

COLUMBIA UNIVERSITY
FU FOUNDATION SCHOOL OF ENGINEERING AND APPLIED SCIENCE

**Acceleration of Digit Classification Using LeNet
on a SoC FPGA**

CSEE 4840 - EMBEDDED SYSTEMS
PROJECT PROPOSAL

Tharun Kumar Jayaprakash (tj2557)
Vasileios Panousopoulos (vp2518)
Prathmesh Patel (pp2870)
Rishit Thakkar (rht2122)

Spring 2024

Introduction

This project presents an approach to digit classification leveraging the DE1-SoC FPGA board. The aim is to implement a high-performance solution for real-time digit recognition using the MNIST dataset and the LeNET convolutional neural network architecture. The FPGA environment offers unique advantages for accelerating computationally intensive tasks like image processing. Through the utilization of SystemVerilog, the CNN model will be encoded directly into FPGA hardware, exploiting the parallelism and inherent efficiency of hardware-level computations. A critical aspect of this project is integrating a camera module with the system. This setup enables the capture of real-world handwritten digits for classification directly on the FPGA. The system will output the predicted class of the digit, providing a seamless and efficient solution for digit recognition applications. By showcasing the practical implementation of machine learning models on FPGA, particularly in the context of image classification, this project aims to demonstrate the feasibility and effectiveness of FPGA-based solutions for real-time applications in the field of computer vision and pattern recognition.

Background

The proposed digit classification system will be built upon the foundational architecture of the LeNET convolutional neural network (CNN). As shown in Figure 1, LeNET comprises a series of convolutional and subsampling layers, followed by fully connected layers, culminating in a softmax classifier. The architecture is designed to effectively capture spatial hierarchies and patterns within input images while minimizing the number of parameters, making it suitable for embedded and resource-constrained environments.

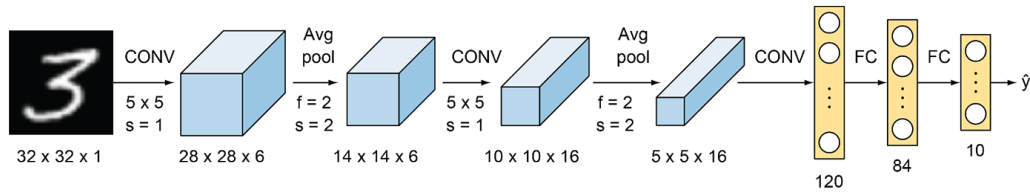


Figure 1: LeNet General Architecture

The initial layers of LeNET perform convolution operations to extract low-level features, followed by subsampling layers that downsample the feature maps, enhancing computational efficiency and reducing dimensionality. Subsequently, the network converges through fully connected layers, ultimately producing class probabilities through a softmax activation function.

Pipeline

The aim of this project is to create an interacting system which will require input from the user and the expected pipeline-dataflow is shown in Figure 2. More specifically, a student will be required to draw a digit on a white paper and then a camera sensor will be used to capture an image and pass it to the ARM processor through a serial communication protocol (i.e. SPI). The initial resolution of the image will depend on the sensor. Following that, the Linux based software layer will pre-process the image, normalizing it to values in the $[0, 1]$ interval and resizing to the input dimensions of the Le-Net CNN, that is 32x32 8-bit pixels. This modified image will be then sent to

the accelerator, which passes it through the convolution and dense layers, resulting in a prediction. Once the prediction is derived, an interrupt will be generated to supply this result to the software layer, which will display the result, along with the original image, on a VGA compatible monitor. This enables the user to see the original image, and the prediction made by the model.

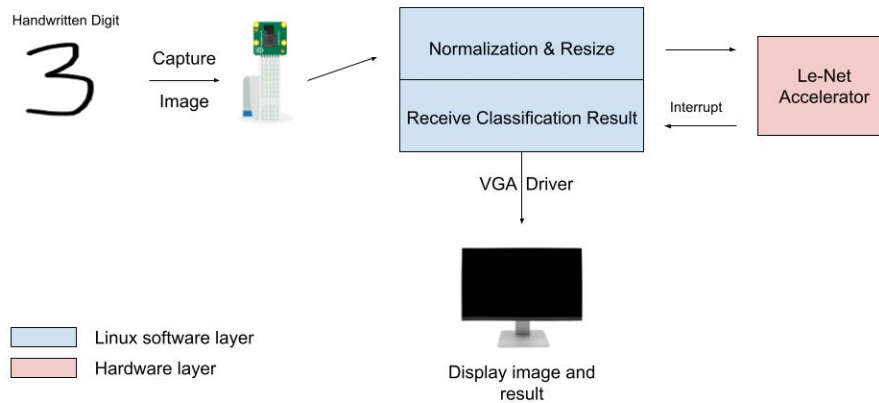


Figure 2: Project Pipeline

Goals & Implementation Steps

The overall goal of this project is to explore how hardware and software stacks can operate cohesively on FPGA devices. This will be done through creating an accelerator for the Le-Net network, using SystemVerilog. This project consists of several steps - milestones that need to be reached and are listed below.

- Design a C-based version of the LeNet architecture
- Decide which camera peripheral will be used and create the interface between the sensor and the Linux processor to be able to receive image data
- Decide the HW pipeline and define the dataflow and synchronization among different SystemVerilog modules, considering memory requirements and constraints (critical)
- Design the device driver for the VGA peripheral
- Implement a first version of the HW layer including only one layer of the network (most likely the final dense layers) and verify it with a Verilator-based testbench
- Design the SW-HW interface to integrate both layers and create a first working version
- Incrementally implement more layers on the HW side

Some of the specifications and requirements that have been already studied and decided are given below.

- Pixel Depth: 8 bits
- Weight representation: 32 bits (fixed point)
- Memory for storing weights on-chip: 240KB