

**OPTIMIZATION OF INTRUDER DETECTION ALGORITHM
USING RASPBERRY PI PLATFORM**

MOHD ASYRAF BIN ZULKALNAIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**OPTIMIZATION OF INTRUDER DETECTION
ALGORITHM USING RASPBERRY PI PLATFORM**

MOHD ASYRAF BIN ZULKALNAIN

**This report is submitted in partial fulfilment of the requirements
for the degree of Bachelor of Electronic Engineering with Honours**

**Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka**

2019

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : OPTIMIZATION OF INTRUDER DETECTION
ALGORITHM USING RASPBERRY PI
PLATFORM

Sesi Pengajian : 2018/2019

Saya MOHD ASYRAF BIN ZULKALNAIN mengaku membenarkan laporan
Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat
kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan
pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

SULIT*

(Mengandungi maklumat yang berdarjah
keselamatan atau kepentingan Malaysia
seperti yang termaktub di dalam AKTA
RAHSIA RASMI 1972)

TERHAD*

(Mengandungi maklumat terhad yang
telah ditentukan oleh organisasi/badan di
mana penyelidikan dijalankan.

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: No.35, Jalan Nilai
Impian 3, Nilai,
71800, Negeri
Sembilan

Tarikh : 01 Januari 2010

Tarikh : 01 Januari 2010

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan
dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled “OPTIMIZATION OF INTRUDER DETECTION ALGORITHM USING RASPBERRY PI PLATFORM” is the result of my own work except for quotes as cited in the references.

Signature :

Author :

Date :

APPROVAL

I 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 Bachelor of Electronic Engineering with Honours.

Signature :

Supervisor Name :

Date :

DEDICATION

I dedicate this work to my family, especially my beloved parents whom always been there for me and been a source of my inspiration to study and work and also provide moral, spiritual, emotional and financial support. I also dedicate this work to my brothers and sisters, friends and teachers who shared their guidance and moral support throughout my life.

ABSTRAK

Projek ini melibatkan penggunaan senibina Rangkaian Neural Konvolusi (CNN) untuk membangunkan algoritma pengesanan penceroboh. Pada asasnya, algoritma pengesanan penceroboh melibatkan tugas klasifikasi imej untuk mengklasifikasikan imej input ke penceroboh dan bukan penceroboh. Baru-baru ini, CNN telah menunjukkan mempunyai ketepatan yang besar pada aplikasi klasifikasi imej. Oleh itu, projek ini berdasarkan model LeNet dan MobileNet, di mana kedua-dua model dilatih untuk mengklasifikasikan imej penceroboh dan bukan penceroboh. Selepas pelbagai model DNN telah dilatih, model terbaik dari segi prestasi telah dipindahkan dari Komputer Peribadi (PC) ke Raspberry Pi 3 Model B+. MobileNet telah menunjukkan mempunyai ketepatan yang tinggi di samping mengekalkan kerumitan model yang rendah. MobileNet v3 telah dipilih untuk dipindahkan ke Raspberry Pi kerana ia telah terbukti mempunyai ketepatan tertinggi apabila diuji dengan imej penguji. Apabila fasa port selesai, prestasi dari segi ketepatan dan kelajuan MobileNet v3 di Raspberry Pi dan PC telah dibandingkan dan dinilai. Kelajuan MobileNet di Raspberry Pi adalah sebanyak 2.855 fps manakala di PC sebanyak 6.263 fps secara purata. Kelajuan dari segi fps MobileNet v3 apabila dilaksanakan pada Raspberry Pi adalah 54.41% lebih perlahan dibandingkan dengan PC. Ini menyimpulkan bahawa

model DNN berjalan lebih lambat pada Raspberry Pi berbanding apabila dilaksanakan pada PC.

ABSTRACT

This project involves the use of Convolutional Neural Network (CNN) architecture to develop an intruder detection algorithm. Basically, the intruder detection algorithm involves image classification task to classify input image into intruder and non-intruder. Recently, CNN have shown to have great accuracy on image classification application. Thus, this project is based on LeNet and MobileNet models, where both of the models were trained to classify intruder and non-intruder images. After various DNN models have been trained, the best model in terms of performance were ported from Personal Computer (PC) to Raspberry Pi 3 Model B+ in the deployment phase. MobileNet have shown to have high accuracy while maintaining low model complexity or number of operations. MobileNet v3 was chosen to be ported to Raspberry Pi because it has proven to have highest accuracy when tested with testing images of intruder and non-intruder. When the porting is complete, the performance in terms of accuracy and speed of MobileNet v3 in Raspberry Pi and PC was compared and evaluated. The average speed of MobileNet v3 when running on Raspberry Pi is 2.855 fps while on PC is 6.263 fps. The speed in terms of frames per second (fps) of MobileNet v3 when executed on Raspberry Pi was 54.41% slower compare to PC. This conclude that DNN model runs slower on Raspberry Pi compared to when executed on PC.

ACKNOWLEDGEMENTS

Foremost, I would like to express my gratitude to Allah the Almighty God for letting me breathe air up until now and giving me strength and opportunity to finish this thesis. I also would like to give my appreciation towards my dearest parents, Zulkalnain bin Mohd Yussof and Norlela binti Mohd Nordin for providing me love, utmost moral and financial support.

Also to my supervisor, Dr.Sani Irwan bin Md. Salim, I would like to express my gratitude for the guidance, motivation and knowledge throughout completing my thesis. And not to mention, all of my lecturers and colleagues that have been there by my side through ups and downs.

TABLE OF CONTENTS

Declaration	
Approval	
Dedication	
Abstrak	i
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Figures	ix
List of Symbols and Abbreviations	xiii
List of Appendices	xiv
CHAPTER 1 INTRODUCTION	15
1.1 Background of project	15
1.2 Problem Statement	16
1.3 Objectives	16
1.4 Scope of Project	17
1.5 Thesis Outline	17

CHAPTER 2 THEORY AND BACKGROUND STUDY	18
2.1 Neural network	18
2.1.1 Perceptron	19
2.1.2 Activation function	20
2.2 Convolutional Neural Network (CNN)	21
2.2.1 Convolution layer	22
2.2.2 Rectified Linear Unit (ReLU) layer	24
2.2.3 Pooling layer	25
2.2.4 Fully connected layer	26
2.3 Recent CNN architecture	27
2.3.1 VGGNet	28
2.3.2 GoogleNet	28
2.3.3 ResNet 29	
2.3.4 DenseNet	30
2.3.5 MobileNet	31
2.4 Popular Datasets for Image Classification	33
2.4.1 ImageNet ILSVRC 2012	33
2.4.2 Stanford Dogs	34
2.5 Transfer learning	34
2.6 Implementing CNN in Raspberry Pi	35

2.6.1	Deep Neural Network (DNN) in Raspberry Pi for Face Recognition	36
2.6.2	Analysis of Real Time DNN inference on Raspberry Pi	36
2.6.3	Performance of keyword spotting in Raspberry Pi	39
CHAPTER 3 METHODOLOGY		42
3.1	Overview of Intruder Detection Algorithm	42
3.2	Image Classification using CNN	44
3.2.1	Training dataset	44
3.2.2	Validation dataset	46
3.3	Training phase	46
3.3.1	MobileNet structure	47
3.3.1.1	Width multiplier	49
3.3.1.2	Resolution multiplier	51
3.4	Experiments	52
3.4.1	Fine-tuning LeNet models	52
3.4.2	Fine-tuning MobileNet models	54
3.4.3	Testing phase	55
CHAPTER 4 RESULTS AND DISCUSSION		56
4.1	Performance of various LeNet versions	56
4.2	Performance of various MobileNet versions	63
4.3	Speed comparison between PC and Raspberry Pi	73

CHAPTER 5 CONCLUSION AND FUTURE WORKS	77
5.1 Future works	78
REFERENCES	80
APPENDICES	83

LIST OF FIGURES

Figure 2.1: Layers in a neural network	19
Figure 2.2: Working principle of the perceptron	20
Figure 2.3 : Example of Activation functions	21
Figure 2.4: Layers in a CNN architecture	22
Figure 2.5: Convolution operation in CNN	22
Figure 2.6: Convolution operation with three filters	23
Figure 2.7: Operation of ReLU activation function	24
Figure 2.8: Example of Pooling operation	25
Figure 2.9: The fully connected layers from the network	26
Figure 2.10: Existing layers in the LeNet-5 architecture [14]	27
Figure 2.11: VGGNet architecture	28
Figure 2.12: Inception model with reduced dimension [8]	29
Figure 2.13: High level diagram of ResNet architecture	30
Figure 2.14: High level diagram of DenseNet architecture	30
Figure 2.15: Comparison between standard convolution filter and depthwise convolution filters [1]	31
Figure 2.16: Comparison between different CNN models in terms of accuracy and number of parameters	33

Figure 2.17: Top-5 Accuracy of different models using different frameworks, darker bar shows the measured accuracy, lighter bar shows the reported accuracy	37
Figure 2.18: Mean throughput for each DNN models and different frameworks [15]	38
Figure 2.19: CNN architecture for keyword spotting	39
Figure 3.1: Flow diagram of intruder detection algorithm	43
Figure 3.2: People dataset	45
Figure 3.3: Without people dataset	45
Figure 3.4 : Left: Standard convolution layer with batch normalization and ReLU layer, Right: Depthwise separable convolution with batch normalization and ReLU	48
Figure 3.5: Example of testing images	55
Figure 4.1: LeNet v1 training accuracy	57
Figure 4.2: LeNet v1 Testing Image for People	58
Figure 4.3: LeNet v1 Testing Image for Not People	58
Figure 4.4: LeNet v2 training accuracy with 50 epochs	59
Figure 4.5: LeNet v2 Testing Image for People	59
Figure 4.6: LeNet v2 Testing Image for Not People	60
Figure 4.7: LeNet v3 training accuracy with 50 epochs and an increased number of filters	60
Figure 4.8: LeNet v3 Testing Image for People	61
Figure 4.9: LeNet v3 Testing Image for Not People	61
Figure 4.10: MobileNet v1 training accuracy	63
Figure 4.11: MobileNet v1 Testing Image for People	64
Figure 4.12: MobileNet v1 Testing Image for Not People	64
Figure 4.13: MobileNet v2 training accuracy	65

Figure 4.14: MobileNet v2 Testing Image for People	65
Figure 4.15: MobileNet v2 Testing Image for Not People	66
Figure 4.16: MobileNet v3 training accuracy	66
Figure 4.17: MobileNet v3 Testing Image for People	67
Figure 4.18: MobileNet v3 Testing Image for Not People	67
Figure 4.19: MobileNet v4 training accuracy	68
Figure 4.20: MobileNet v4 Testing Image for People	68
Figure 4.21: MobileNet v4 Testing Image for Not People	69
Figure 4.22: MobileNet v5 training accuracy	69
Figure 4.23: MobileNet v5 Testing Image for People	70
Figure 4.24: MobileNet v5 Testing Image for Not People	70
Figure 4.25: MobileNet v6 training accuracy	71
Figure 4.26: MobileNet v6 Testing Image for People	71
Figure 4.27: MobileNet v6 Testing Image for Not People	72
Figure 4.28: Setup of the project	74
Figure 4.29: MobileNet v3 is executed using Raspberry Pi	74
Figure 4.30: Approximate frames per second (fps) captured on Raspberry Pi	75
Figure 4.31: Approximate frames per second (fps) captured on PC	76

Table 2.1: Comparison between MobileNet and popular models [11]	32
Table 2.2: Performance of various DNN models on Raspberry Pi [16]	40
Table 2.3: Comparison between 3 background studies	41
Table 3.1: Effect of width multiplier to accuracy of network [11]	50
Table 3.2: Effect of resolution multiplier to accuracy of the network [11]	51
Table 3.3: Detailed description of the layers in LeNet v1 architecture	52
Table 3.4: Model configurations for LeNet v1, LeNet v2 and LeNet v3	53
Table 3.5: Model configurations for MobileNet-128 v1, v2, v3, v4, v5 and v6	54
Table 4.1: Comparison of accuracy of various LeNet version	62
Table 4.2: Comparison of accuracy of various MobileNet version	72
Table 4.3: Frames per second captured on Raspberry Pi	75
Table 4.4: Frames per second captured on PC	76

LIST OF SYMBOLS AND ABBREVIATIONS

CNN	:	Convolutional Neural Network
PC	:	Personal Computer
GPU	:	Graphical Processing Units
DNN	:	Deep Neural Network
CNN	:	Convolutional Neural Network
ANN	:	Artificial Neural Network
MLP	:	Multi-layer perceptron
ReLU	:	Rectified Linear Unit
COCO	:	Common Object in Context
MACs	:	Multiply-Accumulates
ILSVRC	:	ImageNet Large Scale Visual Recognition Competition
Wnid	:	WordNet Identification
ONNX	:	Open Neural Network Exchange
API	:	Application Programming Interface
FTP	:	File Transfer Protocol
RAM	:	Random Access Memory
VPU	:	Vision Processing Unit
ASIC	:	Application-specific Integrated Circuits
TPU	:	Tensor Processing Unit

LIST OF APPENDICES

Appendix A: Training	82
Appendix B: Testing	87
Appendix C: Running trained DNN	88

CHAPTER 1

INTRODUCTION

Convolutional neural network (CNN) is a class of deep neural network that is commonly used to analyze visual images. CNN architecture consists of weights and biases that are learnable in deep learning. There are many applications of CNN such as face detection and recognition, gender recognition, object detection, and other common computer vision tasks.

1.1 Background of project

This project involves the use of CNN architecture to develop an intruder detection algorithm. Basically, the intruder detection algorithm involves image classification task to classify input image into intruder and non-intruder. Recently, CNN have shown to have great accuracy on image classification application. Thus, this project is based

on LeNet and MobileNet models, where both of the models were trained to classify intruder and non-intruder images. After both models have been trained, they were ported from Personal Computer (PC) to Raspberry Pi in the deployment phase. When the porting is complete, the accuracy and losses of both of the networks were evaluated.

1.2 Problem Statement

The main problem is that CNN model requires high complexity in terms of number of operations, thus computationally expensive. To efficiently process the high complexity of CNN models, a very powerful Graphical Processing Units (GPU) is required. In order to implement the CNN model on an embedded and resource constrained system like Raspberry Pi, the CNN model must be optimized so that it will operate using lesser number of operation or complexity while maintaining high accuracy on the Raspberry Pi. More importantly, Raspberry Pi has limited computing power and memory storage to store and run the CNN model. Thus, the CNN model can be optimized experimentally by reducing the computations in the neural network experimentally so that it can fit into Raspberry Pi platform.

1.3 Objectives

The following statements are the objectives of this project:

- To train LeNet and MobileNet models for image classification.
- To implement optimized deep neural network (DNN) on Raspberry Pi.

- To evaluate the performance of deep neural networks on Raspberry Pi in terms of accuracy and speed (frames per second)

1.4 Scope of Project

The scope of this project is to classify input images into intruder and non-intruder using two existing CNN models; LeNet and MobileNet. Both of the CNN models were written in Python language and uses TensorFlow 1.9 and Keras 2.2 framework. The version of Python is 3.5.2. Various versions of LeNet and MobileNet were trained and all of the networks' accuracy were evaluated. Training phase was assisted by using two GPUs; NVIDIA GeForce GTX 970 and Tesla K70c. The best optimized CNN model was selected and then ported to Raspberry Pi 3 Model B+. The accuracy and speed (frames per second) of the CNN model on Raspberry Pi was evaluated.

1.5 Thesis Outline

This thesis consists of five chapters that includes Chapter 1: Introduction, Chapter 2: Theory and Background Study, Chapter 3: Methodology, Chapter 4: Results and Discussion and Chapter 5: Conclusion and Future works. Chapter 1 briefly states the background of project, problem, objectives, and scope of the project. Chapter 2 discusses theories related to Neural Network, recent CNN architecture and several background studies. Chapter 3 shows the method of this project while Chapter 4 depicts the result and analysis. Chapter 5 concludes the project and states the future works.

CHAPTER 2

THEORY AND BACKGROUND STUDY

This chapter covers the theories that are used to understand and train a CNN model. Large deep neural networks like VGGNet, GoogleNet, ResNet, DenseNet and MobileNet were also described briefly. Several background studies were conducted to study a few applications of CNN and the optimization of the DNN models into Raspberry Pi platform.

2.1 Neural network

Neural network or Artificial Neural Networks (ANN) is a computational brain-like model inspired from the way of how humans learn. A neural network has interconnected artificial neurons that transmit data among each other called nodes. The neural network consists of three layers: Input layer, hidden layer and output layer. Figure 2.1 shows the layers in neural network.