IMPROVED FIELD PROGRAMMABLE GATE ARRAYBASED ACCELERATOR OF DEEP NEURAL NETWORKUSING OPENCL

YAP JUNE WAI

MASTER OF SCIENCE IN ELECTRONIC ENGINEERING

2022

2022

Faculty of Electronics and Computer Engineering

IMPROVED FIELD PROGRAMMABLE GATEARRAYBASED ACCELERATOR OF DEEP NEURAL NETWORKUSING OPENCL

Yap June Wai

Master of Science in Electronic Engineering

2022

IMPROVED FIELD PROGRAMMABLE GATEARRAYBASED ACCELERATOR OF DEEP NEURAL NETWORKUSING OPENCL

YAP JUNE WAI

A thesis submitted
in fulfillment of the requirements for the degree of
Master of Science in Electronic Engineering

Faculty of Electronics and Computer Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2022

DECLARATION

I declare that this thesis entitled “Improved Field Programmable Gate Array (FPGA)Based Accelerator of Deep Neural Network Using OpenCL” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Name	:	YAP JUNE WAI
Date	:	15 AUGUST 2022

APPROVAL

hereby declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of the degree of Master of Science in Electronic Engineering.

Signature	:	
Supervisor Name	:	Professor Dr Zulkalnain bin Mohd Yussof
Date	:	25 AUGUST 2022

DEDICATION

Specially dedicated to my beloved family

ABSTRACT

Being compute-intensive and memory expensive, it is hard to deploy Deep Neural Network (DNN) based models into the embedded devices. Despite recent studies that have shown the efforts to explore the Field Programmable Gate Array (FPGA) as an alternative to deploy DNN-based models such as AlexNet and VGG, there is still a lot of challenges to implement DNN-based object detection model on Field Programmable Gate Array (FPGA). Hence, in this research, the design of a scalable parameterised DNN-based object detection model: Tiny YOLOv2 targeting on FPGA: Cyclone V PCIE Development Kit using High-Level-Synthesis (HLS) tool is explored. Considering the hardware resource limitations in term of computational resources and memory bandwidth, data quantization is proposed to convert the floating point (32-bit) of Tiny YOLOv2 into fixed-point (8-bit) design. To achieve the good performance, an in-depth analysis on the computation complexity and memory footprint of the Tiny YOLOv2 is also studied to find the best quantization scheme for Tiny YOLOv2. The proposed quantization scheme improves the memory requirements to store the parameter from 60 MB to 15 MB, which is around ×4 times improvement compared to the original floating-point design. Finally, the proposed implementation achieves a peak performance density of 0.29 Giga-Operation Per Second (GOPS)/Digital Signal Processing Block (DSP) with only 0.4% loss in the accuracy, which the performance is comparable to all other previous works.

PENAMBAHBAIKAN PEMECUT BERASASKAN TATASUSUNAN BOLEH ATUR GET MEDAN UNTUK JARINGAN NEURAL DALAM MENGGUNAKAN OPENCL

ABSTRAK

Pengiraan dan penggunaan ingatan yang intensif telah memberi cabaran untuk pelaksanaan algorithma Rangkaian Neural Dalam (DNN) dalam sistem terbenam. Walaupun kajian baru-baru ini telah menunjukkan usaha untuk menerokai Tatasusunan Logik Boleh Aturcara (FPGA) sebagai alternatif untuk melaksanakan pengklasifikasian objek seperti "*AlexNet*" dan "*VGG*", namun masih terdapat cabaran untuk melaksanakan algorithma pengesanan objek berasaskan rangkaian neural dalam dengan menggunakan tatasusunan logik boleh aturcara. Oleh itu, dalam penyelidikan ini, pengesanan objek: "Tiny YOLOv2" yang dapat diskalakan ke atas tatasusunan logik boleh aturcara bernama "*Cyclone V PCIE Development Kit*" dengan menggunakan peralatan Tahap Tinggi-Sintesis (HLS) akan dieksploitasikan. Memandangkan sumber yang berhad dari segi sumber pengiraan dan kapasiti memori, pengkuantuman data telah dicadangkan untuk menukar titik terapung ke titik tetap. Demi mencapai pretasi yang baik untuk skim pengkuantuman, analisis terhadap kerumitatan pengkiraan dan memori Tiny YOLOv2 telah dikajikan. Skim pengkuantuman yang dicadangkan dapat mengurangkan keperluan memori untuk menyimpan data sebanyak 60 MB sehingga 15 MB, iaitu kira-kira × 4 kali peningkatan berbanding dengan aritmetika titik mengambang. Akhir sekali, kajian yang dicadangkan mencapai 0.29 GOPS / DSP dengan hanya 0.4% pengorbanan dalam pretasi ketepatan pengesan objek. Pretasi yang dicapaikan dapat dibandingkan dengan semua karya sebelumnya

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Professor Dr Zulkalnain bin Mohd Yussof from Faculty of Electronic and Computer Engineering (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM) for his guidance and assistance throughout the completion of this thesis. I would also like to express my sincere acknowledgement to my co-supervisor, Dr Sani Irwan bin Salim from Faculty of Electronic and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM) for his advice throughout the project. It is a genuine pleasure to express my deep sense of appreciation to UTeM Zamalah Scheme for the financial support throughout this project

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
AI	Artificial Intelligence
ALM	Adaptive Logic Module
ALUT	Adaptive Look Up Table
AP	Average Precision
API	Application Programming Interface
BN	Batch Normalization
BSP	Board Support Package
CNN	Convolutional Neural Network
COCO	Common Object in Context
CPU	Central Processing Unit
CU	Compute-Unit
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
DSP	Digital Signal Processing
FF	Flip-Flop
FPGA	Field Programmable Gate Array
GEMM	General Matrix-Matrix Multiplication

GFLOPS	Giga Floating Point Operations Per Second
GMACS	Giga Multiplication-Accumulates per Second
GOPS	Giga Operations Per Second
GPIO	General Purpose Input Output
GPU	Graphic Processing Unit
GUI	Graphic User Interface
HDL	Hardware Description Language
HLS	High Level Synthesis
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IoT	Internet of Things
IOU	Intersection Over Union
MAP	Mean Average Precision
ML	Machine Learning
NN	Neural Network
OpenCL	Open Computing Language
PE	Processing Element
RAM	Random Memory Access
RCNN	Region Based Convolutional Neural Network
ReLU	Rectifier Linear Unit
ROI	Region of Interest
RTL	Register-Transfer Level
SDK	Software Development Kit
SIMD	Single Instruction Multiple Data

SVD	Singular Value Decomposition
SVM	Support Vector Machine
VOC	Visual Object Classification
YOLO	You Only Look Once

LIST OF APPENDICES

LIST OF PUBLICATIONS

Indexed Journal

Yap, J.W., Yussof, Z.M., Salim, S.I. and Lim, K.C., 2018. Fixed Point ImplementationofTiny-Yolo—v2 using OpenCL on FPGA. International Journal of Advanced ComputerScience and Applications (IJACSA), 9(10), pp.62.

Yap, J.W., Yussof, Z.M., and Salim, S.I., 2019. A Scalable FPGA based Accelerator forTiny-YOLO-v2 using OpenCL. International Journal of Reconfigurable and EmbeddingSystem (IJRES), 8(10), pp.206.

CHAPTER 1

INTRODUCTION

1.1 Background

The rise of the new digital industrial revolution, also known as Industry 4.0, is a significant transformation on the road to an end-to-end value chain with Industrial IoT and decentralized artificial intelligent (AI) in manufacturing, production, and logistic. It is undoubtedly that the AI will be the highlight and the key to propelling the rise of the new era. Deep learning, a subfield in AI family, which recently acts as a catalyst to the growing of AI, is inspired by the artificial neural network algorithm. The deep learning is widely adopted in various applications, which include video surveillance (Xu et al., 2015; Liu et al., 2016; Bashbaghi et al., 2018; Shorfuzzaman et al., 2020; Perez et al., 2021), autonomous vehicle (Bojarski et al., 2016; Sallab et al., 2017; Tian et al., 2018, Kuuti et al., 2020), mobile robot vision (Zhu et al., 2017; Levine et al., 2018, Chen et al., 2020). The deep learning led to early success in 1986, notably a neural network classification model named as LENET which is able to recognize the handwritten digits. Unfortunately, the neural network did not catch the attention to solve large scale problem. It is mainly due to the limitations in data availability and computing power in early 1990. With a large database which is known as ImageNet containing millions of labelled images was created in 2010, which led to the biggest breakthrough in Artificial Intelligence (AI) history. With the rapid growth of computer processing power and larger database, the first large scale deep neural network (DNN), namely AlexNet was introduced in 2012. AlexNet achieved a top-5 error of 15.3%, more than 10.8% error rate lower than the runner up in the ImageNet Large Scale Visual

Recognition Challenge (ILSVRC). The results significantly outperformed all the prior competitors and any other traditional classification approach. This led to the rapid evolution of Deep Neural Network (DNN) model, consequently the emerging of DNN-based classification model such as VGG16 and the emerging of DNN- based object detection models such as Recurrent Neural Network (RNN), Region-Based Convolutional Neural Network (RCNN), and You-Only-Look-Once (YOLO). The accuracy of these models has increased tremendously since 2012, with an increase in the complexity and number of layers in the network. The increase in the model size of deep learning model greatly increases the number of parameters, consequently leading to a significant increase in the computational requirements, memory bandwidth and storage required to store the parameters. For example, AlexNet requires around 244 MB of parameters and over 1.4 billion of operations to perform classification on a single input image, while VGG16 requires around 500 MB of parameters and over 30 billion of operations to perform classification on a single input image.

This gives a challenge to the deployment of DNN model in consumer devices that use embedded, such as household appliances, mobile phones, digital camera, and so on. Compared to stationary workstation such as desktop, embedded systems are more limited in terms of computational power, memory bandwidth, and power consumption. Hence, recent researches have explored the hardware implementation of deep learning models as an alternative to accelerate the DNN model. Field Gate Programmable Array (FPGA) can provide a relatively high performance, more energy-efficient, flexibility and fast development cycle especially with the introduction of High-Level synthesis (HLS) tool such as Open Computing Language (OpenCL), enabling the auto-compilation from high- level programming such as C/C++ to Hardware Description Language (HDL) such as Verilog or VHDL. It greatly reduces the effort on hardware development where the development