Design and Implementation of resource-efficient light-weight CNN architecture

Harish J Dr. K. R. Sarathchandran

Department of CSE, Sri Sivasubramaniya Nadar college of Engineering

Highlights of Proposed Model

- Designed a profile-based reconfigurable CNN accelerator using Vitis HLS 2023.1, synthesized in Vivado.
- Enabled dynamic profile selection based on input size (e.g., 28×28, 32×32).
- Supported real-time CNN reconfiguration via AXI-stream, reducing logic redundancy.
- Achieved efficient resource usage (LUTs, DSPs, BRAM) through shared logic across profiles.

Challenges in Implementing CNNs for H/W Synthesis

- Mapping 2D convolutions to 1D streams suitable for FPGA processing.
- Adapting weights and biases dynamically without hardware re-synthesis.

Proposed Model for reconfigurable CNN

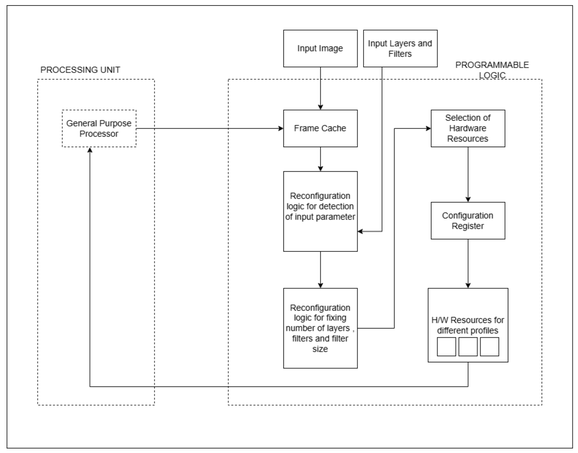

Figure 1. Proposed model Overview

Performance metrics of the wild animal detection

Resource	Profile 1	Profile 2	Full System
LUT	24032	29007	33908
FF	27716	35621	32299
DSP	158	127	127
BRAM	70	58	184
SRL	412	373	2014
LUT Utilization	10.99%	13.27%	15.51%
DSP Utilization	17.56%	14.11%	14.11%
RAM Utilization	12.84%	10.64%	33.76%

Table 1. Comparison of Resource Utilization

Functional Modules and Dataset Description

- **Image Analysis & Configuration**
  - Input image acquisition and frame caching
  - Parameter extraction (resolution, size, channels)
  - Profile selection logic based on input metadata
- **CNN Configuration & Reconfiguration**
  - Dynamic configuration of layers, kernels, and pooling
  - Activation of hardware resources from configuration profiles
  - Register-based control for profile switching
- **CNN Processing**
  - Convolutional operations using streamed weights and image data
  - Max pooling and activation layer operations
  - Fully connected layer processing for classification
- **System Integration & Control**
  - AXI4-Stream interfacing for image, weight, and output data
  - Processing System (PS) coordination and control (Zynq SoC)
  - Synchronization, reset, and dataflow management

- The MNIST dataset consists of grayscale images of handwritten digits (0–9), each with a size of 28x28 pixels, totaling 60,000 samples.
- It is used for digit recognition tasks and serves as the baseline for CNN profiling in Profile 1 configuration.
- For Profile 2, the images were upscaled to 32x32 pixels and trained with this modified resolution.

Functional pipeline of Resource-Efficient Light-Weight CNN Architecture

Dynamic Profile-Based CNN Configuration

- The system starts by analyzing the metadata of the input image (such as image size, resolution) to determine the most suitable CNN configuration.
- Based on the image characteristics, a profile is selected—Profile 1 for 28×28 images and Profile 2 for 32×32 images. Each profile has predefined convolutional layers, kernel sizes, and dense layers.
- This profiling enables dynamic reconfiguration of the CNN model at runtime without requiring multiple static hardware instances.
- Ensures resource efficiency and supports adaptive execution for varying datasets on FPGA using reconfigurable logic blocks.

Hardware-Aware Deployment Using Vitis HLS and Vivado

- The selected profile is compiled into an RTL-compatible block using Vitis HLS and integrated via Vivado Design Suite.
- Data flows through AXI4-Stream interfaces from input (image) to weight loaders, into CNN compute blocks, and finally to output streams.
- The FPGA performs convolution, pooling, activation, and fully connected operations using reconfigurable blocks controlled by a configuration register.
- A Zynq-7000 SoC integrates programmable logic with a general-purpose processor, ensuring low-latency, high-throughput inference on edge devices.

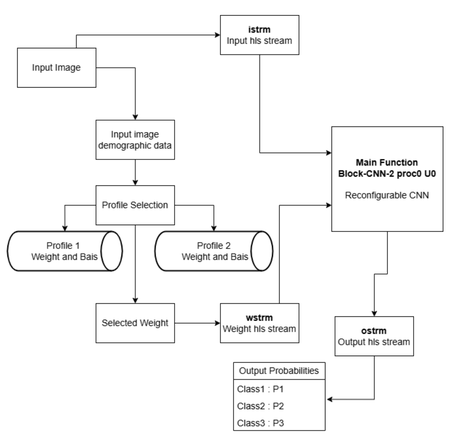

Figure 2. Vitis Hls pipeline

CNN-based Digit/Object Classification

- Convolutional layers extract spatial and local features from input images (e.g., edges, curves, textures).
- Max pooling layers reduce spatial dimensions while retaining essential features, improving computational efficiency.
- Fully connected layers perform classification based on the extracted features, producing confidence scores across target classes.

Block Diagram - Vivado

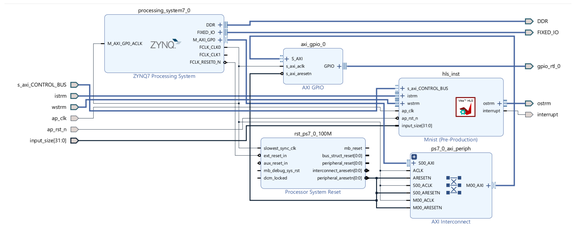

Figure 3. Species detected by YOLOV5

References

1 Esmali, Mohammadreza & Altun, Mustafa. (2023). Energy-Efficient Hardware Implementation of Fully Connected Artificial Neural Networks Using Approximate Arithmetic Blocks. Circuits, Systems, and Signal Processing.

2 Fengyang Feng, Xiao Wei, and Zhicheng Dong. 2024. Reconfigurable Hardware Accelerator for Convolution Operations in Convolutional Neural Networks. In Proceedings of the 2024 12th International Conference on Communications and Broadband Networking (ICCBN '24). Association for Computing Machinery, New York, NY, USA, 20–26

3 H. Wang, D. Li and T. Isshiki, "Reconfigurable CNN Accelerator Embedded in Instruction Extended RISC-V Core," 2023 6th International Conference on Electronics Technology (ICET), Chengdu, China, 2023, pp. 945-954

4 Haobo Ye,Accelerating convolutional neural networks: Exploring FPGA-based architectures and challenges,Journal of Physics: Conference Series,vol 2786, IOP Publishing

5 J. Li, K. -F. Un, W. -H. Yu, P. -I. Mak and R. P. Martins, "An FPGA-Based Energy-Efficient Reconfigurable Convolutional Neural Network Accelerator for Object Recognition Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 9

6 Karakchi, R., Robertson, N. (2024). An Approach to Mitigate CNN Complexity on Domain-Specific Architectures. In: Daimi, K., Al Sadoon, A. (eds) Proceedings of the Second International Conference on Advances in Computing Research (ACR'24). ACR 2024. Lecture Notes in Networks and Systems, vol 956. Springer, Cham.