Proposal for STM32-Based System

Ho Jun Kin
Faculty of Engineering
Universiti Teknologi Malaysia
Skudai 81310 Malaysia
hojunkin@graduate.utm.my

Kevin Ng Zhi Hao
Faculty of Engineering
Universiti Teknologi Malaysia
Skudai 81310 Malaysia
kevinngzhi@graduate.utm.my

Yip Wen Xin
Faculty of Engineering
Universiti Teknologi Malaysia
Skudai 81310 Malaysia
yipwenxin@graduate.utm.my

Abstract - This initiative explores the implementation of image recognition using TensorFlow Lite for Microcontrollers (TFLM) and CMSIS-NN on the Discovery STM32Nucleo-F446RE board. The project is designed to facilitate accuracy and performance testing on any embed-enabled device accommodating the final binary. The model, trained on the CIFAR-10 dataset, classifies objects into ten categories. The methodology involves implementing CMSIS-NN on the STM32F44RE Nucleo board, leveraging its CORTEX-M4 core with SIMD and DSP capabilities. The project processes computer-sourced images on the microcontroller, utilizing the ARM Cortex M4's ART accelerator and FPU over Flash memory. Key steps include neural network configuration, code generation, library integration, and inference execution. The output is transmitted to the computer for tabulation. CMSIS-NN optimization aims to enhance efficiency in neural network inference and image processing tasks, optimizing pixel format conversion for memory efficiency. The Cortex-M4's limitations in memory and cache size contrast with its energy efficiency.

Keywords: TensorFlow Lite, CMSIS-NN, Image Recognition, STM32F44RE, Cortex-M4, CIFAR-10, Embedded Systems, Neural Network, Inference.

I. INTRODUCTION

This initiative aims to demonstrate image recognition using TensorFlow Lite for Microcontrollers (TFLM) and CMSIS-NN specifically tailored for the Discovery STM32nucleo-f446re board. Additionally, the project allows for accuracy and performance testing on any embed-enabled device capable of accommodating the final binary in its memory. The foundational structure of the project is derived from an image recognition example, with the model being trained on the CIFAR-10 dataset, enabling it to classify objects into ten distinct categories: plane, car, bird, cat, deer, dog, frog, horse, ship, and truck.

II. METHODOLOGY

In this project, we focus on implementing the CMSIS-NN (Neural Network) algorithm on the STM32F44RE NUCLEO Development board, which features a CORTEX-M4 core. The CMSIS-NN software library is designed specifically for processors with SIMD capability (CORTEX-M0), DSP extension (CORTEX-M4), and MVE extension (CORTEX-M55). The library provides essential functions such as Convolution, Activation, Fully-connected Layer, SVDF layer, Pooling, Softmax, and Basic Math. [1]

The project involves processing images received from a computer on the STM32F44RE microcontroller using the CMSIS-NN library. To ensure compatibility with the microprocessor, the images are converted to the RGB565 color format, a 16-bit format suitable for efficient processing by the M4 core.

Leveraging the ARM Cortex M4's ART accelerator, which optimally utilizes the Floating-Point Unit (FPU) over Flash memory. The project utilizes a pre-trained Caffe model trained on the CIFAR-10 dataset.

The execution of the inference process on the CORTEX M4 MCU involves several key steps:

(1) Neural Network Configuration: Quantization of neural network activations, weights, and layers.
(2) Code Generation: Generation of Neural Network function calls optimized for the ARM architecture.
(3) Library Integration: Importing CMSIS-NN optimized functions into the project.
(4) Inference Execution: Performing the inference process on the microcontroller using the configured neural network. [1]

The final output of the inference process is then transmitted back to the computer and tabulated in csv file.

Utilizing CMSIS-NN has the potential to enhance the efficiency of both neural network inference and image processing tasks. This optimization could result in accelerated pixel format conversion, ensuring memory efficiency and alignment with the limited resources of the microcontroller in the embedded system.

The main difference from the referenced project is in memory utilization. The Cortex-M4, in contrast to the M7, has more limited memory and cache sizes, impacting its efficiency with larger models but excelling in energy efficiency. Moreover, our approach involves direct image input rather than employing a camera module, a cost-effective strategy in our project implementation.

III. REFERENCE

[1] Ioan Lucan Orășan and Cătălin Daniel Căleanu, “ARM Embedded Low Cost Solution for Implementing Deep Learning Paradigms,” Nov. 2020, doi: https://doi.org/10.1109/isetc50328.2020.9301130.