IMAGE RECOGNITION USING CAPSULE NETWORK ON FPGA

SALIM ALI ABDULRRAZIQ ADREES

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronics System)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JANUARY 2020

DEDICATION

To my beloved family father, mother, brothers and sisters.

ACKNOWLEDGEMENT

I would first like to thank my supervisor Dr. Ismahani Binti Ismail. Her office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge the contributions of Universiti Teknologi Malaysia (UTM). Finally, I would like to thank all my family members, friends, and those people who are giving me the mentality support, courage, and as well as their assistance and supports in completing this project.

ABSTRACT

A capsule neural network (CapsNet) is a new approach in artificial neural network (ANN) that produces a better model hierarchical relationship. A capsule is a set of neurons. Each capsule generates vector which presents the details of an entity. The performance of CapsNet on graphics processing unit (GPU) is considerably better than convolutional neural network (CNN) at recognizing highly overlapping digits in images. Nevertheless, this new method has not been designed as accelerator on field-programmable gate array (FPGA) to measure the speedup performance and compared it with the GPU. This is because of the lack of hardware design experience. This project aims to design the CapsNet model (accelerator) on FPGA using high-level synthesis (HLS). Then, the performance between FPGA and GPU will be compared, mainly in terms speedup and accuracy. Behavioural module is synthesized using HLS tools on FPGA then it is evaluated and validated using MNIST dataset. The module is designed to receive features vectors of handwritten digits image as an input and pass it through several layers to predict the output. The speed-up performance on FPGA is expected to be higher than GPU, but FPGA accuracy is expected to be slightly lower than GPU. The module can be useful in detecting the license plate of fast-moving vehicles and many other applications.

ABSTRAK

Rangkaian neural kapsul (CapsNet) adalah pendekatan baharu dalam rangkaian saraf Buatan (ANN) yang menghasilkan hubungan hierarki model yang lebih baik. Kapsul adalah kumpulan neuron yang menghasilkan vektor dimana mewakili keperincian satu entit. Prestasi CapsNet pada unit pemprosesan grafik (GPU) telah menemui tahap pencapaian pada pangkalan data Institut Kebangsaan Standard dan Teknologi Modified (MNIST) dan jauh lebih baik daripada rangkaian saraf lingkaran untuk mengenali digit yang sangat bertindih dalam imej. Walau bagaimanapun, teori baru ini belum dilaksanakan sebagai pemecut pada cip yang boleh aturcara (FPGA) untuk mengukur prestasi kelajuan dan membandingkannya dengan GPU. Ini disebabkan oleh kekurangan kepakaran berkaitan reka bentuk perkakasan. Projek ini bertujuan untuk merekabentuk model CapsNet (pemecut) pada FPGA menggunakan alat sintesis peringkat tinggi (HLS). Prestasi di antara FPGA dan GPU akan dibandingkan terutamanya kelajuan and ketepatan. Modul kelakuan disintesis menggunakan alat HLS pada FPGA kemudian ia dinilai dan disahkan menggunakan dataset MNIST. Modul ini direka bentuk untuk menerima vektor ciri imej digit tulisan tangan sebagai input dan melalui beberapa lapisan untuk meramalkan outputnya. Prestasi kelajuan pada FPGA dijangka lebih tinggi daripada GPU, tetapi ketepatan FPGA dijangka sedikit lebih rendah daripada GPU. Modul ini berguna dalam mengesan plat lesen kenderaan yang bergerak pantas dan banyak aplikasi lain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	-	Artificial Neural Network
CapsNet	-	Capsule Neural Network
GPU	-	Graphics Processing Unit
MNIST	-	Modified National Institute of Standards and Technology
CNN	-	Convolutional Neural Network
FPGA	-	Field-programmable gate array
HLS	-	High-level Synthesis
UTM	-	Universiti Teknologi Malaysia
LB	-	Logic Blocks
LUT	-	Lookup Table
ROM	-	Read only memory
RAM	-	Random access memory
KNN	-	K-nearest neighbour
GPP	-	General-purpose processor
ADAS	-	Auto drive assistant system
FSM	-	Finite state machine
SoC	-	System-on-Chip
NaN	-	Not a Number
HDL	-	Hardware Description Language
BSOM	-	Binary Self-Organizing Map
ASM	-	Algorithmic State Machine

LIST OF SYMBOLS

W	-	Weights of the model
u	-	Primary capsule vector
b	-	initial logics
c	-	coupling coefficient
S	-	the sum of all inputs in each digit capsule
v	-	Instantiation vector
L_k	-	Margin loss
λ	-	Constant value (0.5)
$m+$	-	Constant value (0.9)
$m-$	-	Constant value (0.1)
r	-	Number of epochs in routing by agreement algorithm
l	-	Number of layers in routing by agreement algorithm
$\hat{u}$	-	The main input for routing by agreement algorithm
MAG	-	Represent the probability of each digit

CHAPTER 1

INTRODUCTION

1.1 Overview

A Capsule Neural Network (CapsNet) is the latest type of artificial neural network (ANN) which propose better hierarchical relationships in the model. A capsule is neurons grouped together which the features vector represents the instantiation parameters of a certain entity [1]. Similar to normal neural network, a CapsNet is designed in multiple layers. The capsules in the first layer are the primary capsules which the small primary features are detected. Capsules in second layers are the routing capsules which detect large and complex objects [2]. Nevertheless, this new machine learning system has not been implemented on any embedded system, for example Field Programmable Gate Array (FPGA). Therefore, in this project, the aim is to identify which part in CapsNet consumes the highest amount of time from overall execution time and design an accelerator on FPGA to implement this part and measure the amount of enhancement in overall execution time without effecting the accuracy of the module.

1.2 Problem statement

The complex computation of the CapsNet algorithm takes a long time and as data sizes increase, its running time can stretch to several hours. The CapsNet has not been implemented on hardware system to compare the speed up performance and accuracy with graphics processing unit (GPU).

1.3 Research objectives

There are three objectives for this research:

1. To implement the CapsNet algorithm in MATLAB using GPU and find the statistics of the module which are the delay to find the results and accuracy of the module in predicting new images.
2. To identify the computation intensive part which consumes the maximum amount of time from over all execution time in MATLAB and design FPGA accelerator for this part and measure the delay of the correct output.
3. To compare the speedup performance and accuracy between FPGA and GPU.

1.4 Scope of the project

The followings are the considered as the project's scope:

1. Capsule network is the latest technology in the field of machine learning, and it achieved a better performance than the convolutional neural networks (CNN). Simulate for this algorithm is done in GPU using high level languages (HLL).
2. FPGA is a strong embedded hardware that offers low power, high parallelism and high reconfigurability to meet the demand of high computational speed of different machine learning algorithms.

1.5 Contributions

This hardware design can be used in several fields such as security reasons. For example, the module can be useful in detecting the license plate of fast-moving vehicles and many other applications.

1.6 Report organization

The are 5 following chapters. Chapter 2 is about literature review which describes the existing works related to this project. Chapter 3 describes the methodology used in the project. Chapter 4 presents the result of the project. Chapter 5 summarizes the conclusion. Chapter 6 will discuss future work.

REFERENCES

[1] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856-3866.

[2] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," 2018.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, p. 436, 2015.

[4] A. Muthuramalingam, S. Himavathi, and E. Srinivasan, "Neural network implementation using FPGA: issues and application," *International journal of information technology,* vol. 4, no. 2, pp. 86-92, 2008.

[5] Y. Taright and M. Hubin, "FPGA implementation of a multilayer perceptron neural network using VHDL," in *ICSP'98. 1998 Fourth International Conference on Signal Processing (Cat. No. 98TH8344)*, 1998, vol. 2, pp. 1311-1314: IEEE.

[6] Z.-H. Li, J.-F. Jin, X.-G. Zhou, and Z.-H. Feng, "K-nearest neighbor algorithm implementation on FPGA using high level synthesis," in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2016, pp. 600-602: IEEE.

[7] R. Narayanan, D. Honbo, G. Memik, A. Choudhary, and J. Zambreno, "Interactive presentation: An FPGA implementation of decision tree classification," presented at the Proceedings of the conference on Design, automation and test in Europe, Nice, France, 2007.

[8] F. Ortega-Zamorano, J. M. Jerez, D. U. Muñoz, R. M. Luque-Baena, and L. Franco, "Efficient implementation of the backpropagation algorithm in FPGAs and microcontrollers," *IEEE transactions on neural networks and learning systems,* vol. 27, no. 9, pp. 1840-1850, 2016.

[9] A. Ahmad, B. Kakillioglu, and S. Velipasalar, "3D Capsule Networks for Object Classification from 3D Model Data," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 2225-2229: IEEE.

[10] H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," *arXiv preprint arXiv:1604.06338,* 2016.

[11] B. Gagana, H. U. Athri, and S. Natarajan, "Activation Function Optimizations for Capsule Networks," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 1172-1178: IEEE.

[12] A. Shahroudnejad, P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Improved explainability of capsule networks: Relevance path by agreement," in *2018 IEEE*

Global Conference on Signal and Information Processing (GlobalSIP), 2018, pp. 549-553: IEEE.

[13] V. Betz and J. Rose, "FPGA routing architecture: Segmentation and buffering to optimize speed and density," in *Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays*, 1999, pp. 59-68: ACM.

[14] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems—A review," *IEEE transactions on industrial electronics,* vol. 54, no. 4, pp. 1824-1842, 2007.

[15] C. Cullinan, C. Wyant, T. Frattesi, and X. Huang, "Computing performance benchmarks among cpu, gpu, and fpga."

[16] D. M. Khalil-hani, *Design of Digital Systems*. UTM Book Series 2018.

[17] J. Su, "Design and Development of an FPGA-based Distributed Computing Processing Platform," 2011.

[18] V. Betz. ([Accessed March 2019]). *"University of Toronto," [Online].* Available: http://www.eecg.toronto.edu/~vaughn/challenge/fpga_arch.html.

[19] Y. Li and W. Chu, "A new non-restoring square root algorithm and its VLSI implementations," in *Proceedings International Conference on Computer Design. VLSI in Computers and Processors*, 1996, pp. 538-544: IEEE.

[20] E. Nurvitadhi *et al.*, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017, pp. 5-14: ACM.

[21] H. M. Vo, "Implementing the on-chip backpropagation learning algorithm on FPGA architecture," in *2017 International Conference on System Science and Engineering (ICSSE)*, 2017, pp. 538-541: IEEE.

[22] H. Meng, K. Appiah, A. Hunter, and P. Dickinson, "Fpga implementation of naive bayes classifier for visual object recognition," in *CVPR 2011 WORKSHOPS*, 2011, pp. 123-128: IEEE.