Accelerating DNN Models with DPU: A Hardware-Software Co-Design Approach

Jiawei Yu[1], Atsutake Kosuge[2], Hunga Amano[3]

The University of Tokyo [1,2,3]

E-mail: yujw24@g.ecc.u-tokyo.ac.jp[1], {kosuge, hunga}@dlab.t.u-tokyo.ac.jp[2,3]

1. Introduction

With the proliferation of IoT applications, there is a growing demand for real-time object detection on edge devices. Traditional cloud computing models face challenges such as high latency, bandwidth limitations, and privacy concerns. Edge computing addresses these challenges by moving computational tasks to end devices, significantly reducing system latency while enhancing data privacy protection.

Among various edge computing platforms, FPGA-based DPU solutions emerge as an ideal choice for deploying deep learning models due to their low power consumption, high performance, and reconfigurability.

2. Network

YOLOv3-tiny, as a lightweight version of YOLOv3, maintains reasonable detection accuracy while significantly reducing model parameters and computational complexity. Its single-stage detection architecture and streamlined network structure make it particularly suitable for edge deployment, meeting real-time requirements.

3. Implementation Details

Leveraging Xilinx's Vitis-AI development toolkit, we implemented a systematic model conversion workflow for YOLOv3-tiny. Vitis-AI provides comprehensive tools for deep learning model optimization and deployment on DPU platforms.

Quantization: Using Vitis-AI's quantizer, the model is optimized through calibration-based quantization, converting FP32 weights to INT8 format while maintaining accuracy.

DPU Compilation: The compiler from Vitis-AI toolchain transforms the quantized model into an optimized xmodel format, specifically tailored for DPU acceleration with architecture-aware optimizations.

4. Results and Analysis

The final result is that the FPS is about 70 and the mAP is 25.1%. If we reduce the frequency of real-time display, the FPS will continue to increase.

5. Conclusion

This approach proves particularly valuable for edge computing scenarios where both computational efficiency and detection performance are crucial. While current DPU implementations show promising results, they face limitations in supported operators, which restricts the acceleration of certain neural network architectures. However, this challenge is expected to be addressed with the continuous evolution of DPU technology.