Final Graduation Project Submissions

Summary of developed files

By

Sara Fouad Zakaria

Summary of the developed files

Folder Structure

The developed files are organized into two folders:

1. **High Level Codes Folder:** Contains the high level codes used to convert the weights, extract the weights into .mem files, the CNN model, train the model code, test the model code, code to prepare the dataset and codes to evaluate the RTL and get the RMSE.
2. **RTL Folder**: Contains the RTL of the Visual Network.

Graduation Project Second Semester Files

File Name	Description
README.pdf	This file represents the documentation of our project thesis, explaining the flow from the high level modelling to the hardware planning.
FPGA-Based CNN-BiLSTM Hardware Accelerator for Localization – 2025.pdf	The Project's thesis.
Sara Fouad GP Video.mp4	The video documenting my journey in the graduation project in the second semester.
Summary.pdf	A PDF having the summary of the 'Graduation Project Second Semester' Folder.

High Level Codes Folder

File Name	Description

RTL Folder

File Name	Description
Activation_relu.v	Comparator-based ReLU activation unit
Batch_Norm.sv	Fixed-point batch normalization FSM
BN_BRAM.sv	Dual-port BRAM for storing BN mean/std and γ/β parameters
BRAM_WEIGHTS.sv	Parameterizable BRAM for convolution and FC weights
BRAM.sv	PingPong BRAM
CNN_wrapper.sv	Top-level integration of all CNN PEs and FC stages
conv2d_top.sv	One PE's top module: line buffer → MAC → BN → ReLU → max-pool
div.v	Iterative fixed-point divider used by the BN unit

fc_controller.v	FSM to sequence fully-connected matrix-vector operations
FIFO_Weights.sv	FIFO for streaming weight values into the MAC array
Fifo.v	Generic parameterized FIFO (used for BN and pooling)
Fully_connected_layer_top.v	Top module for the three FC layers and their pipelining
Line_buffer_FIFO.SV	Sliding window line buffer with built-in FIFO control
Mac_array.sv	Pipelined array of DSP-based multipliers plus accumulation FSM
Mac_bn_fifo.sv	Pipeline register and FIFO interface between MAC and BN
Mac_unit.v	Single-cycle multiply-accumulate unit
Master_controller.v	Global FSM coordinating image read/write, ping-pong BRAM, and PE sequencing
Maxpool_fifo.sv	FIFO for assembling 2×2 max-pool windows
Maxpool.sv	2×2 comparator-tree max pooling unit
PE_ControlUnit.sv	FSM that drives each PE's line buffer, MAC start/stop, BN/ReLU/Pool handshakes
PE_Wrapper.sv	Wrapper to instantiate and connect multiple PEs in the CNN pipeline
Pipelined_fc_network.v	Fully pipelined implementation of the FC network stage
Reg_bn_relu.sv	Pipeline register between BN output and ReLU input
Reg_bnfifo_bn.sv	Register stage between BN FIFO output and BN input
Reg_buffer_mac.sv	Register stage between line buffer output and MAC array
Reg_mac_bnfifo.sv	Register stage between MAC output and BN FIFO input
Reg_maxfifo_maxpool.sv	Register stage between max-pool FIFO output and max-pool unit
Reg_relu_max_fifo.sv	Register stage between ReLU output and pooling FIFO
Relu.sv	Lightweight combinational ReLU with enable/done handshake
Run_wrapper.txt	Script/instructions for running the full CNN wrapper simulation
System_controlUnit.sv	Top-level controller for BRAM bank switching and overall dataflow
Weight_bram.v	Dedicated BRAM for storing all network weights at startup
Wrapper_tb.sv	Testbench for the CNN wrapper, exercising end-to-end inference flow

System Configuration

Hardware Specifications:

- Processor Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
- RAM: 16 GB
- GPU: NVIDIA GeForce RTX 2060

Software Specifications:

- **Operating System**:
  - Windows 10 pro
- **Python Version**:

- Python **3.8+** (Recommended for compatibility with PyTorch and other libraries)

- **Required Libraries & Dependencies**:
  - **PyTorch** (for deep learning framework and model training)
    ```
    bash
    CopyEdit
    pip install torch torchvision torchaudio
    ```
  - **Pandas** (for handling structured data)
    ```
    bash
    CopyEdit
    pip install pandas
    ```
  - **Pillow (PIL)** (for image processing)
    ```
    bash
    CopyEdit
    pip install pillow
    ```
  - **NumPy** (for numerical computations)
    ```
    bash
    CopyEdit
    pip install numpy
    ```
- **GPU Support (Optional but Recommended for Training Speedup)**:
  - NVIDIA GPU (e.g., RTX 3060, RTX 3090, A100)
  - **CUDA Toolkit** (if using GPU acceleration)
    ```
    bash
    CopyEdit
    pip install torch torchvision torchaudio --index-url
    https://download.pytorch.org/whl/cu118
    ```
- **Dataset Requirements**:

  - Ground truth position data in CSV format
  - Camera images in a compatible format (e.g., PNG, JPEG, BMP)