

VEHICLE CRASH PREDICTION USING TRANSFORMER NETWORKS

Don Kalindu Sekarage

218541J

Master of Science in Artificial Intelligence

Department of Computation Mathematics
Faculty of Information Technology

University of Moratuwa
Sri Lanka

June 2024

VEHICLE CRASH PREDICTION USING TRANSFORMER NETWORKS

Don Kalindu Sekarage

218541J

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree

Master of Science in Artificial Intelligence

Department of Computation Mathematics

Faculty of Information Technology

University of Moratuwa

Sri Lanka

June 2024

DECLARATION

I declare that this is my own work and this thesis/dissertation 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.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date: 30-06-2024

The above candidate has carried out research for the Masters/MPhil/PhD thesis/dissertation under my supervision.

Name of Supervisor: Dr.A.L.A.R.R Thanuja

30/06/2024

Signature of the Supervisor:

Date:

DEDICATION

This thesis is dedicated to the notable individuals who have offered unwavering support and encouragement throughout my academic pursuits. I acknowledge with profound gratitude the essential role they have played in my journey.

I extend my deepest appreciation to my beloved family, whose relentless support and faith in my potential have pushed me forward. Your sacrifices and consistent encouragement have laid the foundation for my success. I am eternally thankful for your unwavering presence in my life.

ACKNOWLEDGEMENT

I deeply appreciate the support and guidance from those who helped me complete this thesis. Their constant encouragement and valuable contributions have been essential, and I am sincerely thankful for their involvement.

At the forefront, I owe a debt of gratitude to my supervisor, Dr. Romesh Thanuja, whose invaluable guidance, encouragement, and support have been critical throughout my research journey. His profound insights and expertise significantly influenced the direction and enhanced the quality of this research.

Additionally, I thank the academic staff at the Faculty of Information Technology, University of Moratuwa. Their rigorous standards and passionate teaching have inspired me to improve my critical thinking and advance my skills as a Master's student. Their enthusiasm for their fields has continually motivated my academic endeavors.

ABSTRACT

Road accidents pose a significant threat to human life, causing numerous injuries, fatalities, and economic damage worldwide. Recently, there has been growing interest in leveraging Artificial Intelligence (AI) to create systems that can predict vehicle crashes. This thesis focuses on vehicle crash prediction and aims to develop a solution using pre-trained Convolutional Neural Networks (CNN) and transformer networks to mitigate the occurrence of such accidents. By leveraging advanced deep learning techniques, this research addresses the limitations of traditional crash analysis methods. The objectives of this study involve critically reviewing the evolution of vehicle crash prediction systems, studying modern technologies employed in vehicle crash prediction, designing and implementing a crash prediction system using pre-trained CNN and transformer networks, and evaluating the solution's effectiveness. The motivation behind this research stems from the urgency to reduce the growing frequency of vehicle crashes and their associated consequences. Current approaches to road safety primarily rely on reactive measures, prompting the need for proactive strategies to prevent accidents. The Car Learning to Act (CARLA) simulator was used for data gathering, in that an ego-vehicle attached to RGB and RGB-Depth cameras were used. Ego-vehicle traveled through a simulated environment with different weather conditions and maps and captured a sequence of RGB and RGB-Depth (RGB-D) images in safe and unsafe scenarios. To extract features from RGB and RGB-D images pre-trained CNN were utilized. Four pre-trained CNNs were used for feature extraction. With those extracted features, a transformer network was employed to train a model. After model training and testing, it was observed that the transformer model trained with VGG16-based feature extraction performs better than other methods. This research can be implemented in the real world with real-time predictions and addresses the shortcomings of visual-based systems. Ultimately, it aims to improve road safety by enabling proactive crash prevention and minimizing the human and economic costs associated with accidents.

Keywords: CNN, Transformer Network, CARLA, Feature Extraction

TABLE OF CONTENTS

Declaration	i
Dedication	ii
Abstract	iv
List of Figures	viii
List of Tables	x
List of Abbreviations.....	xi
List of Appendices.....	xii
Chapter 1	1
Introduction	1
1.1 Prolegomena	1
1.2 Objectives	2
1.3 Background and Motivation	2
1.4 Problem in Brief.....	3
1.5 CrashGuard Solution for Vehicle Crash Prediction	4
1.6 Resource Requirements	4
1.7 Structure of the Thesis	4
1.8 Summary	5
Chapter 2	6
Evolution of Vehicle Crash Prediction Systems.....	6
2.1 Introduction.....	6
2.2 Gestation of vehicle crash prediction systems	6
2.3 Major developments in vehicle crash prediction systems	7
2.4 Attention-based autonomous driving	9
2.5 Use of RGB and RGB Depth Camera	9
2.6 Future Directions of The Vehicle Crash Prediction Systems.....	10
2.7 Summary of Challenges in the Vehicle Crash Prediction Systems.....	10
2.8 Problem Definition	11
2.9 Summary	12
Chapter 3	13
Technology adapted	13
3.1 Introduction.....	13
3.2 CARLA Simulator	13
3.2.1 Unreal Engine.....	14
3.2.2 Collision detector	14
3.2.3 RGB Camera	14
3.2.4 RGB Depth Camera	15

3.3 Programming Language.....	15
3.3.1 CARLA Python API.....	15
3.3.2 TensorFlow.....	15
3.4 Cloud Computing.....	16
3.5 CNN.....	16
3.5.1 Convolutional Layer.....	17
3.5.2 Pooling Layer.....	18
3.5.3 ReLU Activation Function.....	18
3.5.4 Fully Connected Layer.....	19
3.5.5 SoftMax Activation Function.....	19
3.6 Pre-trained CNN Models for Feature Extraction.....	19
3.6.1 VGG16.....	20
3.6.2 MobileNet.....	21
3.6.3 ResNet50.....	22
3.6.4 DenseNet121.....	23
3.7 Transformer Network.....	24
3.7.1 Encoder Block Input.....	25
3.7.2 Creating Positional Encoding.....	25
3.7.3 Multi-Head Attention Module.....	26
3.7.4 Add and Normalization.....	28
3.7.5 Decoder Block.....	30
3.7.6 Encoder-Only Models.....	30
3.8 Summary.....	31
Chapter 4.....	32
An Approach to Vehicle Crash Prediction.....	32
4.1 Introduction.....	32
4.2 Hypothesis.....	32
4.3 Input.....	32
4.4 Output.....	33
4.5 Process.....	33
4.6 Features.....	34
4.7 Users.....	34
4.8 Summary.....	35
Chapter 5.....	36
Design of CRASHGUARD.....	36
5.1 Introduction.....	36
5.2 Top-level Architecture.....	36
5.3 Data Collection From CARLA.....	36
5.4 Data Pre-Processing.....	38
5.5 Feature Extraction with Pre-Trained Network.....	38
5.6 Training Model with Transformer.....	39
5.7 Evaluation.....	41

5.8 Summary	41
Chapter 6	42
Implementation of CRASHGUARD.....	42
6.1 Introduction.....	42
6.2 Installing CARLA Simulator	42
6.2 Data Collection	42
6.3 Data Pre-processing	43
6.4 Feature Extraction and Concatenation.....	44
6.5 Data Training Using Transformer Network	44
6.6 Creating Demo Video	45
6.7 Summary.....	45
Chapter 7	46
Evaluation of CRASHGUARD.....	46
7.1 Introduction.....	46
7.2 Transformer Setup.....	46
7.3 Classification Metrics	47
7.3.1 Precision.....	47
7.3.2 Recall.....	48
7.3.3 F1-Score	48
7.4 Evaluation of Feature Extraction Methods	48
7.4.1 Transformer Network with VGG16	49
7.4.2 Transformer Network with MobileNet.....	50
7.4.3 Transformer Network with ResNet50	51
7.4.4 Transformer Network with DenseNet121	53
7.5 Result Comparison with Other Research.....	55
7.6 Summary.....	56
Chapter 8	57
Conclusion and Further Work	57
8.1 Conclusion	57
8.2 Limitations	58
8.3 Further work	58
References	59

LIST OF FIGURES

Figure	Description	Page
Fig. 3.1	CNN Architecture [33]	16
Fig. 3.2	Example of Convolution Operation [34]	17
Fig. 3.3	Example of Max Pooling and Average Pooling [34]	18
Fig. 3.4	Activation Function ReLU [35]	18
Fig. 3.5	Activation Function SoftMax [35]	19
Fig. 3.6	Architecture of VGG16 [37]	20
Fig. 3.7	Architecture of MobileNet [39]	22
Fig. 3.8	Architecture of ResNet50 [40]	23
Fig. 3.9	Architecture of DenseNet121 [41]	24
Fig. 3.10	Transformer Model Architecture [43]	24
Fig. 3.11	Input Embedding for Each Image	25
Fig. 3.12	Combining Input with Positional Encoding	26
Fig. 3.13	Creating, Q,K and V	26
Fig. 3.14	Scaled dot-product (on the left). Multi-Head Attention with Simultaneous Levels(on the right) [20]	27
Fig. 3.15	The Dot Product of Q and KT	27
Fig. 3.16	Final Stage of the Encoder Block	30
Fig. 4.1	Input Image Sequence	33
Fig. 5.1	Top-level Architecture of the Design	36
Fig. 5.2	Different Maps	37
Fig. 5.3	Different Weather Conditions	37
Fig. 5.4	Feature Extraction and Concatenation	39
Fig. 5.5	Encoder Block of the Transformer [43]	40
Fig. 7.1	Accuracy and Loss Graphs for Training vs Validation for VGG16 Based Feature Extraction	49
Fig. 7.2	Model Confusion Matrix for VGG16 Based Feature Extraction	50
Fig. 7.3	Accuracy and Loss Graphs for Training vs Validation for MobileNet Based Feature Extraction	51
Fig. 7.4	Model Confusion Matrix for MobileNet Based Feature Extraction	51
Fig. 7.5	Accuracy and Loss Graphs for Training vs Validation for ResNet50 Based Feature Extraction	52

Fig. 7.6	Model Confusion Matrix for ResNet50 Based Feature Extraction	53
Fig. 7.7	Accuracy and Loss Graphs for Training vs Validation for DenseNet121 Based Feature Extraction	53
Fig. 7.8	Model Confusion Matrix for DenseNet121 Based Feature Extraction	54

LIST OF TABLES

Figure	Description	Page
Table 5.1	CNN Models and No of Extracted Features	39
Table 7.1	Summary of Transformer Parameters	47
Table 7.2	Summary of Performance Metrics for Each Feature Extraction Method	55

LIST OF ABBREVIATIONS

Abbreviation	Description
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
RGB-D	RGB Depth
AI	Artificial Intelligence
SMOTE	Synthetic Minority Over-sampling Technique
RF	Random Forest
KNN	K-Nearest Neighbors
RNN	Recurrent Neural Network
SVM	Support Vector Machines
ReLU	Rectified Linear Unit
ML	Machine Learning
CARLA	Car Learning to Act]
GDP	Gross Domestic Production
CPN	Crash Prediction Network
NLP	Natural Language Processing

LIST OF APPENDICES

Figure	Description	Page
Appendix – A	Script for Importing Necessary Libraries, Setting Basic Configuration, and Connect with CARLA Simulator	61
Appendix – B	CarlaSession Class with add_vehicles Method	61
Appendix – C	Method to Change Weather Settings and Add Pedestrians	62
Appendix – D	Method to Spawn Ego-vehicle and Attach Sensors to It	62
Appendix – E	Method to Trigger Crash	63
Appendix – F	Methods to Run Simulation and Collet RGB and RGB-D Images	63
Appendix – G	RGB and RGB-D Image Preprocessing	64
Appendix – H	Importing Necessary Libraries and Defining CNN Model	64
Appendix – I	Script for Creating Extracted Feature Array	65
Appendix – J	Script for Transformer Positional Embedding	65
Appendix – K	Script for Transformer Encoder	66
Appendix – L	Utility Functions for Training	67
Appendix – M	Script for Creating a Demo Video	68