



شبكة المعلومات الجامعية  
التوثيق الإلكتروني والميكروفيلم

# بسم الله الرحمن الرحيم



**HANAA ALY**



شبكة المعلومات الجامعية  
التوثيق الإلكتروني والميكروفيلم



# شبكة المعلومات الجامعية التوثيق الإلكتروني والميكروفيلم



**HANAA ALY**



شبكة المعلومات الجامعية  
التوثيق الإلكتروني والميكروفيلم

# جامعة عين شمس التوثيق الإلكتروني والميكروفيلم

## قسم

نقسم بالله العظيم أن المادة التي تم توثيقها وتسجيلها  
علي هذه الأقراص المدمجة قد أعدت دون أية تغييرات



## يجب أن

تحفظ هذه الأقراص المدمجة بعيدا عن الغبار



**HANAA ALY**



HARDWARE ACCELERATION OF CONVOLUTIONAL  
NEURAL NETWORKS USING APPROXIMATE  
COMPUTING AND DYNAMIC PARTIAL  
RECONFIGURATION

By

Eman Youssef Ahmed Safina

A Thesis Submitted to the  
Faculty of Engineering at Cairo University  
in Partial Fulfillment of the  
Requirements for the Degree of  
**MASTER OF SCIENCE**  
in  
**Electronics and Communications Engineering**

FACULTY OF ENGINEERING, CAIRO UNIVERSITY  
GIZA, EGYPT  
2021

HARDWARE ACCELERATION OF CONVOLUTIONAL  
NEURAL NETWORKS USING APPROXIMATE  
COMPUTING AND DYNAMIC PARTIAL  
RECONFIGURATION

By  
Eman Youssef Ahmed Safina

A Thesis Submitted to the  
Faculty of Engineering at Cairo University  
in Partial Fulfillment of the  
Requirements for the Degree of  
**MASTER OF SCIENCE**  
in  
**Electronics and Communications Engineering**

Under the Supervision of

Assoc. Prof. Dr. Ahmed Khattab

Prof. Dr. Hamed Elsimary

Associate Professor  
Electronics & Communication  
Engineering Dept.,  
Faculty of Engineering, Cairo University

Professor  
Prince Sattam bin Abdulaziz University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY  
GIZA, EGYPT  
2021

HARDWARE ACCELERATION OF CONVOLUTIONAL  
NEURAL NETWORKS USING APPROXIMATE  
COMPUTING AND DYNAMIC PARTIAL  
RECONFIGURATION

By  
Eman Youssef Ahmed Safina

A Thesis Submitted to the  
Faculty of Engineering at Cairo University  
in Partial Fulfillment of the  
Requirements for the Degree of  
**MASTER OF SCIENCE**  
in  
**Electronics and Communications Engineering**

Approved by the  
Examining Committee

---

**Assoc. Prof. Dr. Ahmed Khattab**, Thesis Main Advisor

---

**Prof. Dr. Hamed Elsimary**, Advisor

---

**Prof. Dr. Mohammed Fathy**, Internal Examiner

---

**Assoc. Prof. Dr. Mohammed Abdelghany**, External Examiner  
German University in Cairo

FACULTY OF ENGINEERING, CAIRO UNIVERSITY  
GIZA, EGYPT  
2021

**Engineer's Name:** Eman Youssef Ahmed Safina  
**Date of Birth:** 3 / 7 / 1991  
**Nationality:** Egyptian  
**E-mail:** emanyousef1991@gmail.com  
**Phone:** 01146063731  
**Address:** Alzyton-Cairo  
**Registration Date:** 3 / 2016  
**Awarding Date:** 7 / 2021  
**Degree:** Master of Science  
**Department:** Electronics and Communications Engineering



**Supervisors:**

Assoc. Prof. Dr. Ahmed kattab  
Prof. Dr. Hamed Elsimary

**Examiners:**

Assoc. Prof. Dr. Ahmed kattab (Thesis main advisor)  
Prof. Dr. Hamed Elsimary (advisor)  
Prof. Dr. Mohammed Fathy (Internal examiner)  
Assoc. Prof. Dr. Mohammed Abdelghny (External examiner)

**Title of Thesis:**

Hardware Acceleration of Convolutional Neural Networks using Approximate Computing and Dynamic Partial Reconfiguration.

**Key Words:**

Convolution Neural Network ; Approximate Computing ; Energy Efficiency ; Precision Scaling ; Dynamic Partial Reconfiguration.

**Summary:**

In this work, I have trained new four different convolutional neural networks (CNNs) to recognize four different datasets MNIST, Fashion MNIST, SVHN and CIFAR-10. Then, the CNNs are tested for recognition. The resulting trainable weights are approximated using precision scaling. The four networks are tested again while using this approximation. A new hardware architecture is proposed to recognize three datasets (MNIST- Fashion MNIST- SVHN) while using precision scaling approximation. This architecture is implemented on Xilinx XC7Z020 FPGA. The resulting power and energy consumed to recognize each image in each dataset is reported. The results show significant reduction in energy consumption while having minor loss in accuracy. This approximation is significant because CNN requires a lot of computation, and hence, consumes large power.

# Disclaimer

I hereby declare that this thesis is my own original work and that no part of it has been submitted for a degree qualification at any other university or institute.

I further declare that I have appropriately acknowledged all sources used and have cited them in the references section.

Name: Eman Youssef Ahmed

Date:     /     / 2021

Signature:



## *Acknowledgements*

Thank you to my supervisor, Dr. Hamed Elsimary, Dr. Magdy Ali El-Moursy, Dr. Ahmed Khattab and Dr. Hassan Mostafa for providing guidance and feedback throughout this thesis.

Also, I send my acknowledgment to the Cloud Computing Center of Excellence at the Electronics Research Institute (ERI) - Egypt, for their help and giving the chance for me to run my program/application on the Cloud/HPC system available.

Finally, I send my acknowledgment to ONE Lab at Cairo University.

# Table of Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Figures</b>	<b>vi</b>
<b>Listings</b>	<b>viii</b>
<b>List of Acronyms</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Convolutional Neural Networks (CNN)	1
1.1.1 CNN for Computer Vision: Common Applications	1
Agriculture	1
Self-driving Cars	1
Surveillance	1
Healthcare	2
1.2 Research Objective	2
1.3 Thesis Outline	2
<b>2 Literature Survey</b>	<b>3</b>
2.1 Neuromorphic Computing	3
2.1.1 Neuromorphic Chips	3
Tianjic Chip	3
Intel's Loihi Chips	4
The Pohoiki Computer	4
Blue Brain Project	4
SpiNNaker	4
2.1.2 Biological Neurons at Creatures' Brain	5
Processing	5
Communication	7
Storage	7
2.1.3 Artificial Neuron	7
Activation Function	7
2.2 Artificial Neural Networks (ANN)	7
2.3 Approximate Multipliers for Neural Networks	9
2.4 Approximate Neurons through Criticality Analysis	16
2.5 FPGA Implementation for Neural Networks	18
<b>3 Background</b>	<b>27</b>

3.1	Convolutional Neural Network and Preliminaries	27
3.1.1	Convolutional Neural Network	27
3.1.2	CNN vs. ANN	27
3.1.3	Convolutional Neural Network Applications	29
3.1.4	Convolutional Neural Network Architecture	30
	Convolution Layer	30
	Pooling Layer	31
	Nonlinear Function	36
	Fully Connected Layer	40
	Setting Number of Layers and Their Sizes	41
	Soft Max Activation Function	43
3.2	Simple Example	43
3.3	DataSets	47
3.3.1	Modified National Institute of Standards and Technology (MNIST)	47
3.3.2	Fashion MNIST (F-MNIST)	47
3.3.3	Street View House Numbers (SVHN)	50
3.3.4	Canadian Institute For Advanced Research (CIFAR-10)	50
<b>4</b>	<b>Proposed Customized Convolutional Neural Network</b>	<b>53</b>
4.1	Proposed Convolutional Neural Network Architecture	53
4.2	Precision Scaled CNN Approximation	57
4.3	Hardware Architecture	60
	Overview	60
	Memory Unit	60
	Computation Unit	61
	MaxPool Units	62
	Memory Access Units	63
	Register File, ReLU Units and Comparator Unit	64
4.4	A field Programmable Gate Array (FPGA)	64
4.5	Dynamic Partial Reconfiguration	66
<b>5</b>	<b>Experimental Results and Discussions</b>	<b>67</b>
5.1	Training Results for CNNs	67
5.2	Test Results for CNNs	72
5.3	Results of Hardware Architecture	72
<b>6</b>	<b>Conclusion and Proposed Future Work</b>	<b>83</b>
6.1	Conclusion	83
6.2	Proposed Future Work	85

# List of Tables

Table 2.1:	Decomposition of multiplicand. . . . .	9
Table 2.2:	The proposed model. . . . .	19
Table 2.3:	CNN accuracy while using different models. . . . .	19
Table 3.1:	ANN vs. CNN . . . . .	28
Table 3.2:	Category of fashion MNIST dataset. . . . .	47
Table 3.3:	Category of CIFAR-10 dataset. . . . .	50
Table 4.1:	Network parameters and number of multiplication for each CNN layer for MNIST, F-MNIST and SVHN datasets. . . . .	55
Table 4.2:	Network parameters and number of multiplication for each CNN layer for CIFAR-10 dataset. . . . .	56
Table 5.1:	CNN accuracy for each dataset using 32-bit floating point number representation. . . . .	72
Table 5.2:	Number of needed bits to represent integer part. . . . .	72
Table 5.3:	CNN accuracy and loss for MNIST, F-MNIST, SVHN and CIFAR- 10 datasets. . . . .	73
Table 5.4:	Accuracy and resource utilization of FPGA for MNIST with different bitwidth, ANN [24] and LENET CNN [26]. . . . .	75
Table 5.5:	Resource utilization of FPGA for F-MNIST with different bitwidth. . . . .	76
Table 5.6:	Resource utilization of FPGA for SVHN with different bitwidth. . . . .	77
Table 5.7:	The recognition time and throughput of three datasets. . . . .	78
Table 5.8:	Power and energy per image consumption of approximated CNN for MNIST dataset . . . . .	78
Table 5.9:	Power and energy per image consumption of approximated CNN for F-MNIST dataset . . . . .	78
Table 5.10:	Power and energy per image consumption of approximated CNN for SVHN dataset . . . . .	79
Table 5.11:	Reconfiguration time for each dataset. . . . .	80
Table 5.12:	Comparison between proposed designs and ANN [24] . . . . .	81

# List of Figures

Figure 2.1:	Intel’s neuromorphic chip Loihi adopted from [3]. . . . .	4
Figure 2.2:	Biological neuron adopted from [8]. . . . .	5
Figure 2.3:	A view of the neuron cells and connections from a very small area of the cortex adopted from [8]. . . . .	6
Figure 2.4:	The neuron as component adopted from [9]. . . . .	8
Figure 2.5:	8-bits 4-alphabet alphabet set multiplier (ASM) adopted from [10].	10
Figure 2.6:	4-alphabet ASMs while using CSHM architecture adopted from [10].	10
Figure 2.7:	8-bits 1-alphabet ASM. adopted from [10]. . . . .	11
Figure 2.8:	Conventional operation of convolution and max pool layers adopted from [12]. . . . .	12
Figure 2.9:	LCP operation adopted from [12]. . . . .	12
Figure 2.10:	The methodology used to approximate neural network with approximate multipliers generated using GCP adopted from [13].	14
Figure 2.11:	Methodology of conditional deep learning network (CDLN) adopted from [16]. . . . .	15
Figure 2.12:	Methodology of approximate neural network (AxNN) adopted from [18]. . . . .	17
Figure 2.13:	Mapping neuron to PE adopted from [19]. . . . .	18
Figure 2.14:	Hardware architecture of artificial neural network adopted from [24]. . . . .	20
Figure 2.15:	Hardware architecture of the convolutional neural network accelerator adopted from [26]. . . . .	21
Figure 2.16:	The architecture of the second design adopted from [30]. . . . .	22
Figure 2.17:	The vision chip architecture adopted from [32] . . . . .	23
Figure 3.1:	ANN architecture. . . . .	28
Figure 3.2:	3D CNN architecture. . . . .	29
Figure 3.3:	CNN architecture. . . . .	30
Figure 3.4:	The convolution layer Operation. . . . .	32
Figure 3.5:	Down sampling of pooling layer. . . . .	33
Figure 3.6:	Max pooling and average pooling operations. . . . .	33
Figure 3.7:	Detecting horizontal line Image . . . . .	35
Figure 3.8:	CNN with and without non-linearity. . . . .	37
Figure 3.9:	Sigmoid activation function. . . . .	38
Figure 3.10:	Tanh activation function. . . . .	39
Figure 3.11:	ReLU activation function. . . . .	39
Figure 3.12:	PReLU activation function. . . . .	40
Figure 3.13:	Neuron at fully connected layer. . . . .	41
Figure 3.14:	Three neural network with different number of neurons. . . . .	42
Figure 3.15:	Softmax activation function. . . . .	43
Figure 3.16:	Simple CNN example:The images to be classified. . . . .	46
Figure 3.17:	Some samples from MNIST dataset. . . . .	48

Figure 3.18:	Some samples from fashion MNIST dataset. . . . .	49
Figure 3.19:	Some samples from SVHN dataset. . . . .	51
Figure 3.20:	Some samples from CIFAR-10 dataset. . . . .	52
Figure 4.1:	EEPS-CNN architecture. . . . .	54
Figure 4.2:	Available $n$ bitwidth decomposed into three parts. . . . .	57
Figure 4.3:	Available $2n$ bitwidth decomposed into three parts. . . . .	57
Figure 4.4:	Example of Convolution Operation. . . . .	58
Figure 4.5:	How neuron is connected with previous neurons at fully connect layer. . . . .	59
Figure 4.6:	Quantization flow. . . . .	59
Figure 4.7:	Block diagram of the proposed CNN architecture . . . . .	61
Figure 4.8:	The architecture of computational unit. . . . .	62
Figure 4.9:	The architecture of the PE. . . . .	62
Figure 4.10:	How $m$ (number of bits for integer part) is adjusted at hardware, the appropriate $m$ is selected according to the layer number. . . . .	63
Figure 4.11:	The connection between memory and MaxPool unit. . . . .	63
Figure 4.12:	The architecture of the MaxPool unit. . . . .	64
Figure 4.13:	Architecture of memory access units. . . . .	65
Figure 4.14:	A simplified model of the Zynq architecture. . . . .	66
Figure 4.15:	DPR system block diagram. . . . .	66
Figure 5.1:	Accuracy and loss model for MNIST dataset. . . . .	68
Figure 5.2:	Accuracy and loss model for F-MNIST dataset. . . . .	69
Figure 5.3:	Accuracy and loss model for SVHN dataset. . . . .	70
Figure 5.4:	Accuracy and loss model for CIFAR-10 dataset. . . . .	71
Figure 5.5:	Normalized accuracy of EEPS-CNN. . . . .	74
Figure 5.6:	Reduction in energy. . . . .	79
Figure 5.7:	Accuracy loss against energy reduction for approximated CNN. . . . .	80

# Listings

3.1	Images definition	34
3.2	CNN definition	35
3.3	Define the filter	35
3.4	Classifying images example	43