

LB/TH/15/2026
TH6171

**IMPLEMENTATION OF DEEP NEURAL
NETWORKS ON EMBEDDED SYSTEMS FOR
REAL-TIME HAND GESTURE RECOGNITION
USING A 24 GHz DOPPLER RADAR**

M.A.M. Ijaz

219436B

MSc in Telecommunication

Department of Electronic and Telecommunication Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2025

**IMPLEMENTATION OF DEEP NEURAL
NETWORKS ON EMBEDDED SYSTEMS FOR
REAL-TIME HAND GESTURE RECOGNITION
USING A 24 GHz DOPPLER RADAR**

M.A.M. Ijaz

219436B

Thesis submitted in partial fulfillment of the requirements for the degree
MSc in Telecommunication

Department of Electronic and Telecommunication Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2025

DECLARATION

I declare that this is my own work and this Thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: M.A.M Ijaz  Digitally signed
by M.A.M Ijaz

Date: 13/06/2025

The supervisors should certify the Thesis with the following declaration.

The above candidate has carried out research for the MSc in Telecommunication Thesis under our supervision. We confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Chamira Edussooriya

Signature of the Supervisor:  Digitally signed
by Chamira U.
S. Edussooriya

Date: 13/06/2025

Name of Supervisor: Eng. Kithsiri Samarasinghe

Signature of the Supervisor:  Digitally signed
by A.T.L.K. Samarasinghe
2025.06.13
10:22:18
+05'30'
2025.1.0

Date: 13/06/2025

DEDICATION

To everyone who helped us make this project a success.

ACKNOWLEDGEMENT

I express my sincere gratitude to our project supervisors, Dr. Chamira Edussooriya and Eng. Kithsiri Samarasinghe from ENTC, University of Moratuwa, for guiding me through the project and encouraging me to reach the milestones that i have reached along my journey to make this project a success.

Then i would like to express my gratitude to the members of the previous year final year project group Mr. Gayangana, Mr. Kavindu , Mr. Nisal and Mr. Dhanuka for their support and guidance they have provided.

I would also extend my gratitude to all the staff members of the Department of Electronic and Telecommunication Engineering who helped me by providing valuable suggestions. My family and friends for the support and motivation provided throughout the project.

Last but not least, I would like to extend my thanks to my employer, Sri Lanka Telecom PLC and its management for supporting me throughout the journey and partially financing my studies.

ABSTRACT

Real-time hand gesture recognition is a key component of next-generation human-computer interaction, with applications in smart environments, assistive technologies, and contactless control systems. This research uses deep neural networks in embedded systems for radar-based gesture recognition, where the gesture signals are captured using a 24 GHz Doppler radar sensor, and the processing is done in the embedded edge computing resources.

The project uses two types of deep learning models: a vision transformer (ViT) that captures long-range components for superior classification and a deep convolutional neural network (DCNN) that captures local spatial features. This work uses hyperparameter optimization, model pruning, and integer quantization to increase accuracy and efficiency, but with minimal loss, to achieve real-time performance on low-power hardware. Hyperparameter optimization is used to improve model architecture performance. In order to achieve better performance and lower resource consumption, pruning is employed. The integer quantization shrinks the size of the deep learning models for lower-power devices with an increased inference speed.

The proposed system is deployed on an embedded platform while ensuring low latency, reduced power consumption, and real-time inference. The deployment results validate the feasibility of radar-based gesture recognition in embedded devices. The experiments compare the optimized DCNN and ViT models, evaluating their speed, memory usage, accuracy, and energy efficiency. The results highlight the trade-offs between accuracy and efficiency, providing valuable insight into the feasibility of deep learning for real-time radar-based gesture recognition on embedded devices. This research contributes to the advancement in the deployment of deep learning models in embedded devices.

Keywords: Hand Gesture Recognition, Vision Transformers, Deep Convolutional Neural Networks, Deep Learning Model Optimization, Embedded Systems

TABLE OF CONTENTS

Declaration of the Candidate & Supervisor	i
Dedication	ii
Acknowledgement	iii
Abstract	iv
Table of Contents	v
List of Figures	ix
List of Tables	xi
List of Abbreviations	xi
List of Appendices	xiii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Proposed Solution	2
1.4 Scope and Expected Deliverables	3
1.5 Contribution of the Dissertation	3
1.6 Organization of the Dissertation	4
2 LITERATURE REVIEW	5
2.1 Radar Systems and Micro-Doppler Signatures	5
2.2 Deep Learning Algorithms for Hand Gesture Recognition	6
2.3 Deep Learning Model Optimization	7
2.4 Embedded System Implementation	8
3 METHODOLOGY	10
3.1 System Architecture	10
3.2 Doppler Transceiver	11
3.3 Signal processing	12
3.4 Data Sets	13
3.4.1 Data Set : 14 Hand Gestures	13

3.4.2	Data Set : 6 Hand Gestures	15
3.5	Deep Learning Model	16
3.5.1	Vision Transformer Architecture	16
3.5.2	Deep Convolutional Neural Networks Architecture	18
3.6	Optimization of Deep Learning Model	19
3.6.1	Hyperparameters Tuning	19
3.6.2	Model Pruning	21
3.6.3	Model Quantization	22
3.7	Model Parameters	23
3.8	Building the Model	24
3.8.1	Training Model	24
3.8.2	Converting Model to TFLite	25
3.9	Storing / Loading Models	26
3.9.1	Storing Complete Trained Model	26
3.9.2	Storing Trained Model Weights	26
3.10	Data Prepossessing	27
3.11	Embedded System	28
3.11.1	Embedded System Development Environment	29
3.11.2	Embedded System Minimum Requirements	29
3.11.3	Cost-Benefit Analysis and Selecting an Embedded System	30
3.12	Generating C Files	31
3.13	Development Environment Setup	32
3.14	The Embedded System Application	34
3.15	Debugging the Embedded System	35
3.16	Energy Measurement	36
4	RESULTS	37
4.1	Terminology and Metrics	37
4.1.1	Confusion Matrix	37
4.1.2	Accuracy	37
4.1.3	Precision	38
4.1.4	Recall	38

4.1.5	F1-Score	38
4.1.6	Specificity	38
4.2	Deep Learning Model Training Results	39
4.2.1	DCNN Model	39
4.2.2	ViT Model	40
4.3	Deep Learning Model Quantization Results	41
4.3.1	DCNN Model Size Comparison	41
4.3.2	ViT Model Size Comparison	41
4.3.3	DCNN Model with Float Weights	42
4.3.4	DCNN Model with 8 bit Integer Weights	43
4.3.5	ViT Model with Float Weights	44
4.3.6	ViT Model with 8 bit Integer Weights	45
4.4	Deep Learning Model Pruning Results	46
4.4.1	Sparsity of Model	46
4.4.2	Sparsity DCNN Model Size Comparison	46
4.4.3	DCNN Model Pruned with Float Weights	47
4.4.4	DCNN Model Pruned with 8 bit Integer Weights	48
4.5	Deep Learning Model Inference Time	49
4.5.1	Emulation Results	49
4.5.2	Implementation Results	50
4.6	Energy Measurement Results	52
4.6.1	DCNN Model Energy Consumption	52
4.6.2	ViT Model with Encode and Decoder Energy Consumption	53
4.7	Discussion	54
4.7.1	Deep Learning Model	54
4.7.2	Deep Learning Model Hyperparameter Tuning	54
4.7.3	Deep Learning Model Pruning	54
4.7.4	Deep Learning Model Quantization	54
4.7.5	Model Inference Time	55
4.7.6	Model Power Consumption	55
4.7.7	Impact of the Project	55

4.7.8 Possible Applications	55
5 CONCLUSION AND FUTURE WORKS	56
References	57
Appendix A Embedded System Comparison	60
Appendix B Energy Measurement : Selected Components	61
B.1 Power Monitor IC	61
B.2 Power Monitor Controller	61
B.3 Power Source	62
Appendix C DCNN Model	63
Appendix D ViT Model	65
Appendix E Power Measurement Controller	72

LIST OF FIGURES

Figure	Description	Page
Figure 3.1	System Architecture	10
Figure 3.2	Application Diagram	11
Figure 3.3	Data Set 1 - Image of the gestures used	14
Figure 3.4	Data Set 2 - Swipe Diagonally Doppler Signature	15
Figure 3.5	ViT with Convolutional Encoder Decode	16
Figure 3.6	ViT Attention Module	17
Figure 3.7	DCNN Architecture	18
Figure 3.8	Sparsity Pruning	21
Figure 3.9	Quantization	22
Figure 3.10	Google Colab	24
Figure 3.11	TFLite Conversion	25
Figure 3.12	Embedded Systems	28
Figure 3.13	ST Edge AI Developer Cloud	31
Figure 3.14	STM32Cube-Board	32
Figure 3.15	STM32Cube-Settings	33
Figure 3.16	Debugging the Embedded System	35
Figure 3.17	Energy Measurement Setup	36
Figure 4.1	Accuracy-DCNN	39
Figure 4.2	Accuracy-ViT	40
Figure 4.3	Confusion Matrix-DCNN-Float	42
Figure 4.4	Confusion Matrix-DCNN	43
Figure 4.5	Confusion Matrix-ViT-Float	44
Figure 4.6	Confusion Matrix-ViT	45
Figure 4.7	Sparsity vs Epochs	46
Figure 4.8	Confusion Matrix-DCNN-Pruned	47
Figure 4.9	Confusion Matrix-DCNN-Pruned-Quantized	48
Figure 4.10	Performance Summary Chart	49
Figure 4.11	DCNN Inference Time	50
Figure 4.12	ViT Inference Time	51
Figure 4.13	DCNN Model Energy Consumption	52
Figure 4.14	ViT Model with Encode and Decoder Energy Consumption	53
Figure A.1	ARM Processor Class	60
Figure B.1	Power Monitor IC - INA226	61
Figure B.2	Power Monitor Controller - Arduino Nano 33 BLE Sense	61

LIST OF TABLES

Table	Description	Page
Table 3.1	Deep Learning Models Comparison	16
Table 3.2	ViT Model - Encoder Filter Specification	17
Table 3.3	DCNN Model - Filter Specification	19
Table 3.4	DCNN Hyperparameters Tuning	20
Table 3.5	ViT Hyperparameters Tuning	20
Table 3.6	Software Packages	24
Table 3.7	Data Prepossessing	27
Table 3.8	Embedded Systems Comparison	30
Table 4.1	DCNN Model Size	41
Table 4.2	ViT Model Size	41
Table 4.3	DCNN Model with Float Weights	42
Table 4.4	DCNN Model with 8 bit integer Weights	43
Table 4.5	ViT Model with Float Weights	44
Table 4.6	ViT Model with 8 bit integer Weights	45
Table 4.7	DCNN Model Size before and after Pruning	46
Table 4.8	Confusion Matrix-DCNN-Pruned	47
Table 4.9	Confusion Matrix-DCNN-Pruned-Quantized	48
Table 4.10	DCNN Model Energy Consumption	52
Table 4.11	ViT Model with Encode and Decoder Energy Consumption	53

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	artificial intelligence
BLSTM	bidirectional long short-term memory
CNN	convolutional neural network
CTC	connectionist temporal classification
CW	continuous wave
DCNN	deep convolutional neural network
DL	deep learning
DT	decision tree
FFT	fast fourier transform
FMCW	frequency modulated CW
FPU	floating point unit
HAL	hardware abstraction layer
HCI	human-computer interaction
HGR	hand gesture recognition
HMM	hidden markov model
HMR	human motion recognition
IoT	internet of things
LO	local oscillator
MDS	micro Doppler signatures
MLP	multi-layer perceptron
MSENet	multiscale squeeze-and-excitation network
MViT	multiscale vision transformers
NLP	natural language processing
PTQ	post-training quantization
QAT	quantization-aware training
SFCW	single frequency CW
STFT	short time fourier transform
SVM	support vector machine
TFLite	tensorflow lite
ViT	vision transformer

LIST OF APPENDICES

Appendix	Description	Page
Appendix -A	Embedded System Comparison	60
Appendix -B	Energy Measurement : Selected Components	61
Appendix -C	DCNN Model	63
Appendix -D	ViT Model	65
Appendix -E	Power Measurement Controller	72