CONVOLUTION AND MAX POOLING LAYER ACCELERATOR FOR CONVOLUTIONAL NEURAL NETWORK

GOH JINN CHYN

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Electrical Engineering (Computer and Microelectronic Systems)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JANUARY 2020

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time. Lastly, this project also dedicated to my wife who always keep me harmonious and helping me putting pieces together.

ACKNOWLEDGEMENT

I would like to thank all the people who contributed in some ways to the works described in this thesis. First and foremost, I would like to express my sincere gratitude towards my Master Project Supervisor, Prof. Madya Dr. Muhammad Nadzir Bin Marsono for his dedicated guidance and advice throughout the project. During this research, he contributed to a rewarding graduate school experience by giving me intellectual freedom in my work, engaging me in new ideas, and demanding a high quality of work in all my endeavors.

I would like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love. Lastly, I would like to give my thanks to all my friends and family for helping survive all the stress from this year and not letting me gives up.

ABSTRACT

Convolutional Neural Network (CNN) are widely used in the field of computer vision and show its great advantages in image classification, object recognition, video surveillance. Hence, the performance of CNN playing more important role during the development of the application which applying CNN algorithm. In this paper, an accelerator is developed for improving the performance of CNN. The proposed accelerator targeted the most computation intensive functions in CNN, which are convolution and max pooling. The developed accelerator is targeting on CNN with 64×64 input image size, 5×5 filter size and 2×2 max pooling. By using Vivado, the period, clock cycle and resources required to run convolution and max pooling are measured. Three proposed methodology are combined to enhance the performance of CNN: (i) unrolling (ii) pipelining (iii) combination of convolution and max pooling layer. Tradeoff between the performance and hardware cost required to build the accelerator are simulated and analyzed. The performance of the new proposed accelerator are proven to be four times better and with limited increase of the hardware cost, addition of 60% of logic gates compared to the existing work.

ABSTRAK

Rangkaian neural convolutional (CNN) telah digunakan dalam bidang visi komputer dan menunjukkan kelebihannya dalam klasifikasi imej, pengenalan objek, pengawasan video. Oleh itu, prestasi CNN memainkan peranan yang amat penting dalam perkembangan aplikasi yang menggunakan algoritma CNN. Dalam makalah ini, perkakasan akan dibangunkan untuk meningkatkan prestasi CNN. Perkakasan yang dicadangkan itu menyasarkan fungsi intensif pengiraan yang paling banyak di CNN, iaitu pengumpulan dan penggabungan maks. Penderas yang dicadangkan akan menumpukan CNN dengan 64 kali 64 saiz imej, 5 kali 5 saiz penapis dan 2 kali 2 maks pooling. Dengan menggunakan Vivado, tempoh, kitaran jam dan sumber yang diperlukan untuk menjalankan pengukuhan dan pengumpulan maksima diukur. Tiga metodologi yang diajukan digabungkan untuk meningkatkan prestasi CNN: (i) pembongkaran (ii) perpaduan lapisan (iii) kombinasi lapisan convolusi dan max. Pembatalan antara prestasi dan kos perkakasan yang diperlukan untuk membina perkakasan akan disimulasikan dan dianalisis. Prestasi pemecut yang dicadangkan baru terbukti empat kali lebih baik dan dengan peningkatan kos perkakasan yang terhad, iaitu penambahan sebanyak 60% daripada pintu logik berbanding dengan kerja yang sedia ada.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2-D	-	Two Dimension
ALU	-	Arithmetic Logic Unit
ASIC	-	Application Specific Integrated Circuit
CPU	-	Central Processing Unit
CNN	-	Convolution Neural Network
FPGA	-	Fixed Programmable Gate Array
GPU	-	Graphic Processing Unit
HLS	-	High Level Synthesis
IoT	-	Internet of Things
MAC	-	Multiplier-Accumulator
OFM	-	Output Feature Mapping
OFMP	-	Output Feature Mapping Parallelism
PE	-	Processing Element
RAM	-	Random Access Memory
SoC	-	System on Chip

CHAPTER 1

INTRODUCTION

1.1 Research Background

Convolutions Neural Networks (CNN) are widely used in the field of computer vision and show its great advantages in image classification, object recognition, video surveillance [1]. The applications of CNN are usually realized by Central Computing Unit (CPU) and Graphic Processing Unit (GPU). However, general purpose computer has limited computing resources and parallelism. Although GPU in the computer is still able to perform CNN with its special characteristics of parallel computing of large-scale data, the hardware resource and power comsumption is too high (for example: 33W for Nvidia GTX840M and 235W for NVIDIA Tesla K40) [2][3]. Hence, CNN accelerators require a trade-off between flexibility and energy efficiency. Application Specific Integrated Circuit (ASIC) design is one of the good choice to obtain the best power efficiency but only specialized CNN models are able to be implemented into ASIC circuits due to its flexibility [1].

In recent years, there are some Fixed-Programmable-Gate-Array (FPGA)-based CNN accelerators developed [4][5]. The current trends showing more and more FPGA-based CNN accelerator implemented by using high level synthesis tools [4][5]. By using FPGA, not only the productivity of the engineers or programmers increased but also enable them to play around with the FPGA just like CPUs/GPUs [4]. The specialization of the FPGAs provides a compromise between the flexibility of a general purpose processor (GPP) and the performance of the ASIC. FPGAs and ASICs are both designed for specific applications but FPGAs are programmable and its design can be modified from time to time therefore FPGAs are more flexible compared to ASICs. In the widely used embedded systems and SoC world, FPGAs are the great choice to be implemented. The evaluation of GPP, ASIC and FPGA in different perspective are shown in Table 1.1

Table 1.1: Evaluation GPP, ASIC and FPGA in different perspective

	GPP	ASIC	FPGA
Performance	Low	Very High	High
Power	Large	Small	Moderate
Flexibility	Excellent	Poor	Excellent
Hardware Design	Not Available	Large	Moderate
Software Design	Large	Not Available	Large
Reuse	Excellent	Poor	Excellent
Market	Very Large	Small	Very Large

From Table 1.1, the strengths and weaknesses of GPP, ASIC,and FPGA can be determined easily. FPGA is the best choice for developing a dedicated application accelerator for performance and flexibility.

1.2 Problem Statement

CNN can be divided into two sections which are training and processing. Once the CNN models completed its training, users will only run the CNN processing part to work on the specific application. In processing parts, there are 3 main layers exist in CNN algorithm. Convolution, pooling and full connected layers contribute up to 90% of execution time [6].

To improve the performance of CNN algorithm to overcome the bottleneck of software programming in GPP, there are several hardware solution proposed by other researchers. Different kind of approaches are introduced by them to utilize the hardware for running the algorithm with optimized solution. Besides that, there are also some approaches introduced to combine the different layer in CNN to minimize the data loading or storing to the main memory to achieve better memory organization.

Although there are some studies had done for developing accelerators, there are some limitation exist in the proposed accelerators. On the other hand, the proposed accelerators also have their own strength and advantages. As an example, the unrolling methodology introduced by [7][8] enable the convolution layer run in parallel but it might be limited by the resources of the processors.

By combining the methodology of unrolling, pipe-lining, and good memory organization, a new basic model accelerator are introduced. By profiling the CNN algorithm in the new basic accelerator, an accurate profiling result can be obtained. With the new profiled results, we can always fine tune the methodologies implemented in the processor to obtain the optimized result. In the end of the project, a new dedicated accelerator is developed for CNN algorithm.

1.3 Project Objective

The aim of this project is to develop an accelerator for CNN application. To achieve the target, the objective of this projects can be divided as below:

- To perform software profiling to investigate time-consuming CNN internal function.
- To propose hardware accelerator architecture for improving convolution and max pooling layer performance.
- To evaluate the performance-area trade-off of proposed CNN hardware accelerator against the conventional CNN implementation.

1.4 Project Scope

The CNN algorithms plays an important role on several fields and it is able to be applied in various types of platform. Hence, there are some scopes set for ensuring the developing of the new enhancing accelerator can be done efficiently. The CNN algorithm is able to applied in GPP, ASIC, and FPGA processor. In this project, we will focus on the development of FPGA-based accelerator. In this project, the most intensive function of CNN algorithm are analysed and we only focusing on improving convolution and max pooling layer in CNN algorithm.

Vivado High Level Synthesis (HLS) is used to compile the c-code of CNN algorithm for calculating the resources needed. Besides that, the performance of the

CNN algorithm run in conventional accelerator and proposed accelerator are simulated and evaluated by calculating the maximum frequency allowed and number of cycles needed to complete CNN algorithm.

1.5 Chapter Organization

The report consists of five chapters. Chapter 1 is the introduction of the project background, problem statement, project objectives, and scopes of this project.

Chapter 2 will focus on the literature review. The theory of the CNN algorithms and the most computing intensive arithmetic function in CNN are reviewed and studied. Besides that, the proposed conventional accelerators for CNN are analyzed. The strength and the weakness of those conventional accelerators are investigated. The factors that contribute to advantages and weaknesses of those accelerators are reviewed as well.

Chapter 3 presents the steps to develop the proposed algorithms. The details of their implementation are presented in this chapter. This chapter also shows the planning of the whole research and also the execution work have done in Project 1. Chapter 4 is the chapter to show the profiled result in current work and also tabulate the expected outcome for the new accelerator developed. Chapter 5 summarize the content the paper and also presenting the future work can be done to improve the project.

REFERENCES

1. Tu, F., Yin, S., Ouyang, P., Tang, S., Liu, L. and Wei, S. Deep Convolutional Neural Network Architecture With Reconfigurable Computation Patterns. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017. 25(8): 2220–2233. ISSN 1063-8210. doi:10.1109/TVLSI.2017.2688340.

2. V. Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J. and Y. Ng, A. Building high-level features using large scale unsupervised learning. *Proceedings of ICML*, 2011. 1.

3. NVIDIAÂ® TESLAÂ® K40 Workstation Card.

4. Huang, C., Ni, S. and Chen, G. A layer-based structured design of CNN on FPGA. *2017 IEEE 12th International Conference on ASIC (ASICON)*. 2017. 1037–1040. doi:10.1109/ASICON.2017.8252656.

5. Yao, Y., Duan, Q., Zhang, Z., Gao, J., Wang, J., Yang, M., Tao, X. and Lai, J. A FPGA-based Hardware Accelerator for Multiple Convolutional Neural Networks. *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. 2018. 1–3. doi: 10.1109/ICSICT.2018.8565657.

6. Wang, Y., Li, H. and Li, X. A Case of On-Chip Memory Subsystem Design for Low-Power CNN Accelerators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018. 37(10): 1971–1984. ISSN 0278-0070. doi:10.1109/TCAD.2017.2778060.

7. Chidambaram, S., Riviello, A., PierreLanglois, J. M. and David, J. Accelerating the Inference Phase in Ternary Convolutional Neural Networks Using Configurable Processors. *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*. 2018. 94–99. doi:10.1109/DASIP. 2018.8596860.

8. Chen, W., Wu, H., Wei, S., He, A. and Chen, H. An Asynchronous Energy-Efficient CNN Accelerator with Reconfigurable Architecture. *2018 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. 2018. 51–54. doi:10.1109/ASSCC. 2018.8579304.

9. Xie, W., Zhang, C., Zhang, Y., Hu, C., Jiang, H. and Wang, Z. An Energy-Efficient FPGA-Based Embedded System for CNN Application. 2018. 1–2. doi:10.1109/EDSSC.2018.8487057.

10. Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., Li, L., Chen, T., Xu, Z., Sun, N. and Temam, O. DaDianNao: A Machine-Learning Supercomputer. *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 2014. ISSN 1072-4451. 609–622. doi:10.1109/MICRO.2014.58.

11. Jafri, S. M. A. H., Hemani, A. and Stathis, D. Can a reconfigurable architecture beat ASIC as a CNN accelerator? *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. 2017. 97–104. doi:10.1109/SAMOS.2017.8344616.

12. Bong, K., Choi, S., Kim, C. and Yoo, H. Low-Power Convolutional Neural Network Processor for a Face-Recognition System. *IEEE Micro*, 2017. 37(6): 30–38. doi:10.1109/MM.2017.4241350.

13. FPGA Fundamentals. URL `https://www.ni.com/en-us/innovations/white-papers/08/fpga-fundamentals.html`.

14. Gan Feng, Zuyi Hu, Song Chen and Feng Wu. Energy-efficient and high-throughput FPGA-based accelerator for Convolutional Neural Networks. *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. 2016. 624–626. doi:10.1109/ICSICT.2016.7998996.

15. Mujawar, S., Kiran, D. and Ramasangu, H. An Efficient CNN Architecture for Image Classification on FPGA Accelerator. *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*. 2018. 1–4. doi:10.1109/ICAECC.2018.8479517.

16. Ma, Y., Cao, Y., Vrudhula, S. and Seo, J. Optimizing the Convolution Operation to Accelerate Deep Neural Networks on FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2018. 26(7): 1354–1367. ISSN 1557-9999. doi:10.1109/TVLSI.2018.2815603.

17. Gong, L., Wang, C., Li, X., Chen, H. and Zhou, X. MALOC: A Fully Pipelined FPGA Accelerator for Convolutional Neural Networks With All Layers Mapped on Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018. 37(11): 2601–2612. doi:10.1109/TCAD.2018.2857078.

18. Alizadeh, M. and Sharifkhani, M. Extending RISC- V ISA for Accelerating the H.265/HEVC Deblocking Filter. 2018. 126–129. doi:10.1109/ICCKE.2018.8566467.

19. Banz, C., Dolar, C., Cholewa, F. and Blume, H. Instruction set extension for high throughput disparity estimation in stereo image processing. *ASAP 2011 - 22nd IEEE International Conference on Application-specific Systems, Architectures and Processors*. 2011. ISSN 1063-6862. 169–175. doi:10.1109/ASAP.2011.6043265.

20. Stallings, W. *COMPUTER ORGANIZATION AND ARCHITECTURE*. 2018.