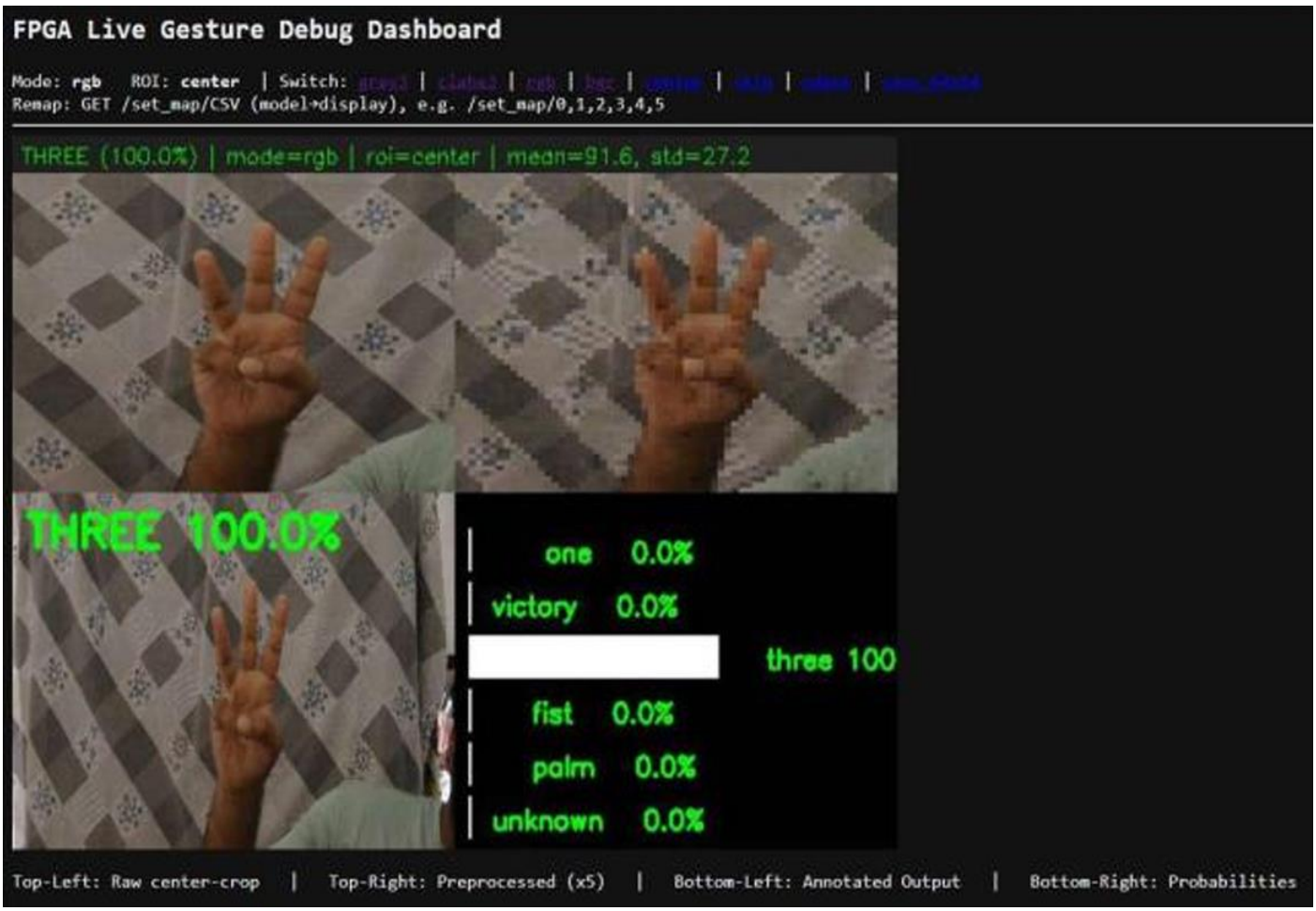

Fig 6 Real-Time Gesture Recognition Webpage Displaying Live Predictions.

VII. DISCUSSION

The presented system demonstrates the practicality of de- ploying CNN-based hand gesture recognition on low-cost FPGA platforms. While deeper models may offer higher accuracy, the chosen lightweight architecture enables real-time operation within resource constraints. The hardware–software co-design approach provides flexibility for future enhance- ments, such as model refinement or additional preprocessing.

VIII. CONCLUSION AND FUTURE WORK

This paper presented a real-time hand gesture recognition system implemented on a PYNQ-Z2 FPGA platform using a lightweight CNN accelerator. The system integrates live image acquisition, hardware-accelerated inference, and web- based visualization. Experimental observations confirm the feasibility of deploying CNN-based gesture recognition on embedded FPGA platforms. Future work includes improving model robustness, exploring quantized inference, and extend- ing the system to dynamic gesture recognition.

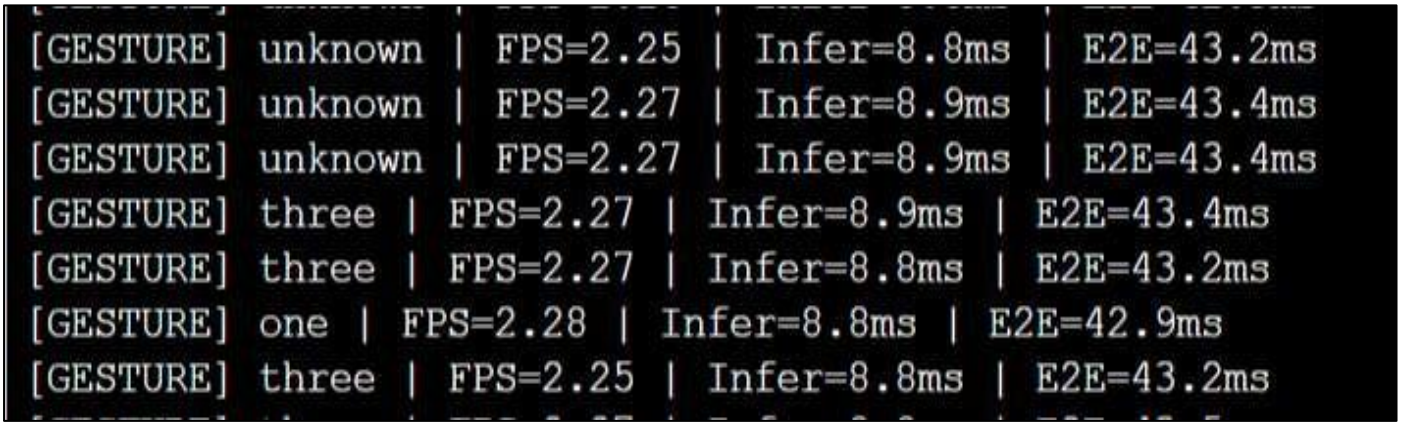

Fig 7 Console Output Showing Runtime Performance Metrics During Live Inference.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jan329

Table 3 Runtime Performance Metrics of the Proposed System

Metric	Observed Value
Frames Per Second (FPS)	2.25 – 2.28
Inference Latency (ms)	8.8 – 8.9
End-to-End Latency (ms)	42.9 – 43.4

REFERENCES

[1]. S. Rautaray and A. Agrawal, "Hand gesture recognition for human- computer interaction: A survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[2]. J. S. Sun, T. J. Zhang, J. Yang, and G. Ji, "Research on hand gesture recognition based on deep learning," in *Proc. 12th Int. Symp. Antennas, Propagation and EM Theory*, 2018, pp. 1–4.

[3]. G. Plouffe and A. Cretu, "Static and dynamic hand gesture recognition in depth data using time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, 2016.

[4]. A. Kumar, S. Verma, and R. Agarwal, "A pattern recognition model for hand gestures recognition using convolutional neural networks," *Procedia Computer Science*, vol. 167, pp. 133–142, 2020.

[5]. S. Mumtaz et al., "FPGA implementation of a convolutional neural network for classification," *IEEE Access*, vol. 8, pp. 74941–74950, 2020.

[6]. F. Chollet, "Xception: Deep learning with depthwise separable convolu- tions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[7]. L. Bai, Y. Zhao, and X. Zhang, "A CNN accelerator on FPGA using depthwise separable convolution," *IEEE Trans. Circuits Syst. II*, vol. 65, no. 10, pp. 1415–1419, Oct. 2018.

[8]. M. Mitalainen, S. Pangi, J. Holappa, and O. Silven, "Dynamic hand gesture recognition using effective feature extraction and attention-based deep neural networks," *IEEE Access*, vol. 8, pp. 110120–110130, 2020.

[9]. P. K. Pisharady and A. P. L. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 403–419, 2013.

[10]. W. Zhang, J. Wang, and L. Fan, "Dynamic hand gesture recognition based on short-term sampling neural networks," *IEEE/CAA J. Automat- ica Sinica*, vol. 8, no. 1, pp. 110–120, 2021.

[11]. C. Zhang et al., "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA Int. Symp. FPGA*, 2015, pp. 161–170.

[12]. K. Guo et al., "Angel-Eye: A complete design flow for mapping CNN onto embedded FPGA," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 1, pp. 35–47, 2018.

[13]. L. Bai and X. Huang, "High-speed low-cost CNN inference accelerator for depthwise separable convolution," *IEEE Trans. Circuits Syst. II*, 2019.

[14]. P. Barros et al., "A multimodal convolutional neural network for hand posture recognition," in *Proc. Int. Conf. Neural Networks*, 2014.

[15]. V. C. Johnson et al., "FPGA-based hardware acceleration using PYNQ- Z2," in *Proc. IEEE ICEEICT*, 2023.

[16]. A. Ghoward, M. Zhu, B. Chen, D. Kalenichenko, and W. Wang, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[17]. M. Mitalainen, S. Pangi, J. Holappa, and O. Silven, "OUHANDS database for hand detection and pose recognition," in *Proc. Int. Conf. Image Process. Theory, Tools and Applications*, 2016.

[18]. M. Everingham et al., "The Pascal visual object classes (VOC) chal- lenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[19]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[20]. T.-H. Tsai, Y.-C. Hsu, and C.-J. Wu, "Hardware architecture design for hand gesture recognition system on FPGA," *IEEE Access*, vol. 11, pp. 24567–24578, 2023.