



Faculty of Computer Science and Information Technology

**Deep Learning Framework for Automated Vehicle Classification and
Counting Using Road Energy-Harvesting Sensors Data**

Amir Raza

**Master of Science
2025**

Deep Learning Framework for Automated Vehicle Classification and Counting Using Road Energy-Harvesting Sensors Data

Amir Raza

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Artificial Intelligence)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.



.....

Signature

Name: Amir Raza

Matric No.: 24020032

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 10/10/2025

ACKNOWLEDGEMENT

A handful gratitude to ALLAH (SWT), the most gracious and the most merciful Whose guidance and protection throughout my entire life and makes my educational career a reality with enormous success. Special gratitude goes to my supervisor Dr. Johari Bin Abdullah for his proper supervision, encouragement, motivation, fatherly advice, and support throughout my study period at Universiti Malaysia Sarawak. His knowledge and expertise in Network Security will always remain with me as guidance to follow. Without his guidance, this research work would have been impossible. I would also like to thank my co-supervisors, Dr. Rehman Ullah for his support and sincere advice.

My most profound gratitude goes to my parents, my brother and sister and my entire family members and friends for their sense of understanding, support, and prayers.

My sincere gratitude to the Centre for Graduate Studies, for the guidance and support given during my period of study in Universiti Malaysia Sarawak.

Finally, I would like to thank the management of Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak for giving me opportunity me to complete my study here in Sarawak. Thank you all.

ABSTRACT

Traffic congestion and ineffective vehicle monitoring systems remain pressing challenges in urban transportation which is leading to time delays, accidents, and reduced road efficiency. Conventional approaches such as manual counting, inductive loop sensors, and traditional image processing often lack accuracy, scalability, and adaptability to real-world conditions. These limitations create a research gap for automated, reliable, and intelligent solutions. This study proposes a deep learning based framework for automated vehicle classification and counting using convolutional neural networks (CNNs). Using piezoelectric sensors, a numerical data of road traffic is collected, pre-processed, and augmented to enhance data quality and reduce overfitting. The 1D-CNN model is trained to classify multiple vehicle categories and integrated into a counting mechanism. Model performance is evaluated using standard metrics, including accuracy, precision, recall, F1-score. Achieving high classification accuracy across diverse traffic conditions, these experimental results show that the proposed framework significantly outperforms conventional machine learning and image processing methods. This research contributes both academically and practically by providing: (1) a CNN-based methodology tailored for vehicle detection and counting, (2) a benchmark dataset for evaluation, and (3) an effective framework that can support intelligent transportation systems (ITS).

Keywords: Vehicle classification, vehicle counting, 1D convolutional neural network (1D-CNN), road energy-harvesting sensors, intelligent transportation system (ITS)

***Pengelasan Automatik dan Pengiraan Kenderaan Menggunakan Pembelajaran
Mendalam Pendekatan untuk Pengurusan Jalan dan Pengangkutan***

ABSTRAK

Kesesakan lalu lintas dan sistem pemantauan kenderaan yang tidak berkesan kekal menjadi cabaran utama dalam pengangkutan bandar yang membawa kepada kelewatan masa, kemalangan, dan pengurangan kecekapan jalan. Pendekatan konvensional seperti pengiraan manual, sensor gegelung induktif, dan pemprosesan imej tradisional sering kali kurang tepat, tidak berskala, dan kurang menyesuaikan diri dengan keadaan sebenar. Kelemahan ini mewujudkan jurang penyelidikan untuk penyelesaian automatik, boleh dipercayai, dan pintar. Kajian ini mencadangkan satu rangka kerja berasaskan pembelajaran mendalam untuk pengelasan dan pengiraan kenderaan secara automatik menggunakan rangkaian neural konvolusi (CNN). Dengan menggunakan sensor piezoelektrik, data berangka trafik jalan dikumpul, dipra-proses, dan diperkaya untuk meningkatkan kualiti data dan mengurangkan overfitting. Model 1D-CNN dilatih untuk mengklasifikasikan pelbagai kategori kenderaan dan digabungkan ke dalam mekanisme pengiraan. Prestasi model dinilai menggunakan metrik standard, termasuk ketepatan (accuracy), ketelitian (precision), pemanggilan semula (recall), dan skor F1 (F1-score). Dengan mencapai ketepatan pengelasan yang tinggi dalam pelbagai keadaan trafik, keputusan eksperimen ini menunjukkan bahawa rangka kerja yang dicadangkan mengatasi kaedah pembelajaran mesin dan pemprosesan imej konvensional dengan ketara. Penyelidikan ini menyumbang dari segi akademik dan praktikal dengan menyediakan: (1) metodologi berasaskan CNN yang disesuaikan untuk pengelasan dan pengiraan kenderaan, (2) set data penanda aras untuk penilaian, dan (3) rangka kerja berkesan yang dapat menyokong sistem pengangkutan pintar (ITS).

Kata kunci: *Pengelasan kendaraan, pengiraan kendaraan, rangkaian neural konvolusi (1D-CNN), sensor penuaian tenaga jalan raya, sistem pengangkutan pintar (ITS)*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
ABSTRAK	v
TABLE OF CONTENTS	vii
LIST OF TABLES	xv
LIST OF FIGURES	xvi
LIST OF ABBREVIATIONS	xviii
CHAPTER 1: INTRODUCTION	1
1.1 Study Background	1
1.2 Problem Statement	5
1.3 Research Objectives	7
1.4 Research Questions	8
1.5 Research Hypothesis	8
1.6 Idea and Motivation	10
1.7 Scope of Study	10
1.8 Significance of Work	10
1.9 Chapter Summary	11

CHAPTER 2: LITERATURE REVIEW	12
2.1 Introduction	12
2.2 Overview of Road Transportation Management	12
2.3 Conventional Methods for Vehicle Classification and Counting	13
2.4 Mechanical Methods for Vehicle Classification and Counting	14
2.4.1 Piezoelectric Sensors	15
2.4.2 Acoustic Sensors	16
2.4.3 Mechanical counting gates	17
2.4.4 Integration with Deep Learning Approaches	18
2.5 Application for Artificial Intelligence in Transportation Management	19
2.5.1 Traffic analysis	19
2.5.2 Vehicle Detection	20
2.5.3 Vehicle Classification	21
2.6 Deep Learning Techniques for Vehicle Detection and Classification	23
2.6.1 Convolutional Neural Networks (CNNs)	23
2.6.2 Applications of Convolutional Neural Networks (CNNs)	25
2.6.3 Recurrent Neural Networks (RNNs)	25
2.6.4 Application of Recurrent Neural Networks (RNNs)	26
2.7 Related Work	29
2.7.1 Conventional Methods	29

2.7.2	Digital Techniques	29
2.7.3	Comparative Analysis	30
2.7.4	Data Quality and Accuracy	30
2.7.5	Challenges of Traditional Techniques	31
2.7.6	Challenges of Digital Techniques	32
2.7.7	Solutions	32
2.8	Summary	48
CHAPTER 3 METHODOLOGY		49
3.1	Introduction	49
3.2	Research Design	49
3.3	Constructing Dataset from Road Energy Harvesting System	51
3.3.1	Data Collection	52
3.4	Data Preprocessing	52
3.4.1	Data Cleaning	53
3.4.2	Normalization	53
3.4.3	Segmentation	53
3.4.4	Label Encoding	54
3.5	Baseline Study	54
3.6	1D- CNN Model Architecture	56
3.6.1	Input Layer	57

3.6.2	Convolutional layers	57
3.6.3	Pooling Layers	59
3.6.4	Fully Connected Layers	59
3.6.5	Output layer	60
3.7	Model Training	61
3.7.1	Data Split	61
3.7.2	Loss Function	62
3.7.3	Optimizer	63
3.7.4	Epochs	63
3.7.5	Batch Size	63
3.8	Model Evaluation	64
3.8.1	Accuracy	64
3.8.2	Precision	65
3.8.3	Recall	65
3.8.4	Confusion Matrix	66
3.9	Post- Processing	66
3.9.1	Thresholding	67
3.9.2	Smoothing	67
3.9.3	Counting	68
3.10	Summary	69

CHAPTER 4: Model Implementation and Training	70
4.1 Overview	70
4.2 Experimental Design	70
4.3 Dataset Selection	71
4.3.1 Synthetic Dataset	71
4.3.2 Hybrid Dataset	72
4.4 Data Preprocessing and Preparation	73
4.4.1 Data Cleaning	73
4.4.2 Encoding Categorical Data	74
4.4.3 Normalization of Features	75
4.4.4 Data Segmentation	75
4.4.5 Splitting the Dataset	75
4.5 Model Training and Hyperparameter Tuning	76
4.5.1 Model Architecture	77
4.5.2 Training Configuration	77
4.5.3 Early Stopping Mechanism	78
4.6 Performance Evaluation	79
4.6.1 Accuracy Analysis	80
4.6.2 Precision and Recall Analysis	80
4.6.3 Vehicle Counting Mechanism	81

4.7	Results	82
4.7.1	1D CNN Model Performance on Synthetic Dataset	83
4.7.2	1D CNN Model Performance on Hybrid Dataset	84
4.7.3	Performance of other Four Deep Learning Models on Synthetic Dataset	85
4.7.4	Performance of other Four Deep Learning Models on Hybrid Dataset	87
4.8	Comparative Analysis with Other Deep Learning Models	89
4.9	Results Comparison	90
4.10	Discussion	95
4.11	Summary	96
	CHAPTER 5: Experimental Results and Performance Analysis	98
5.1	Overview	98
5.2	Framework Overview	99
5.3	Data Acquisition	101
5.3.1	Data Preprocessing and Preparation	102
5.4	Model Training	102
5.5	Scenario 1: Vehicle Classification	105
5.6	Scenario 2: Vehicle Counting	108
5.7	Performance Evaluation	111
5.7.1	Classification Metrics: Accuracy, Precision, Recall, F1-score	111

5.7.2	Detection and Counting Metrics: Mean Average Precision (mAP), Confusion Matrix, and Counting Accuracy	112
5.7.3	Computational Efficiency: Inference Speed and Resource Utilization	112
5.8	Implementation Details	113
5.8.1	Framework Development Tools	113
5.8.2	TensorFlow and PyTorch for Deep Learning	113
5.8.3	Scikit-learn for Evaluation Metrics	114
5.8.4	Hardware: GPU-Accelerated Training for Efficiency	114
5.9	Results	115
5.9.1	Framework Performance Overview	115
5.9.2	Classification Metrics Evaluation	118
5.9.3	Detection and Counting Metrics Evaluation	119
5.9.4	Performance Analysis on Unseen Data	121
5.9.5	Computational Efficiency and Real-Time Feasibility	123
5.10	Summary of the Results	123
	CHAPTER 6: CONCLUSION	125
6.1	Summary of Findings	125
6.1.1	Data Collection and Preprocessing	127
6.1.2	Model Development and Training	127
6.1.3	Performance Evaluation	128

6.1.4	Comparative Analysis	128
6.2	Contributions of Research	129
6.2.1	Real-Time Feasibility and Deployment Considerations	129
6.3	Limitations of the Study	129
6.3.1	Dataset Constraints	130
6.3.2	Sensor Noise and Environmental Factors	130
6.3.3	Scalability and Computational Costs	130
6.4	Future Research Directions	133
6.4.1	Expansion of Dataset and Data Augmentation	133
6.4.2	Hybrid Deep Learning Models	133
6.4.3	Real-Time Edge Computing Implementation	134
6.4.4	Integration with Intelligent Transportation Systems (ITS)	134
6.4.5	Adaptive Learning and Model Retraining	134
6.5	Conclusion	134
	REFERENCES	136

Page

LIST OF TABLES

		Page
Table 1.1	Total Industry volume 2024 versus 2023	2
Table 1.2	Research Questions, Research Objectives and Research hypothesis	8
Table 2.1	Comparison of the Detection Methods	26
Table 2.2	Existing tools for vehicles classification and counting	26
Table 2.3	Comparison Related literature	35
Table 3.1	Comparison of Four Models with 1D-CNN	55
Table 3.2	Code for layers of 1D-CNN	60
Table 4.1	Dataset Structure	69
Table 4.2	Model Architecture Configuration	77
Table 4.3	Training Configuration	78
Table 4.4	Precision, Recall and F1-Score	81
Table 4.5	Models Performance for Synthetic Dataset (10k and 100k Data points)	90
Table 4.6	Models Performance for Hybrid Dataset (10k and 100k Data points)	93
Table 5.1	Architecture of 1D CNN Model presented in a Structured Table	103
Table 5.2	Hyperparameters for the Model	104
Table 5.3	Classification of Vehicles Parameters	118
Table 5.4	Detection and Counting Metrics Performance	119
Table 5.5	Classification performance on Unseen Data	121
Table 5.6	Confusion Matrix for Unseen Data	122
Table 6.1	Summary of Research Work	126
Table 6.2	Comparison of Models	128

LIST OF FIGURES

	Page
Figure 1.1	Roadside vehicle census using human personnel. 6
Figure 2.1	Vehicle Classification Taxonomy 13
Figure 2.2	Conventional Methods Process Flow Diagram 14
Figure 2.3	Mechanical Methods Process Flow Diagram 15
Figure 2.4	Schematic showing the process flow of electric and acoustic impedance matching involved in the design of piezoelectric receiver 16
Figure 2.5	For a pulse–echo mode where the same piezoelectric element is used both as actuator and sensor. 17
Figure 2.6	The layout of traffic congestion prediction system. 20
Figure 2.7	Data flow of AI-based Vehicle Detection system 21
Figure 2.8	Taxonomy of Vehicle Classification Methods Information 22
Figure 2.9	Digital Methods Process Flow Diagram 23
Figure 3.1	Research Design 50
Figure 3.2	Schematic Design of Smart Road Energy Harvesting System 51
Figure 3.3	Architecture of 1D CNN Model 57
Figure 4.1	Accuracy and Trend loss of 1D CNN Model Performance on synthetic Dataset (10k Data Points) 83
Figure 4.2	Accuracy and Trend loss of 1D CNN Model Performance on synthetic Dataset (100k Data Points) 84
Figure 4.3	Accuracy and Trend loss of 1D CNN Model Performance on Hybrid Dataset (10k Data Points) 85
Figure 4.4	Accuracy and Trend loss of 1D CNN Model Performance on Hybrid Dataset (100k Data Points) 85

Figure 4.5	Four model accuracy outputs (Synthetic Dataset 10k and 100k)	87
Figure 4.6	Four model accuracy outputs (Hybrid Dataset 10k and 100k)	89
Figure 5.1	Framework Overview	99
Figure 5.2	Workflow of Vehicles Classification	106
Figure 5.3	Workflow of Vehicles Counting	109
Figure 5.4	Framework's Model Accuracy and loss	116

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
1D-CNN	One-Dimensional Convolutional Neural Network
DL	Deep Learning
FBG	Fiber Bragg Grating
ITS	Intelligent Transportation System
IoU	Intersection over Union
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
ML	Machine Learning
RNN	Recurrent Neural Network
YOLO	You Only Look Once (Object Detection Algorithm)
CGS	Centre for Graduate Studies
UNIMAS	Universiti Malaysia Sarawak
SNR	Signal-to-Noise Ratio
FPS	Frames Per Second (implied in model speed evaluation)
CRFV	Close-Range Following Vehicle
MAA	Malaysia Automotive Association
CGS	Centre for Graduate Studies
UNIMAS	Universiti Malaysia Sarawak

CHAPTER 1

INTRODUCTION

1.1 Study Background

Road transportation is considered one of the key drivers in economic and social development in any city. It plays a vital role in world commerce where companies rely on road transport to deliver goods and individuals are enabled to trade online orders. Malaysia as a developing country has considered that the road transportation network is significant to the economic generation. Particularly in urban areas the increasing trend of vehicle usage in Malaysia highlights the urgent need for intelligent traffic monitoring systems. The rapid population, rising economic growth and increasing employment are factors that affect the growth of the number of vehicles. With greater comfort, convenience, flexibility, and freedom to own private vehicle to reach any destination at any time, there results in a boost to the demand for vehicles. According to the Malaysia Automotive Association (MAA), the total number of vehicles registrations in Malaysia hit 36.3 million units (Daim, 2023). The growth rate of vehicles indicates that there is the need to properly monitor the flow of the traffic to enable proper planning and management of the road in terms of expansion, rerouting, installation of traffic lights. This research aims to explore the traffic monitoring systems. The total industry volume 2024 forecast is as follows shows in the Table 1.1.

Table 0.1: Total Industry volume 2024 versus 2023
(Malaysian Automotive Association (MAA), 2024)

MARKETSEGMENT	2024 (FORECAST)	2023 (ACTUAL)	VARIANCE	
			UNITS	%
Passenger vehicles	666,000	719,160	(53,160)	(7.4)
Commercial vehicles	74,000	80,571	(6,571)	(8.2)
Total vehicles	740,000	799,731	(59,731)	(7.5)

According to Jabatan Kerja Raya (JKR) Malaysia (2016) discussed the critical issue of traffic blocking, which refers to the temporary obstruction of vehicle movement due to road works, accidents, or congestion, poses significant challenges to traffic flow efficiency in urban areas of Malaysia, particularly in cities like Kuala Lumpur, Penang, and Johor Bharu. It highlights the necessity of traffic data collection for road development and management, and outlines the challenges faced due to rapid urbanization, poorly planned road networks, increasing vehicle numbers, and lack of efficient public transportation. The report details the methodology of conducting traffic surveys at intersections and screen lines, emphasizing manual counting during peak hours, and categorizing vehicles into five classes. It also addresses the difficulties in gathering accurate traffic data, such as the need for trained staff, proper equipment, and safety considerations. The outcome of these surveys is crucial for understanding current traffic volumes and developing mitigation measures to improve the Level of Service (LOS) on roads, aiming to improve the negative impacts of traffic jams (Jalan, 2016).

A traffic census directs the systematic collection of data related to traffic flows, patterns, and characteristics at specific locations or across a network over a defined period. This data typically includes information such as vehicle counts, types of vehicles, speeds,

and sometimes additional details like vehicle classifications (e.g., cars, trucks, motorcycles). The primary objective of a traffic census is to obtain accurate understandings into the use of transportation infrastructure, assisting in the optimization of traffic flow, infrastructure planning, and the development of balanced transportation policies.

Artificial intelligence (AI) is like teaching computers to think and learn in ways like humans. Imagine you have a smart robot friend. You can teach it how to do things by giving it examples and rules to follow. Over time, it can use what it is learned to solve problems or make decisions on its own. AI can help with all sorts of tasks, like recognizing faces in photos, answering questions, driving cars, or even playing games. According to Mhlanga AI is described as the science and engineering behind the creation of intelligent devices, especially intelligent computer programs (Mhlanga, 2023) .

Deep learning (DL) is a subset of ML that aims on the development of algorithms and models inspired by the structure and function of the human brain. DL algorithms work by learning from data, rather than relying on explicit instructions or pre-defined rules. These algorithms use multiple layers of interconnected artificial neurons (also known as nodes). These nodes are stacked next to each other in three categories such as the input layer, the hidden layers (s) and the output layer. According to Jeremy, this computational technique plays a fundamental role in extracting features from data and learning from data. These models can be applied in many fields such speech recognition and image classification (Jeremy Howard & Sylvain Gugger, 2020).

Similarly, in deep learning matrices are initial elements essential to neural network operations. They serve various roles, including representing input data, storing weights and biases that define network parameters, describing activation outputs after applying nonlinear functions, and measuring loss and gradients during training. Matrices are essential in

specialized architectures like convolution neural networks (CNNs) and recurrent neural networks (RNNs) help in drawing off characteristics and in developing temporal models, respectively. Through matrix operations like multiplication and addition, data is processed by DL models, getting predictable results from it, and improving performance across several tasks.

There are several types of classifications such as binary, multi class, multi class label, hierarchical and temporal. Binary classification model predicts one of the two classes for each input instance. This classification is mostly used in image recognition (identifying objects in images). According to Yari using the binary classification, the aim is to make an algorithm that could predict one of the two classes (Yari et al., 2020).

Unlike binary classification, multi-class classification involves predicting one class out of multiple classes for each input instance. Similarly, multi-label classification manages assigning multiple labels to a given instance (Mohammed et al., 2020). This type of classification is widespread in tasks such as document categorization (assigning multiple topics to a document) and scene understanding (labelling images with multiple objects). On the other side, hierarchical classification organizes classes into a hierarchy, where classes at higher levels are broader categories, and classes at lower levels are more specific. This structure helps in handling large and complex classification tasks by breaking them down into smaller, more controllable subtasks. Temporal classification, also known as sequence labelling or sequence classification, involves predicting labels for sequences of data over time. These classifications models can be applied in various fields.

In the field of diagnosis, medical image classification plays a key role in prediction and early diagnosis of critical diseases. Some researchers have introduced Optimal DL, a classifier for lung cancer, brain images, and Alzheimer's disease (Raj et al., 2020). It

combines pre-processing, feature selection, and classification to enhance medical image analysis. Using the Opposition-based Crow Search algorithm, it selects optimal features like multi-texture and grey-level attributes from pre-processed images. This approach significantly improves classification accuracy, specificity, and sensitivity, promising better diagnostic outcomes for patients.

Recently, Text classification is the highly fundamental and essential task in natural language processing. The last decade has seen a flow of research in this area due to the record success of deep learning. Numerous methods, datasets, and evaluation metrics have been proposed in this area. Some researchers applied their innovative techniques to create taxonomy for text classification and feature extraction (Q. Li et al., 2022).

Image Recognition is another significant field of computer vision and pattern recognition. According to Zhu deep learning is a process of simulating the human visual system and the cognitive link of the human brain, extracting data features in a structured manner, and finally obtaining deep information of the image. Their model achieved a good image recognition result (Zhu,2022). Due to their ability to recognize objects in images, these classifiers are useful in a variety of applications, including security (identifying intruders in surveillance footage), medical imaging (detecting diseases from scans), and autonomous cars controlling (identifying pedestrians, traffic signs, etc.).

1.2 Problem Statement

There are many problems with regards to road transportation within a city area, the city management body such as city council and related agencies such as Road Transport Authority, City Planning Unit faces multitude of challenges to manage road system, monitor traffic situation, and prepare future expansion of road and relevant infrastructures. For

example, expansion of existing roads or the construction of new road requires data on traffic volume to ascertain the technical specification of the expansion and the cost involved.

One of the most challenging issues within road transportation domain is to monitor and collect information on volume of vehicles passing through roads and specific location within the road system. One of the common approaches to count vehicle traffic volume is to conduct roadside vehicle census whereby human personnel are employed to count the number of vehicles passing at specific points or junctions as shown in Figure 1.1. This approach is time consuming, involves human labour, considerable risk at roadside, unfavourable condition (heat and rain), can only be scheduled if required due to cost involvement.



Figure 0.1: Roadside vehicle census using human personnel.

Although there are several methods of vehicle detection and counting, the current methods have constraints with regards to accuracy, scalability and suitability to a complex

traffic environment. Mechanical approaches have expensive infrastructure and traditional image processing techniques are not effective when dealing with diverse conditions of light and weather. Despite improvements by machine learning techniques, they usually rely on manually developed features and the generalization is therefore constrained. Deep learning, especially the convolutional neural networks (CNNs) have excelled more at image recognition tasks as they automatically learn hierarchical features out of raw data. Conversely, limited studies have applied CNNs specifically to vehicle classification and counting in Malaysian traffic conditions. Thus, this study fills the gap by creating a CNN-based vehicle classification and counting framework, to improve its accuracy and efficiency in the real-world traffic management.

1.3 Research Objectives

The main objective of this project is to formulate a framework for automated detection, classification, and counting of vehicles utilizing a deep learning approach using dataset constructed from road energy harvesting system.

Other objectives include:

- i. To construct training dataset from the raw data collected and processed from road energy harvesting system.
- ii. To design and develop a framework for detection, classification and counting of vehicles utilizing a deep learning approach.
- iii. To evaluate the model for performance and accuracy with baseline models.

1.4 Research Questions

RQ1: How to process/prepare raw data from energy harvesting system to be used in training model for detection of vehicle?

RQ2: How can we design and develop a deep learning model for the detection, classification, and counting of vehicles that outperform traditional methods and existing automated systems in terms of accuracy, efficiency, and strength?

RQ3: How does assessing the model contribute to understanding their performance and accuracy, and what enhancements can be made based on these insights?

1.5 Research Hypothesis

RH1: Processing raw road energy harvesting data, including voltage and frequency signals, can provide features like signal magnitude and duration, which are effective for training a deep learning model to accurately count and classify vehicles.

RH2: The proposed CNN framework improves vehicle counting performance under varying traffic conditions.

RH3: CNN-based models achieve significantly higher accuracy in vehicle classification compared to traditional machine learning approaches.

Table 0.2: Research Questions, Research Objectives and Research hypothesis

Problem Statement (PS)	Research Question (RQ)	Research hypothesis (RH)	Research Objectives (RO)
------------------------	------------------------	--------------------------	--------------------------

Table 1.2 continued

The challenge is to determine if the data collected from road energy harvesting systems can be effectively utilized to train a deep learning architecture for the purpose of vehicle census.	Does the data collected from energy harvesting system can be used to train a deep learning architecture to identify a vehicle?	Processing raw road energy harvesting data, including voltage and frequency signals, can provide features like signal magnitude and duration, which are effective for training a deep learning model to accurately count and classify vehicles.	To construct training dataset from raw data collected and processed from road energy harvesting system
Existing methods for tracking and categorizing vehicles in traffic have limitations, causing inaccuracies and inefficiencies in monitoring systems. There is a need for a better way to count and classify vehicles accurately and quickly.	How can we design and develop a deep learning model for the detection, classification, and counting of vehicles that outperform traditional methods and existing automated systems in terms of accuracy, efficiency, and strength?	The proposed CNN framework improves vehicle counting performance under varying traffic conditions.	To design and develop a framework for detection, classification and counting of vehicles utilizing deep learning approach.
The efficiency and accuracy of a model needs measurement to ensure optimal performance.	How does assessing the model contribute to understanding their performance and accuracy, and what enhancements can be made based on these insights?	CNN-based models achieve significantly higher accuracy in vehicle classification compared to traditional machine learning approaches.	To evaluate the model for performance and accuracy.

The above Table 1.2 shows the summary of Problem statement, research question, research hypothesis and research objectives.

1.6 Idea and Motivation

This research is motivated by the growing challenges of traffic congestion in Malaysia and globally. With rapid urbanization, there is a strong need for intelligent traffic monitoring systems that can automatically classify and count vehicles to support traffic management authorities. Deep learning provides a promising direction due to its proven success in visual recognition tasks, which motivates this study to apply CNNs in the domain of intelligent transportation systems.

1.7 Scope of Study

The scope of this research is limited to vehicle classification and counting using road traffic video data. The study focuses on commonly observed vehicle categories such as cars, motorcycles, buses, and trucks. The dataset is limited to publicly available traffic footage and selected case studies in Malaysia. The research is restricted to CNN-based models and does not cover other deep learning architectures such as RNNs or transformers. The study addresses vehicles classification and counting.

1.8 Significance of Work

This research contributes rationally by progressing the application of CNNs in automated vehicle detection, classification, and counting. Practically, the proposed framework can assist traffic authorities by providing accurate, real-time traffic data for better decision-making. The consequences of this study can support smart city initiatives and intelligent transportation systems by present a cost-effective, scalable, and automated solution for traffic monitoring.

1.9 Chapter Summary

This thesis is structured into six chapters. Chapter 1 provides an overview of the research, covering the background, problem statement, objectives, hypothesis, scope, significance, and overall organization. Chapter 2 then goes into a detailed literature review of similar work, dealing with deep learning approaches to vehicle detection, classification, and counting. Chapter 3 provides details of the proposed approach, such as the preparation of the dataset and the construction of the 1D-CNN architecture and experiment. Chapter 4 then gives the details of how the model was trained and implemented. Chapter 5 analyses the performance of the framework by providing the results of the experiment. Lastly, in Chapter 6 the research findings are summarized and recommendations are made on future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews existing literature related to vehicle detection, classification, and counting. It begins with an overview of conventional methods such as manual counting and sensor-based techniques, followed by image processing and traditional machine learning approaches. The chapter subsequently examines the advance developments in the field of deep learning, and specifically the convolutional neural networks (CNNs), and how they have found application in the intelligent transportation systems. Finally, the chapter highlights research gaps that justify the proposed study.

2.2 Overview of Road Transportation Management

Road transportation forms an integral part of urban infrastructure, which serves as the primary backbone of economic and social development. The efficiency of road networks is dependent upon the smooth flow of vehicular traffic, with effective urban planning capable of attaining sustainable development. This section gives a background of the problems in managing road transportation, emphasizing the need for advanced technologies, such as Conventional methods, mechanical methods and digital/AI methods for vehicle classification and counting shown in the Fig 2.1 below.

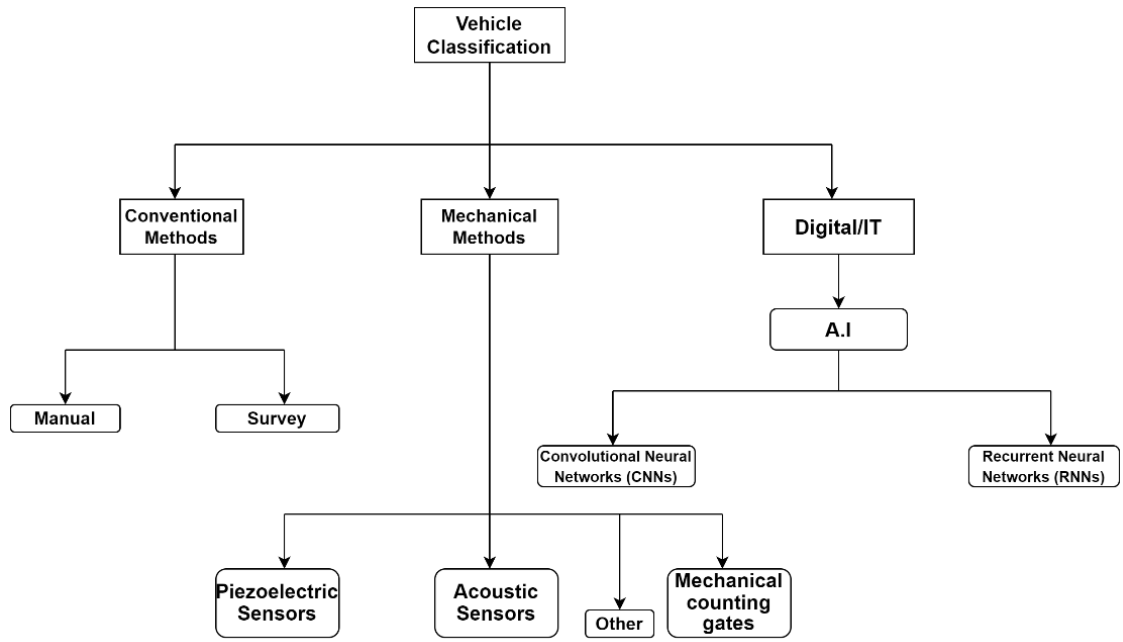


Figure 2.1: Vehicle Classification Taxonomy

2.3 Conventional Methods for Vehicle Classification and Counting

Vehicle classification and counting have traditionally been done through manual and survey methods. These include labour-intensive collection of data. In manual method involve human interference to identify and extract vehicles from images or video frames. Operators manually draw bounding boxes around vehicles or annotate vehicle class. Similarly, surveys collect data on traffic flow, vehicle types and counts. This can be conducted using roadside manual observations. According to Chen researchers analyse survey data to understand traffic patterns, vehicle distribution, and congestion and provide valuable insights but may not capture real-time dynamics. The conventional methods are labour-intensive, time-consuming, prone to errors and not suitable for real-time applications (C.-Y. Chen et al., 2014). This section describes the limitations of manual methods and

proves the need for an automated system to address the challenges of manual methods. The Fig 2.2 illustrates the process flow of conventional methods.

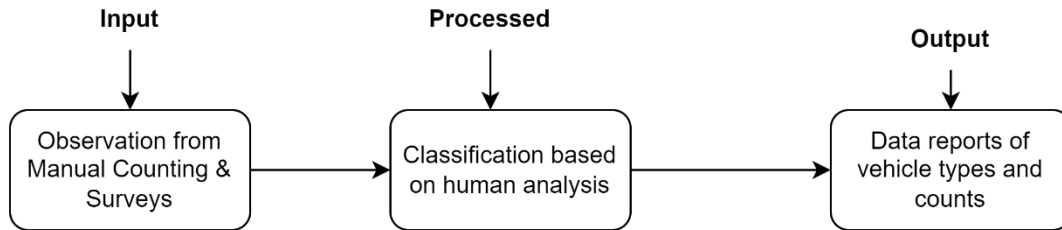


Figure 2.2: Conventional Methods Process Flow Diagram

2.4 Mechanical Methods for Vehicle Classification and Counting

Mechanical methods for vehicle classification and counting have redefined traffic monitoring and management systems. These methods usually come with the embedded sensor in the road surfaces or mounted on the roadside infrastructure. These systems use technologies such as piezoelectric sensors, inductive loops, or pneumatic tubes to detect the features of moving vehicles. Accordingly, they can conduct precise vehicle counting and classification based on vehicle size and metal composition.

These mechanical methods over conventional counting techniques are increased accuracy, reliability, and efficiency. These systems eliminate the human dimension in vehicle detection and the associated classification to reduce human intervention, and the potential for errors in the manual counting methods. They offer real-time data on traffic flow patterns, which allows transportation authorities to make data-driven decisions on traffic management strategies. The adoption of mechanical methods for vehicle classification and counting has led to significant improvements in traffic-monitoring capabilities through more effective and data-driven approaches to urban mobility management and the development of

transportation infrastructure. While deep learning approaches offer advanced capabilities for automated vehicle classification and counting, these methods have usually been employed for these purposes. This section covers an overview of the mechanical methods used in vehicle classification and counting. The Fig 2.3 illustrates the process flow of mechanical methods.

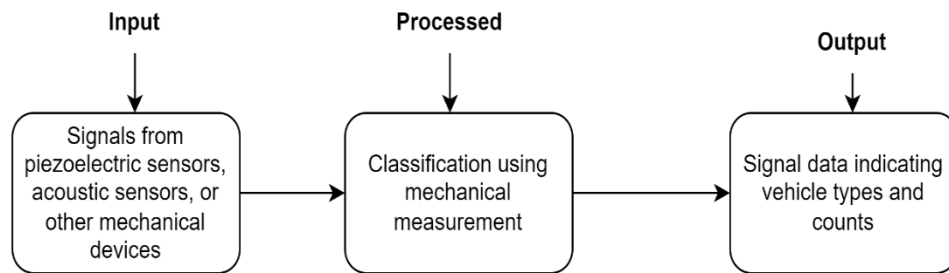


Figure 2.3: Mechanical Methods Process Flow Diagram

2.4.1 Piezoelectric Sensors

Piezoelectric sensors convert the mechanical stress or pressure applied to the sensor into electrical signals due to the piezoelectric effect (Rathod, 2020). They can be embedded directly in road surfaces or installed beneath roadway layers to detect vehicle movements and classify vehicles according to the pressure applied by their tires. Piezoelectric sensors can detect and count vehicles with a high degree of accuracy and are often incorporated into intelligent transportation systems for real-time monitoring of traffic. These sensors represent a significant asset in the field of traffic engineering, providing invaluable and reliable data for the implementation of systems and applications in the monitoring and management of traffic flow. The Fig 2.4 illustrates the process flow of electric and acoustic impedance matching involved in the design of piezoelectric receiver.

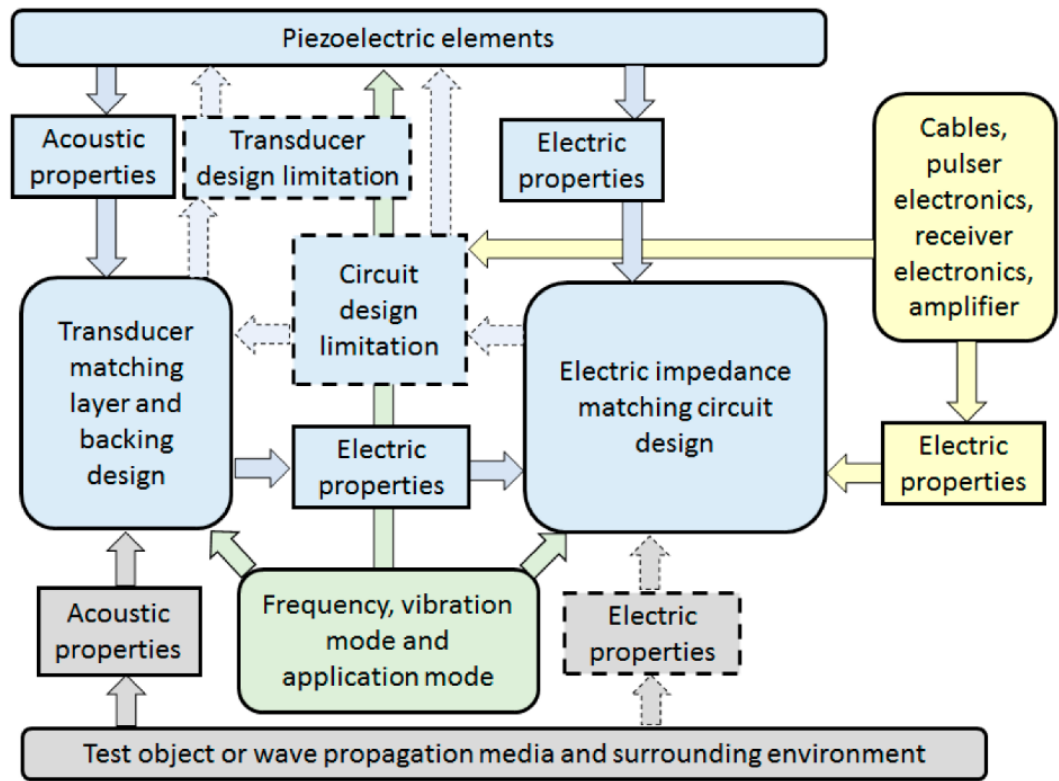


Figure 2.4: Schematic showing the process flow of electric and acoustic impedance matching involved in the design of piezoelectric receiver.

(Rathod, 2020)

According to shokravai 2020 these sensors are sensitive to temperature and surface conditions due to voltage variation. Any changes in temperature and surface characteristics really do affect the electrical properties of the sensor, and thus accurate calibration and monitoring of the environment are needed to have appropriate and reliable measurements (Shokravi et al., 2020).

2.4.2 Acoustic Sensors

Acoustic sensors use sound waves to detect and classify vehicles using their acoustic signatures. The acoustic sensors can identify vehicles by analysing the unique sound patterns produced by their engines, tires, and other components. Acoustic sensors are helpful in situations where the visual detection might be poor, such as in tunnels or under conditions

of low light. They provide a non-optical, non-electromagnetic method of classifying and counting vehicles, accordingly, complementing optical and electromagnetic methods.

Acoustic sensors are devices that detect sound waves or vibrations in various environments and convert them into electrical signals for analysis. And these are highlighted for their low cost, simplicity, and non-intrusive nature. The Figure 2.5 illustrates the process for a pulse–echo mode where the same piezoelectric element is used both as actuator and sensor. Although acoustic sensors were reviewed in the literature as a related technology, they were not incorporated into the methodology of this study.

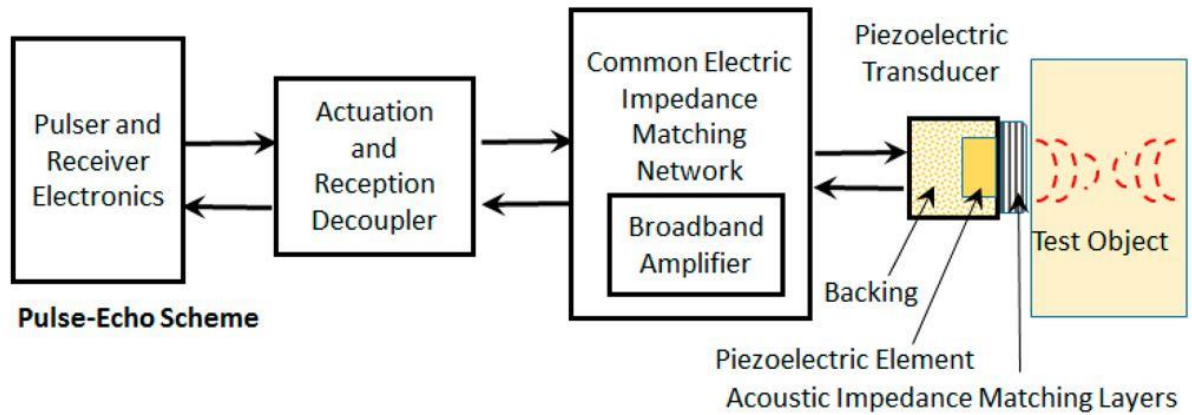


Figure 2.5: For a pulse–echo mode where the same piezoelectric element is used both as actuator and sensor.

(Rathod, 2020)

2.4.3 Mechanical counting gates

Mechanical counting gates refer to physical barriers or structures laid across roadways purposely for counting and classifying vehicles. They are equipped with sensors or mechanisms that register passing vehicles. When the vehicle passes on the gate, sensors contained within the construction identify them and thus lead to the activation of the counting mechanism, which then correctly counts the number of vehicles passing. According to

Ohidujjaman this method helps in reducing traffic congestion and managing vehicle parking efficiently by providing accurate vehicle counts to the traffic management authorities (Ohidujjaman et al., 2021).

The mechanical counting gates are dependable, accurate, and automated. The primary features of the systems described are that they could be particularly useful and could be designed in such a way that they can classify vehicles depending on their size, their weight, or their axle. The installation and maintenance requirements of counting gates, the sensitivity to environmental disturbances, limitations in space, and limited classification capabilities for some vehicle classes.

2.4.4 Integration with Deep Learning Approaches

While Mechanical methods provide accurate data about the detection and counting of vehicles, they may not be able to classify the vehicles with fine stability. Recently, much research interest has been shown in merging mechanical sensors with deep learning approaches to make the system much more accurate and efficient for vehicle classification. According to one of the researchers the development of hybrid systems that can provide insights into the dynamics of traffic and reduce the limitations of individual approaches by combining the strengths of both mechanical and deep learning methods (Neupane et al., 2022).

Vehicle classification and counting depend on mechanical methods, which use piezoelectric sensors, acoustic sensors, and mechanical counting gates. Such methods provide real-time data, which is accurate when used in traffic management applications and can be regarded as an addition to deep learning techniques. The use of deep learning approaches includes the training of artificial neural networks to automatically learn and

recognize patterns in data, such as vehicle types or counts, from a large dataset. The integration with these mechanical methods improves the accuracy and efficiency of the deep learning methods in the classification and counting of vehicles.

2.5 Application for Artificial Intelligence in Transportation Management

Artificial Intelligence, from its subfield Machine Learning, has emerged as a powerful set of technologies in the transformation of a variety of sectors, including transportation management. This part outlines the applications of AI (Artificial Intelligence) and ML (Machine Learning) techniques in traffic analysis, vehicle detection, and classification. Especially, it considers the role that deep learning techniques are now performing in changing the automatic systems for vehicle detection, classification, and counting.

2.5.1 Traffic analysis

Accurate traffic analysis is vital for the management of transportation because it provides key insights into traffic patterns, congestion levels, and the dynamics of the overall flow. In the process, AI, and ML play important roles in processing the data and obtaining meaning from vast amounts of traffic-related data collected from a variety of sources which include traffic cameras, road sensors, GPS devices, and mobile applications (Akhtar & Moridpour, 2021). The use of AI means data analysis methods is used by transportation authorities to derive useful inferences for making informed decisions in relation to traffic management strategies, planning infrastructure, and allocating resources. Similarly, by using AI and ML algorithms for the analysis of traffic, one can find recurring patterns, anticipate traffic congestion, and optimize traffic flow to ensure efficient and reliable transportation systems. The following Figure 2.6 illustrates layout of traffic congestion prediction system.

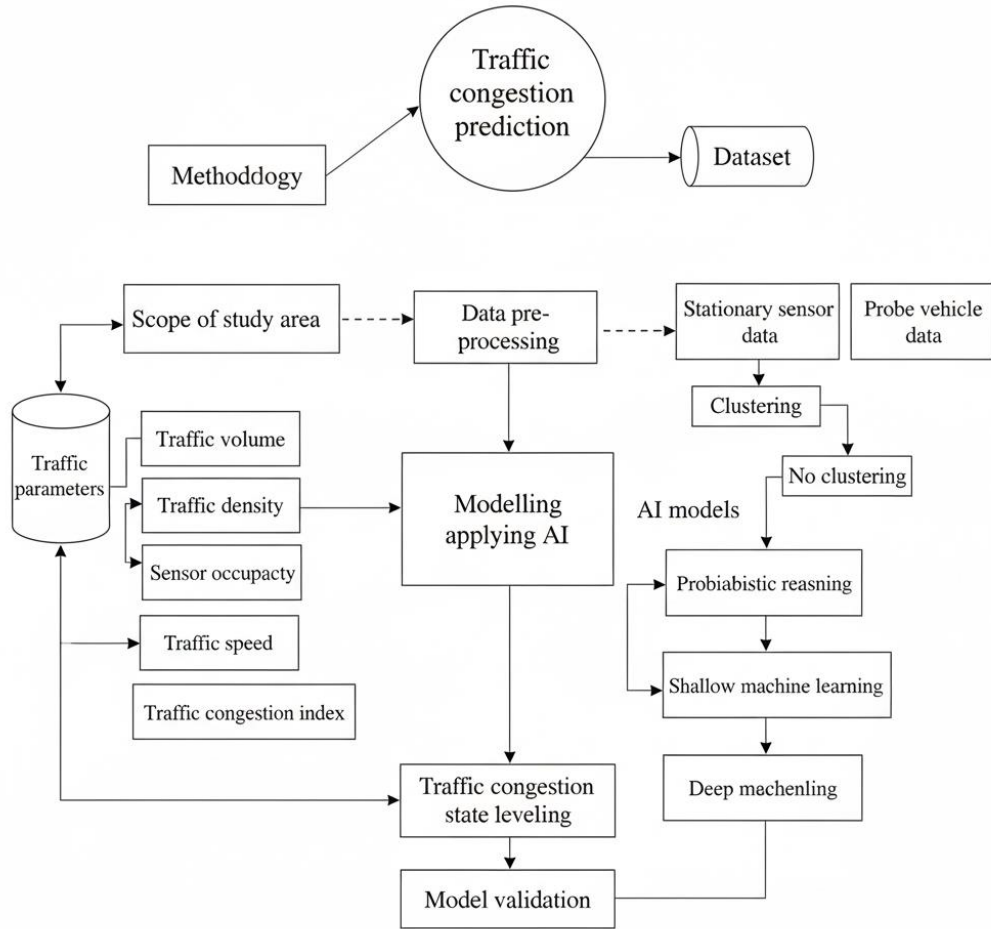


Figure 2.6: The layout of traffic congestion prediction system.
(Akhtar & Moridpour, 2021)

2.5.2 Vehicle Detection

Another key component of intelligent transportation management is vehicle detection. Which is necessary to accomplish activities like observing traffic, managing congestion, or detecting incidents (Tak et al., 2021) . AI technologies, especially computer vision algorithms, play a crucial role in the detection and tracking of moving vehicles in real time, as seen through video feeds originating from traffic cameras. These algorithms are implemented to use advanced image processing techniques for detecting vehicles based on

their visual attributes shape, colour, and motion patterns. Object detection models using ML techniques, such as Convolution Neural Networks, achieve this task of recognizing vehicles from images or video frames with high precision. Additionally, this implementation captures and detects vehicles in images or video frames using advanced image processing techniques. Such techniques, when combined with data from various sensors such as cameras, LiDAR, radar, and ultrasonic sensors, enable transportation authorities to provide higher reliability and effectiveness of vehicle detection systems even under challenging conditions, including harsh weather or reduced visibility. Fig 2.7 below illustrates Data flow of AI-based Vehicle Detection system.

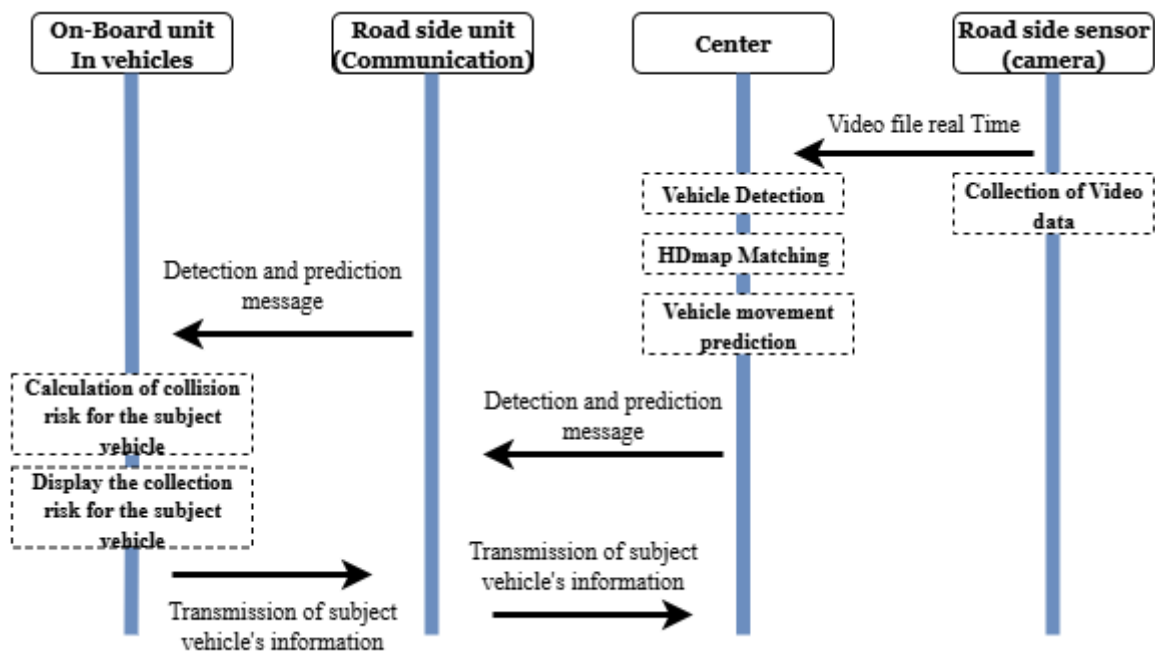


Figure 2.7: Data flow of AI-based Vehicle Detection system
(Tak et al., 2021)

2.5.3 Vehicle Classification

Vehicle classification extends the key to understanding the composition of traffic, analyzing travel patterns, and optimizing traffic management strategies. Particularly, AI

technology through deep learning algorithms has demonstrated remarkable capabilities in automatically categorizing vehicles into different classes based on their characteristics such as size, shape, and purpose (Gholamhosseinian & Seitz, 2021). The algorithms learn discriminative features for classification from large datasets of annotated images of vehicles. By accurately classifying vehicles in real time while they pass through monitored sections, transportation managers will have insightful knowledge of the composition of traffic, the type of vehicles, and their usage trends. This information is critical to optimize traffic flow, safety, and the overall efficiency of a transportation system. AI technologies in vehicle classification help transportation authority take data-driven decisions, implement targeted interventions, and enhance the performance of transportation systems. The following Fig 2.8 illustrates Taxonomy of Vehicle Classification Methods Information.

Deployment	Category	Method	Global Position	Accel.	Dir.	Speed	Axle Config	Type/Model	Weight	Count	Auto
Non-intrusive	Vision-based	Video Images	•	•	•	•	•	•	•	•	•
Non-intrusive	Sound-based	Acoustic	•	•	•	•	•	•	•	•	•
Non-intrusive	Sound-based	Ultrasonic	•	•	•	•	•	•	•	•	•
Non-intrusive	Remote Sensing	Infrared	•	•	•	•	•	•	•	•	•
Non-intrusive	Remote Sensing	Laser Scanner	•	•	•	•	•	•	•	•	•
Non-intrusive	Remote Sensing	LiDAR	•	•	•	•	•	•	•	•	•
Non-intrusive	Remote Sensing	Radar	•	•	•	•	•	•	•	•	•
Non-intrusive	Remote Sensing	RF Transceivers	•	•	•	•	•	•	•	•	•
Non-intrusive	Remote Sensing	Wi-Fi/LTE	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Inductive Loops	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Magnetic Sensors	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Fiber Optic	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Piezoelectric	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Pneumatic	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Strain Gauge	•	•	•	•	•	•	•	•	•
Intrusive	Contact	Seismic & Vibration	•	•	•	•	•	•	•	•	•
Off-road	Aerial	UAVs	•	•	•	•	•	•	•	•	•
Off-road	Aerial	Satellite	•	•	•	•	•	•	•	•	•
Off-road	GPS-based	In-vehicle GPS Device	•	•	•	•	•	•	•	•	•
Off-road	GPS-based	Mobile Apparatus	•	•	•	•	•	•	•	•	•
Hybrid	Multi-Method	WIM	•	•	•	•	•	•	•	•	•
Hybrid	Multi-Method	WSN	•	•	•	•	•	•	•	•	•
Hybrid	Multi-Method	VANETs	•	•	•	•	•	•	•	•	•
Manual	Manual	Manual Observation	•	•	•	•	•	•	•	•	•

Figure 2.8: Taxonomy of Vehicle Classification Methods Information
(Gholamhosseinian,2021)

2.6 Deep Learning Techniques for Vehicle Detection and Classification

In recent years, deep learning methods have empowered the field of vehicle recognition: the ability to detect and classify vehicles from still images and video streams using complex neural network architectures. According to Jagannathan Deep Learning Techniques for Vehicle Detection and Classification are sensitive to advanced neural network architectures that automatically identify and classify vehicles in images and videos (Jagannathan et al., 2021). Similarly, these methods have improved the effectiveness and accuracy of the vehicle recognition system, enabling diverse applications in traffic monitoring, video surveillance, and autonomous driving. In this section, we discuss the most popular deep learning approaches for vehicle detection and classification, among which are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), describing their architectures, training procedures, and application scenarios. The below Fig 2.9 illustrates Digital Methods Process Flow.

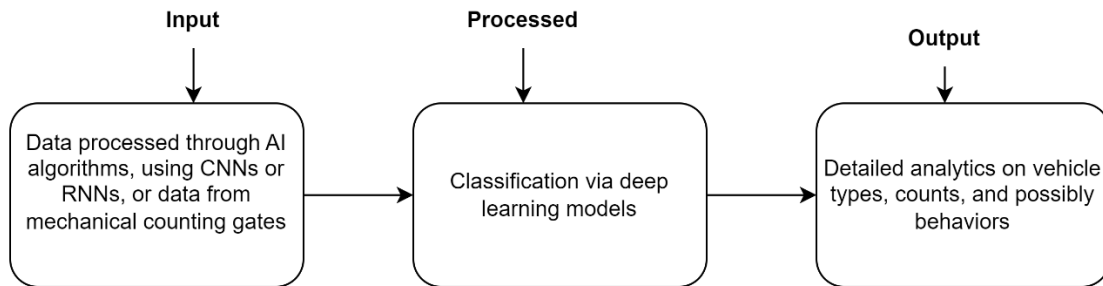


Figure 2.9: Digital Methods Process Flow Diagram

2.6.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks have become the core for deep learning-based computer vision undertakings because they efficiently capture the spatial hierarchies in visual information. CNNs are built on many different layers, the most important being

convolutional layers, pooling layers, and fully connected layers. According to dabiri the CNN (Convolutional Neural Networks) Vehicle Classification is a CNN-layer stack to extract abstract features from the data, a pooling operation for capturing the most valuable information is done, followed by a SoftMax layer to perform the classification task (Dabiri et al., 2020).

This is performed by feeding annotated vehicle images to a CNN and adjusting the network's parameters (weights and biases) to minimize some specified loss function. Typically used loss functions include categorical cross-entropy for classification tasks. Training is normally done using gradient-based optimization algorithms such as Stochastic Gradient Descent (SGD) or Adam, which update the network's parameters to minimize the loss function in an iterative approach.

The relevance of CNNs to this study in particular is due to the capacity to derive hierarchical features of one-dimensional, time-series data. The piezoelectric sensor raw input signals in this work are handled as a sequential stream of data which CNNs can process. These signals are auto-learned by the network and include the identification of the important patterns which are unique to every vehicle type like voltage variations and signal duration. Contrary to classic approaches that rely on a set of rules, CNNs learn by themselves to distinguish between types of vehicles, making them capable of strong classification regardless of the traffic conditions. We explored CNNs in this work with one of the optimization of their parameters (e.g., kernel size, the number of convolutional layers, learning rate, and activation functions) to achieve the maximum classification accuracy, which proves the direct relevance of CNNs to the task of traffic monitoring by sensor data.

2.6.2 Applications of Convolutional Neural Networks (CNNs)

The integration of various technologies in transportation systems has significantly enhanced road safety and traffic management. According to Kothai his proposed Traffic Monitoring System, situated within the framework of Vehicular Adhoc Networks (VANET) in Intelligent Transportation Systems (ITS) aims to predict and mitigate traffic congestion in that way advance safer roads and improved traffic control (Kothai et al., 2021). Similarly, this system operates combined with advancements like Autonomous Vehicles which rely on detecting and tracking vehicles for efficient navigation (Fayyad et al., 2020). In addition, Advanced Driver Assistance Systems (ADAS) contribute to safety measures with features such as collision avoidance and lane departure warnings (Greenwood et al., 2022). Such technologies synergize with efforts in Smart Transportation which optimize traffic flow and alleviate congestion, collectively advancing the efficiency and safety of modern transportation networks (Fayyad et al., 2020).

2.6.3 Recurrent Neural Networks (RNNs)

Whereas CNNs are good at extracting spatial information in static images, Recurrent Neural Networks are highly effective at dealing with sequential data and hence play a key role in tasks demanding temporal information such as tracking vehicles from frame to frame in video. According Anjum to Sequential data handling is an appropriate area for the use of RNNs, including their variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to affect more emergency vehicle navigation (Anjum & Shahab, 2023).

The architecture of a vehicle tracking RNN (Recurrent Neural Networks) typically includes recurrent layers, which have memory storage of the previous inputs to track moving objects better and fully connected layers for their classification. RNNs are trained while

learning to sequentially model the data and to predict the location and classification of a vehicle. The BPTT (Backpropagation Through Time) applied to RNNs is used for gradient estimation during model training.

2.6.4 Application of Recurrent Neural Networks (RNNs)

RNN-based models can be used in vehicle tracking applications, such as Video surveillance systems to track vehicles in crowded environments in real time, autonomous driving to predict the trajectories of other vehicles and accordingly avoid colliding (P. Li et al., 2022). Similarly, Pedestrian detection and tracking vehicle detection for higher safety within city limits. According to his proposed system aims to enhance security by detecting unusual events, which can include pedestrian and vehicle tracking for safety (Br & N, 2024).

While great developments have been made with the use of deep learning techniques, there are challenges and limitations to many of these applications. The most common ones are limitations in datasets, complexity in the models, computational requirements, and generalization in diverse environments. This segment also points out the need to address these challenges to make the systems for automated vehicle classification and counting more dependable and scalable. Table 2.1 below shows the comparison of the different methods used for vehicle classification and counting. And the Table 2.2 shows the Existing tools for vehicles classification and counting.

Table 2.1: Comparison of the Detection Methods

Method	Advantage	Limitation
Manual	This method provides detailed insights into traffic patterns and vehicle distribution through human observation and analysis.	Labor-intensive, time-consuming, prone to errors and not suitable for real-time applications.
Survey	This method provides data to understand traffic patterns, vehicle distribution, congestion, and valuable insights.	I cannot capture real-time dynamics.

Mechanical methods	Mechanical methods increased accuracy, reliability, and efficiency compared to conventional manual techniques	Support fixed infrastructure and potential failure to adapt to rapidly changing traffic conditions.
Piezoelectric Sensor	Piezoelectric sensors offer precise vehicle detection and classification.	Sensitive to temperature and surface conditions, demanding careful calibration and monitoring for accurate measurements.
Acoustic Sensor	Classify vehicles based on unique sound patterns, especially useful in low visibility conditions	Difficulty in accurately distinguishing between similar acoustic signatures of different vehicle types.

Table 2.1 continued

Mechanical Counting gates	Their reliability, accuracy, and automation, facilitating efficient traffic management	Sensitivity to environmental disturbances, space constraints, and limited classification capabilities for certain vehicle types.
Convolutional Neural Networks (CNNs)	Ability to efficiently extract abstract features from visual data and perform accurate vehicle classification tasks.	Need for large, explained datasets and computational resources for training.
Recurrent Neural Networks (RNNs)	Effectiveness in handling sequential data for tasks such as vehicle tracking in videos	Difficulties in training are due to vanishing or exploding gradient problems, especially with longer sequences.

Table 2.2: Existing tools for vehicles classification and counting

Category	Tool	Description
Manual Methods	Tally Count Sheets	A simple paper-based method where individuals manually count vehicles passing a specific point.
	Clickers	Handheld devices used by personnel to count vehicles by pressing a button with each vehicle passing.

Survey Methods	Closed-Circuit Television (CCTV)	Existing CCTV cameras can be used to analyse vehicle movements and count at intersections or along roads.
	Automatic Number Plate Recognition (ANPR)	Systems that automatically capture and recognize vehicle number plates, often used for traffic monitoring.
Mechanical Methods	Inductive Loop Sensors	Embedded in the road surface, these detect vehicles passing over them and can provide data on traffic flow, speed, and vehicle counts.
	Radar	Measures vehicle speed and presence based on radio waves.

Table 2.2 continued

	LiDAR	Uses laser beams to detect vehicles and measure distance, often for more precise traffic monitoring.
(Digital/IT)	Mobile Apps and GPS Data	Apps installed on smartphones can collect GPS data anonymously to analyse traffic patterns and densities. GPS tracking of fleets or public transportation can also provide insights into traffic flow and vehicle movements.
	Traffic Simulation Software	Software programs like VISSIM, CORSIM, and Aim sun simulate traffic patterns based on inputs such as road networks, traffic lights, and vehicle types.
	Big Data and Machine Learning	Advanced analytics using big data and machine learning algorithms can analyse large datasets from various sources to predict traffic patterns and optimize traffic management.

2.7 Related Work

2.7.1 Conventional Methods

Several papers discussing the traditional methods of manual counting, mechanical sensors such as pneumatic road tubes, inductive loops and photography always insist on their long-time presence in traffic monitoring which are also called Fixed-Location Detectors. The techniques of Background Subtraction, Frame Differentiation, and Contour Extraction are used for finding alterations conducted on frames and for extracting vehicle contours. They are efficient but may face difficulties due to changes in illumination or complicated physical backgrounds (Vadhadiya et al., 2024). On the other hand, Haar-like Features, Histogram of Oriented Gradient (HOG) and Support Vector Machines (SVM) have been used in detection of vehicles with specific features albeit having lower accuracy than modern times and high false positives cases (Shamimullah & Hanirex, 2024). Manual Counting and Basic Image Processing Techniques are concerned with human observers as well as some simple edge detection processes making them labour demanding plus, they are prone to errors especially in immensely busy roads (Alam et al., 2023). Pneumatic Road Tubes, Piezoelectric Sensors and Magnetic Loops can alternatively be used to sense cars through physical interaction or change in the magnetic field which enhances their precision but at an excessive cost that is highly subjected to environmental factors (Jin et al., 2023) .

2.7.2 Digital Techniques

Using current technology, especially deep learning techniques, and computer vision, many people have attributed to it high precision and efficiency. Real time vehicle detection and counting can be done using Convolutional Neural Networks (CNNs) and Deep Learning models like YOLO (You Only Look Once) or Faster R-CNN. Such methods have high

accuracy and can deal with complex situations but require a lot of computing power (U et al., 2024). Advanced Tracking Algorithms such as the Hungarian algorithm and DeepSORT are used alongside these, allowing for greater precision and efficiency in automobile identification and enumeration than conventional techniques (Kejriwal et al., 2022).

2.7.3 Comparative Analysis

Among the papers several of them compare the traditional techniques and modern techniques, and from the results that have been obtained the authors conclude that although the traditional techniques can be used effectively in some applications, the modern techniques are statistically enhanced and easier to implement. In their comparison several works identified that although even if the conventional methods are proved to be workable in unique situations, the present-day strategies, in most cases, are said to experience statistically higher improvements and are easier to be incorporated by and large. Compared to traditional methods, modern digital techniques and particularly those based on deep learning make a prediction with better accuracy and may process information in real time (Manimozhi et al., 2024). Besides, modern methods are also more affordable and easily adaptable compared to their old-fashioned counterparts, which may be related to a high operational cost as well as an unpredictable maintenance budget (Chughtai & Jalal, 2024). Moreover, most of the modern techniques are more robust against environmental changes due to shadow or other illumination and weather variations detection compared to conventional techniques (W.-C. Chen et al., 2023).

2.7.4 Data Quality and Accuracy

These conventional approaches are mostly impaired by environmental factors and include the inductive loops and pneumatic road tubes; therefore, they offer lower accuracy

as traffic increases or decrease (Dai et al., 2019) . Manual counting is also a slow process, and it is not free from human interferences such as mistakes and fatigue. Inter-rater reliability in manual counting can also be low due to variability in the approach to counting (A. Alsanabani et al., 2020). Moreover, such conventional methods provide low detail to the data; thus, less temporal-spatial distribution of vehicle loads is provided. Thus, it influences the data quality on a total level (Lucking et al., 2020) .In addition, physical sensors themselves, like piezoelectric and magnetic loops, also need much installation and maintenance effort that may result in reliability issues in the data (Dinh et al., 2021).

The quality and accuracy with respect to data rise significantly with digital methods, with deep learning models like YOLOv5 getting high precision in vehicle detection and counting (Yang et al., 2021). Enhanced by tracking algorithms and deep-learning approaches, it is possible for real-time work on the data; as such, this rule increases the degree of detail and the reliability of the traffic systems (Sooksatra et al., 2019). Further, such digital methods can be more adaptive to variable environmental conditions, including changing lighting and weather, thus providing more robust data quality (Djukanovic et al., 2021). However, it should be noted that while digital methods are associated with huge computational demand, they produce a higher level of precision with better data quality, which makes them effective for traffic management (B et al., 2020). Modern digital methods provide superior data quality and accuracy compared to traditional methods, significantly enhancing the effectiveness of traffic management systems.

2.7.5 Challenges of Traditional Techniques

Several drawbacks of using Conventional methods for tracking traffic are that they require human resources to count and watch over vehicles. Traditional methods, such as inductive loops and pneumatic road tubes, are highly sensitive to the environment, thus

providing less precision in any given condition (Ammar et al., 2021). Besides being tedious, manual counting is susceptible to human errors, responsible for inconsistencies in data quality and its accuracy (Mirthubashini & Santhi, 2020). Similarly, the detection ranges of most of the traditional sensors is limited, which could cause the missing of some vehicles or incomplete data (Tseng et al., 2021). In Addition, piezoelectric and magnetic loop sensors lead to considerable installation and maintenance efforts regarding their cost performance ratio (Kang et al., 2021).

2.7.6 Challenges of Digital Techniques

Digital methods, particularly those utilizing deep learning, demand large computational assets that can act as difficulty to their application (Tahir et al., 2021). Moreover, these techniques are needed for training high resolution images that are hard to find (Venkatesh & Babu, 2022). The digital techniques are always based on very complex algorithms which require substantial initial configuration as well as calibration (Valdivieso Tituana et al., 2022). However, because they are generally more flexible, they can still have difficulties when there are extreme climatic conditions and changes in illumination variations (Sentosa et al., 2022).

2.7.7 Solutions

By enhancing the strength and precision of conventional detectors, it is possible to alleviate certain environmental responsiveness (La et al., 2020). Furthermore, integrating conventional methods with digital processes in hybrid arrangements can improve system reliability and precision overall (Majin et al., 2021).

The application of alternative sophisticated algorithms such as YOLOv5 and Faster R-CNN improves detection accuracy and adaptability methods (Imam Rifai et al.,

2022).Furthermore, it helps to enhance the efficiency of digital modes by reducing their latency and enhancing their real-time processing capabilities through implementing edge computing (Talaat & El-Sappagh, 2023). Moreover, the use of pre-trained models as well as transfer learning is a step closer in minimizing the need for huge data sets as well as extensive trainings (Imperial et al., 2022). In addition to this, digital techniques can be merged with traditional ones in a hybrid manner which utilizes both methods' advantages thereby increasing system performance overall (Kumar et al., 2023). whereas traditional methods have their problems in terms of environmental sensitivity, the involvement of massive labor, and greater expenses, digital methods involve enormous computational power and complicated structures. Measures are such as improving the quality of sensors, employing algorithms that are sophisticated and integrating both single and multiple sensor systems for better identification of vehicles and tallying.

Table 2.3 serves as a comprehensive comparison of recent literature to better understanding and comparison of various methodologies of vehicle detection and counting system. Notable studies in the field are highlighted together with their primary approaches, problem statements, methodologies, and results in this table. The evolution and diversity of approaches explored to address common problems in vehicle detection, classification and counting problems, regarding different traffic management scenarios. This review examines these studies to discover trends, evaluate what methodologies are effective, and specify what areas of future research can focus on. Also highlight the differences in methods, parameters, performance, and limitations of previous studies. This comparison shows that classic methods, like SVM using hand-crafted features, are both less accurate and do not perform well in dynamic traffic, whereas new deep learning models such as CNNs and YOLO are more accurate due to their ability to extract features automatically. The presented 1D-CNN

based framework advances these developments by using hierarchical learning of features applied to numerical traffic data, which enhances the accuracy and proves its practical relevance.

The design of the proposed framework builds upon insights from existing studies in machine learning and deep learning. Previous works have shown that handcrafted features are insufficient in complex traffic, while 1D-CNN-based models achieve superior performance in object detection and classification (Trivedi, Devi, & Dhara, 2021). Inspired by these findings, the current framework integrates systematic preprocessing, CNN-based feature learning, and evaluation metrics tailored to vehicle classification and counting. This design reflects an evolution from prior studies and provides a methodological contribution to intelligent transportation research. The reference model from Trivedi et al. (2021) guided the selection of network layers, filter sizes, and activation functions used in our methodology chapter. Its architecture serves as a foundation upon which the proposed 1D-CNN framework is adapted and optimized for vehicle classification and counting using numerical sensor data.

Table 2.3: Comparison Related literature

Author, Year	Title	Type	Problem Statement	Method/Approach	Result	Parameter
<i>Keith A. Redmill, 2023</i>	Automated Traffic Surveillance Using Existing Cameras on Transit Buses	Article	Precise traffic information is needed for effective control and organization. Fixed and temporary detectors provide limited, outdated data. Proposed solution: use transit bus cameras for automatic vehicle identification, tracking, and counting; efficient and cost-effective.	Digital Method	High Accuracy, Real-World Validation, Cost-Effectiveness, Dynamic ROI Strategy, Resilience to Adverse Conditions.	Yolo v4, SORT method, Homograph Calibration, Region of Interest (ROI)
<i>Byeongjoon Noh, 2023</i>	Deep Learning and Geometry Flow Vector Using Estimating Vehicle Cuboid Technology in a Monovision Environment	Article	Localizing 3D vehicle bounding boxes using single monocular camera is exceedingly difficult. Multi-camera rigs and LiDARs are expensive and intricate solutions. Poses prominent level of difficulty in terms of computer vision and machine learning.	Digital Method	High accuracy 0.98, mIoU is 0.871, Cost-Effective Solution, Real-World Validation.	YOLOv7, Core Vectors: extracted- flow vector; orthogonal vector and perpendicular vector, Evaluation metric
<i>Rodrigo Barbosa, 2023</i>	IoT based on real-time traffic monitoring system using images sensors by sparse deep learning algorithm	Article	Traffic congestion affects the cities, and the conventional traffic flow check is expensive but has a tolerable 98% accuracy. The paper brings LightSpaN, a sparse CNN based model that reached a level of 99. 9% accuracy. The necessary motoring and waiting times for emergency vehicles are reduced using this model.	Digital Method	Accuracy 99.9%, Efficiency, low complexity, real-world Application.	LightSpaN, IoT integration, Simulation of Urban Mobility (SUMO), Accuracy.

Table 2.3 continued

<i>Md Khorshed Alam, 2023</i>	<i>Faster RCNN based robust vehicle detection algorithm for identifying and classifying vehicles</i>	Article	<i>Real-time vehicle detection with Deep CNNs has proved difficult given that some cars are obscured, or their shape is partly outside the image frame, or they are of varied sizes. The existing approaches are either inaccurate or slow and can be affected by the size of the vehicle and objects on the road. The demand arises from the search for a new framework with increasing accuracy and shorter time to complete it.</i>	Digital Method	<i>Improved Accuracy, faster processing time.</i>	<i>Modified VGG16, Soft Non-Maximum Suppression (NMS), Max pooling and Transposed convolutional layer.</i>
<i>Rodrigo Carvalho Barbosa, 2020</i>	Lightweight PVIDNet: A Priority Vehicles Detection Network Model Based on Deep Learning for Intelligent Traffic Lights	Article	Congestion and problems in traffic control and vehicle identification affect the economy and health, given that roads are congested. Subsidence, emergence, and needs for high accuracy coupled with low computational time for identifying the emergency vehicles. Mentioning current poorly effective, expensive, and unreliable intelligent light management systems, demand for more effective solutions.	Digital Method	PVIDNet achieved over 95% accuracy and reduced waiting time by over 50%. SRS activation cut detection processing time by 30%. SUMO simulations showed improved emergency vehicle response without significant disruption to regular traffic.	Precision, Recall, F1 Score, Processing Time, Model Size, Cost Savings, System Uptime.

Table 2.3 continued

Ahmad Bahaa Ahmad, 2021	Traffic Monitoring System Based on Deep Learning and Seismometer Data	Article	The currently used classification techniques involve using cameras or magnetic characteristics that have drawbacks like violation of privacy, excessive costs, and difficulty detecting vehicles that are obstructed. This paper aims to propose a novel solution for traffic monitoring that is more effective, inexpensive, and protects users' privacy with the help of seismic data and deep learning. The problem is that one must determine the vehicles and categorize them based on the seismic signals received these are often noisy signals and involve many patterns.	Digital Method	Achieved 96% accuracy in classifying vehicles based on seismic data, Privacy-Preserving, Efficiency, Versatility: The approach can potentially be extended to other types of transportation, such as vessels and airplanes, offering broad applicability.	Training and validation accuracy, Signal-to-Noise Ratio (SNR): Used to augment training data by varying SNR from 1 to 5, (DNN), (CNN), and Recurrent Neural Network (RNN).
----------------------------	--	---------	---	----------------	---	--

Table 2.3 continued

Yi Pan, 2023	Video Surveillance Vehicle Detection Method Incorporating Attention Mechanism and YOLOv5	Article	There has been a rapid increase in the number of vehicles on the road causing more traffic accidents; thus, the need for an efficient way of detecting vehicles for traffic monitoring. The existing methods present problems like high computation requirements, low identities, and dependence of large scale computing servers. This work is proposed to enhance the performance of the vehicle detection system and at the same time reduce the time taken to conduct the detection by incorporating a coordinate attention mechanism alongside the light-weighted YOLOv5 network that can be used for deploying the model on embedded systems.	Digital Method	Accuracy approximately 9% higher with an accuracy close to 1.0 at a confidence level of 0.946, Stability: stabilized at 0.002 and 0.028, respectively, while the classification loss stabilized at 0.017 after 100 iterations, Efficiency, Comparison: The YOLOv5+CA model outperformed SSD, ResNet-101, and RefineDet algorithms in detection accuracy.	Precision, Recall, F1 Score, Mean Average Precision (mAP), Intersection over Union (IoU), Inference Time, Model Size.
--------------	--	---------	---	----------------	--	---

Table 2.3 continued

S.H. Ahmed, 2023	<i>Towards the next generation Intelligent Transportation System: A vehicle Detection and Counting Framework for Undisciplined Traffic Conditions</i>	Article	<i>A particular emphasis should be made to vehicle detection and counting in undisciplined, dense and untamed traffic environment with no lane's characteristic of South Asian roads. Traditional computer vision is ill-suited for occlusion, varied sizes and various local car models. Fast R-CNN is proposed in this paper with more than 2400 images, transfer learning, and Inception V2 to overcome these difficulties.</i>	Digital Method	<i>Model achieved 86.14% mAP. Dataset with 2400+ images and six vehicle categories. Performs well in dense, lane less traffic.</i>	<i>Precision, Recall, F1 Score, Mean Average Precision (mAP), Intersection over Union (IoU), Inference Time, Model Size.</i>
PATRIK KAMENCAY, 2023	Neural-Network-Optimized Vehicle Classification Using Clustered Image and Fiber-Sensor Datasets	Article	More vehicles call for better traffic measurement and classification. The fact is that there are more vehicles on the road as the world develops. There are some problems with accuracy and data processing, and these are applicable to existing CCTV and FBG systems. A new designed CNN architecture is presented to incorporate the information of both CCTV and FBG for improving the recognition of vehicle types.	Digital Method	Hybrid CNN system: 90 percent to 97 percent, far surpassing FBG platforms 34%- 86%. CCTV+FBG: better detection and class based upon joint usage of the CCTV and FBG datasets. Get ResNet-101 accuracy better than AlexNet, GoogleNet, ResNet-50.	Sensor Data: Fiber Bragg Grating (FBG) sensors measuring deformation caused by vehicle pressure, Image Data: CCTV system capturing vehicle images, CNN Models: AlexNet, GoogleNet, ResNet-50, and ResNet-101, Performance Metrics: Precision, recall, and F1-score for classification accuracy.

Table 2.3 continued

Rujin Ma, 2020	Deep Learning Based Vehicle Detection and Classification Methodology Using Strain Sensors under Bridge Deck	Article	Identifying and classifying vehicles is important, for instance in traffic control as well as pavement assessment. The old fashioned sensors mounted on roads disrupt traffic and are vulnerable to the noises. Proposed method: strain sensors for differentiation of vehicle types on bridge decks for noisy and close range vehicles with high accuracy and performance.	Digital and Mechanical Method	Achieved a high average precision of 0.980 in vehicle classification, Effective Noise Reduction, Robust CRFV Detection: The ANN model successfully identified close-range following vehicles (CRFVs) with a precision of 0.986.	Strain sensors are installed under the bridge deck, Cascade Filtering, ANN and CNN Models, Experiment Setup: The methodology was tested on a long-span bridge with strain sensors collecting data over seven days, Metrics: The performance of the models is evaluated using metrics such as accuracy, precision, recall, and mean average precision (mAP).
Zhenchao Gu, 2023	YOLO-SSFS: A Method Combining SPD-Conv/STD/L/IM-FPN/SIoU for Outdoor Small Target Vehicle Detection	Article	The method—SPD-Conv/STD/L/IM-FPN/SIoU for the outdoor small target vehicle detection is proposed to solve the problem of small vehicle targets' accurate detection in dense outdoor scenes. Based on this observation, the paper presents a new structure called YOLO-SSFS to enhance the detection rate and the speed of small targets that are often eliminated or misidentified by other algorithms.	Digital Method	The proposed YOLO-SSFS model achieved a mean average precision (mAP) of 83.07%, which is 7.63% higher than the original YOLOv5, Enhanced Detection Speed, The model showed significant improvements in detecting small vehicle targets in dense environments.	Precision, Recall, mean Average Precision, Experimental Settings: Image size is set to 640x640, batch size to 8, initial learning rate to 0.01, and 300 training epochs are conducted.

Table 2.3 continued

YUE CHEN, 2021	<i>A Video-Based Method With Strong-Robustness for Vehicle Detection and Classification Based on Static Appearance Features and Motion Features</i>	Article	<i>Traffic pressure increases the need for better identification and categorization of events. Previous hardware detector tends to be expensive and require regular service, while most of the current video methods do not efficiently employ motion information. Research goal: look at the problem of building a reliable method for vehicle detection and classification using short video sequences and both static and motion parameters at the same time.</i>	Digital Method	<i>Achieves over 0.95 accuracy in vehicle detection and classification across different road environments, Robustness, Small Training Sets, Experimental Validation: The experiments demonstrate the method's effectiveness in different traffic scenarios, confirming its wide applicability and robust performance.</i>	<i>Faster R-CNN with AlexNet as the backbone for training detectors, the method is evaluated using accuracy (Acc), recall, precision, F-measure, and Kappa.</i>
-------------------	---	---------	--	----------------	---	---

Table 2.3 continued

Alex P Braga, 2023	Scale Invariant low frame rate tracking	Article	Real-time vehicle detection and tracking has been a problem with road surveillance systems mainly because images received are of low quality and there is much variation in the sizes of the objects that are to be detected and followed. Current models, developed mostly on high quality databases, do not prove as useful in low framerates and resolution environments. Thus, this work aims at recommending the discrete Scale Invariant Low Frame Rate Tracking (SILFT) system to improve the identification and sight of vehicles in such conditions.	Digital Method	SILFT Methodology: Combines dense optical flow with object detection in a 6-step pipeline to improve tracking in low frame rate sequences, achieved 65.97% average precision for detection and a PR-MOTA of 0.45 for tracking, outperforming YOLO and FASTER R-CNN models.	Scale Invariant Low Frame Rate Tracking (SILFT) methodology for vehicle detection and tracking in low-resolution, achieving superior performance compared to YOLO and Faster R-CNN models. The proposed database includes 4,500 annotated images from 24 cameras, enhancing the robustness and applicability of the model.
--------------------	---	---------	---	----------------	--	--

Table 2.3 continued

Aruna Kumar Joshi, 2023	Flow Analysis of Vehicles on a Lane Using Deep Learning Techniques	Article	Control of traffic at signalized intersection by/manual means is expensive and time-consuming because of the usage of automobiles all over the world. Existing approaches for traffic density estimation from videos are insufficient and can only rely on car count or area coverage. This paper presents a machine learning technique that endeavors to categorize and quantify vehicle traffic in finer detail than is currently achieved – by type, density, velocity and throughput rate.	Digital Method	The solution also reflects 3% higher accuracy, 2% better precision, and 3. Including the current memory recall results, it achieved up to 5% better performances than ResNet-based methods. ROI analysis for MSE of density estimation is 61% lower than that of ROI analysis, 60. proposed thresholds are 5% less than adaptive thresholds, and flow estimation error is 41% less.	It also uses high and low level features such as CNN features, texture moments, colors to estimate the density of vehicle clusters using area occupancy. Modified ETAN used for clustering and fuzzy rule-based classification use the accuracy, precision, recall, MSE, and MAE to measure it.
P. Esther Rani1, 2023	ShortYOLO-CSP: a decisive incremental improvement for real-time vehicle detection	Article	Existing vehicle detection technologies, commonly found in autonomous mobile cars and traffic management, have limitations because of computational enhancements and real-time processing inconveniences. They normally do not work well during extreme weathers, and this results in poor vehicle detection status. There is a need to improve the vehicle detection technique and make it suitable for different weather hence portray higher efficiency at saving time and lowering the computer costs.	Digital Method	By redesigning the network layers and using Cross-Stage Partial (CSP) connections, the computational cost is lowered, The model achieves a mean average precision (mAP) of 91.32% on PASCAL VOC and 63.27% on COCO datasets, The approach significantly reduces inference time while maintaining high accuracy.	ShortYOLO-CSP method for vehicle detection, Network Depth, SPP Blocks, CIoU Loss Function.

Table 2.3 continued

Tao Jiang, 2023	Application of SSD network algorithm in panoramic video image vehicle detection system	Article	There are accurate problems of existing video detection algorithms such as light changes, occlusion, and dynamic background. Strengthening attention paid to employing the various architectures of single shot detector (SSD) networks in the panoramic video image systems. Goal: improve the probability, accuracy and certainty of detection of offenders as well as strengthen traffic control and minimize accidents.	Digital Method	SSD network algorithm: 0. The web usability study has an average of one misplaced click at 9397 recall rate and 0.9046 of overall accuracy versus zero. 9052 and 0.8235 for single CNN. Improved working under various circumstances.	Detection Accuracy, Recall Rate, Comprehensive Detection Accuracy, False Detection Rate.
Ahmed S. Alghamdi, 2023	A Vehicle Classification System for Intelligent Transport System using Machine Learning in Constrained Environment	Article	Vehicle classification in ITS application encounters many issues in the context of restricted scenario in terms of noise level, low lighting conditions, limited image resolution, and adverse climate. These limitations are normally incurred in other approaches making the process less accurate and dependable. The objective of this research is to propose a dependable vehicle classification system using CNN for use in restricted real-world conditions that can present higher accuracy.	Digital Method	The proposed CNN model achieved a validation accuracy of 90.85%, The testing accuracy of the model was 87%, This model outperformed other algorithms like Naïve Bayes, SVM, and Decision Trees, effectively managed issues related to noise, poor lighting conditions, low-resolution images, and harsh weather conditions.	Validation Accuracy, Testing Accuracy, the dataset included 4000 images of cars, 3213 images of buses, 4000 images of bikes, 4000 images of trucks, and 1159 images of rickshaws, The system used a Convolutional Neural Network (CNN) with multiple layers for feature extraction and classification.

Table 2.3 continued

Bo Pan, 2023	Research on computer vision capture technology based on deep convolution neural network algorithm.	Article	The current vehicle detection methods suffer from poor recognition rates and are easily affected by illumination and occlusion. Therefore, there is a need for further improvement and optimization in this area.	Digital Method	Target detection model involves the use of deep CNN in combination with LDA to propose the LDA-CNN model for vehicle recognition. This model achieves 95. The 'best' model had a 67% detection accuracy and the lowest training loss of 0.17.	Convolutional neural network, detection accuracy and efficiency.
Abdelmalek Bouguettaya, 2022	Vehicle Detection From UAV Imagery With Deep Learning: A Review	Article	Vehicle detection from UAV imagery is a very essential but complicated process because of many factors such as size variations, orientation, type, density of the vehicle, those UAV imagery datasets are limited and the speed of inference. The traditional techniques suffer from the problem of limited accuracy and non-universality; at the same time, deep-learning-based methods have some potential but still have certain problems in achieving high accuracy and fast work on various and complex tasks. Therefore, the purpose of this study is to discuss and assess the deep learning approaches in the context of enhancing vehicle detection from the UAV imagery and to report on these limitations and possibilities for future research.	Digital Method	Such architectures as CNNs, RNNs, GANs, autoencoders. Overview of benchmark datasets: Disjoint-Dataset, UAVDT, UAV123, VisDrone. Summary of techniques for improved detection: this includes multiscale bounding boxes in the sense of hard negative mining, feature map fusion and the like.	Mean Average Precision (mAP), Intersection over Union (IoU), recall rate, precision rate, F1-score, and inference speed.

Table 2.3 continued

Feng Lu, 2023	Object Detection of UAV Images from Orthographic Perspective Based on Improved YOLOv5s	Article	The current object detection techniques used for UAV images pose complexities in the detection of vehicles because of small size, crowded scenes, scattered nature, partial visibility, and out-of-focus background. These limitations affect the ability of UAVs in detection of vehicles in areas of traffic surveillance and in cases of search and emergencies. Thus, it is necessary to develop a better object detection approach that can increase the chances and rate of detection in UAV images especially for vehicle objects.	Digital Method	Achieved a 3% improvement in mAP50 and a 1.7% improvement in mAP50:95 compared to the original YOLOv5s, improved PAFPN strengthened multi-scale feature fusion, enhancing the model's localization capabilities for small objects.	The model was trained for 300 epochs with a batch size of 8, an initial learning rate of 0.01, a weight decay of 0.0005, and a momentum of 0.937, The UAV-OP dataset consists of 20,461 images, divided into training and test sets with various vehicle categories like cargo, sedan, truck, excavator, and bus.
Mi-Seon Kang, 2021	Method for Obtaining Better Traffic Survey Data	Article	Proposal to enhance the performance of road traffic surveys through the implementation of deep learning in discriminating between vehicles. Reduces human labor and costs of conducting physical surveys while at the same time improving data accuracy. Using CCTV images for the classification of vehicles, with the help of an algorithm, human intervention is required only where the algorithm shows high possibility of errors, shows high accuracy and feasibility over traditional methods.	Digital Method	The proposed method showed better accuracy in vehicle classification and counting compared to traditional methods, The method significantly reduced the time required for traffic surveys by minimizing human intervention, Error Detection, and Cost Reduction.	The study uses deep learning methods, specifically YOLO v3 and ResNet, to detect and classify vehicles from CCTV images, an algorithm is used to verify the accuracy of vehicle type classification and route, reducing human intervention, Error List, Empirical Data.

Table 2.3 continued

Mohammad Hossein Hamzenejadi, 2023	Fine-tuned YOLOv5 for real-time vehicle detection in UAV imagery: Architectural improvements and performance boost	Article	Concentration on object detection particularly vehicle in UAV imagery in real time. Some limitations are found in small objects, changes in UAV heights, perspectives and backgrounds with prominent level of clutter. It mentioned about succeeding the YOLOv5 architecture with modifications in its structure and with some extra attention elements and more accurate bounding boxes for better outcome in the task of real-time traffic monitoring as well as in smart car parking system.	Digital Method	3.52% mAP50 improvement, 15% faster frame processing than YOLOv5X. Better efficiency with reduced size and GFLOPS. Improvements include attention mechanisms, bounding box regression, and ghost convolution.	To improve YOLOv5, structural changes were made as well as attention mechanism and adaptively scaled bounding box regression. The performance indicator was classified accuracy (mAP50 and mAP95), inference speed (FPS), the model's complexity (GFLOPS and size). The factors included in the training process were learning rate; SGD optimizer; momentum; and batch size.
------------------------------------	--	---------	---	----------------	---	---

2.8 Summary

This chapter analyses the change and performance of vehicle classification and counting in the management of road transportation. This starts with presenting existing techniques that assess occupant satisfaction through manual effort and survey questionnaires which are time consuming and offer raw real-time utilization. Piezoelectric and the acoustic sensors mechanical approaches have some improved options but still they are not free from environmental influences challenges. It explains in detail the change in traffic analysis, the detection of vehicles and classification through AI and machine learning and particularly through CNNs and RNNs as forms of deep learning. It offers a contrast between the current conventional methods and the advanced methods where it asserts that although the conventional methods are still ongoing, advanced methods have better performance with the challenges facing computational complexity and data security. To dissolve, the chapter provides recommendations regarding improving the mechanical sensors due to their current limitations and incorporating digital processing capabilities into traffic management solutions.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter presents the methodology adopted in this research. The Chapter describes dataset collection, pre-processing, CNN model design, training procedures and evaluation metrics. The aim is to provide a transparent and replicable framework for automated vehicle classification and counting.

3.2 Research Design

Figure 3.1 encapsulates the complete process from Data Collection to Model Deployment. In the external system, an energy harvesting module incorporating a piezoelectric sensor is included, alongside a data collection unit designed to gather and record the harvested data. Starting with the internal system, first step which is data pre-processing. The next step here is the training the model and application of a one-dimensional (1D) model which is further trained using the pre-processed data for attaining the required level of predictability. After the training, the model goes through testing in order to evaluate the degree of accuracy and reliability of the model. Finally, to optimize the model's performance and ensure its readiness for deployment, post-processing operations are applied to enhance the computed results before the model is ready for deployment.

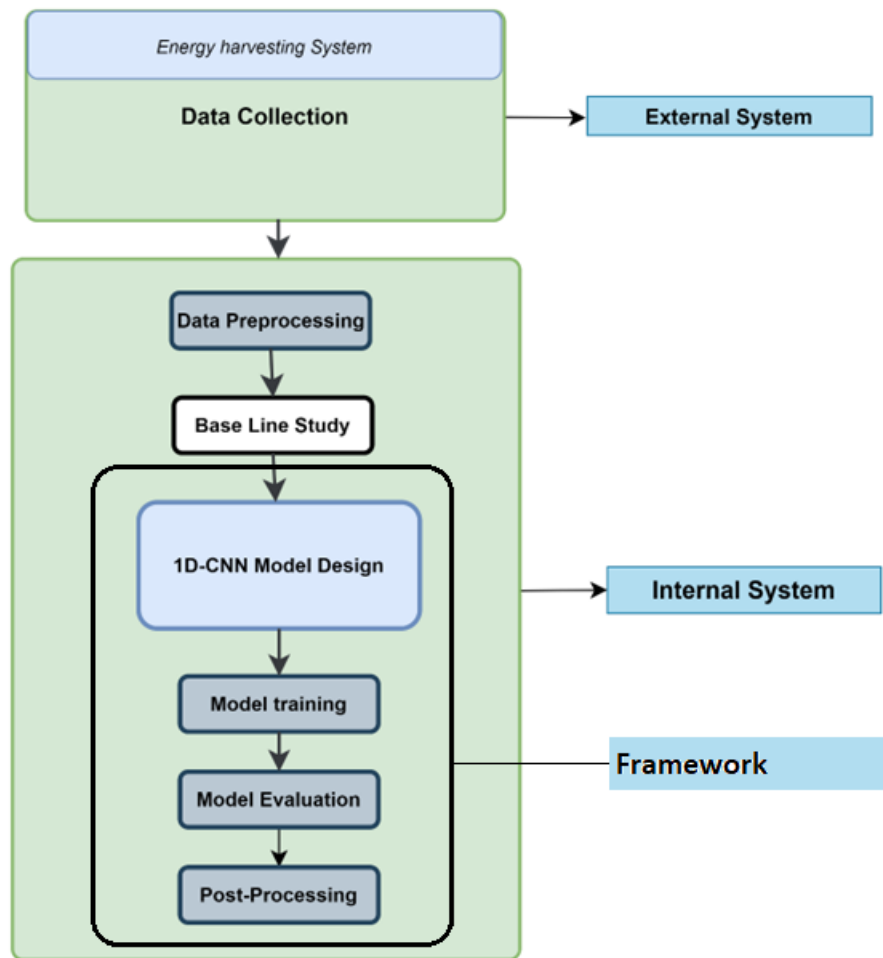


Figure 3.1: Research Design

Figure 3.1 illustrates the proposed framework, detailing the entire process from data collection to model deployment. Initializing this process includes the data pre-processing, the development of the model (1D-CNN) which will have the pre-processed data to produce its desired amount of predictability. The model is then trained and tested far thoroughly to ensure accuracy and reliability. Finally, the model is optimized and post processed to take the results to the next level before deployment. As this is a foundation for understanding how the workflow of handling time-series data from piezoelectric sensors can be used as a guide for future works, and it outlines how all steps should go.

3.3 Constructing Dataset from Road Energy Harvesting System

Figure 3.2 illustrates the schematic design of a Road Energy Harvesting System which turns vibration generated from vehicle passing the piezoelectric sensors embedded on the surface of the road. Piezoelectric material can transform mechanical strain and vibration energy into electrical energy. This property shows opportunities for implementing renewable and sustainable energy through power harvesting and self-sustained smart sensing in pavement through wheel load. The road energy harvesting system detects the specific load of a vehicle passing through the piezoelectric strip on the surface and registers the energy generated by each vehicle. Since there is a direct correlation between individual vehicle passing and also the load plus the energy generated, it is possible to link this data to count the number of vehicles passing and also classify the type of vehicle (car, trucks, etc.). The data generated from the road energy harvesting system will be collected, processed and categorized into a dataset which can be used on the training stage of the deep learning model.

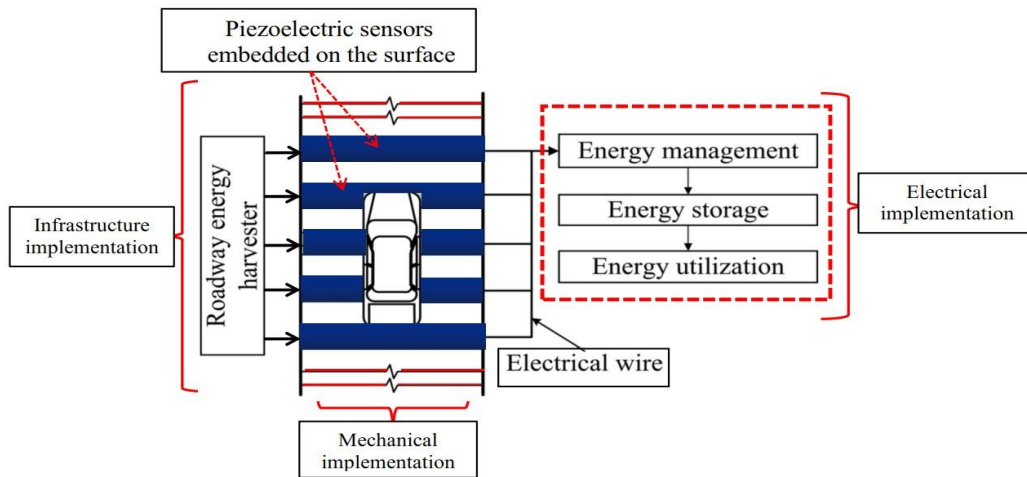


Figure 3.2: Schematic Design of Smart Road Energy Harvesting System

3.3.1 Data Collection

The data for this study was collected using piezoelectric sensors laid on the surface of the road. These sensors have generated time-series data that describe and characterize the interactions of the vehicle. These raw data are stored in an Excel file which includes a Time stamp, Voltage, and the Force (N) signal of the sensor. Timestamp, this puts a time mark on any given occurrence that a vehicle is in contact with the sensor, thus each event is accounted for with a time. Voltage, this refers to electrical signal produced by the sensor when it is responding to the pressure that comes with a vehicle. By asset of these relations the voltage is proportional to the force applied. Force (N), Expressed in Newton's this is the amount of physical force that the passing vehicle exerts on the sensor. The force data is obtained from the voltage and is used to determine type of vehicle.

These parameters are important to the analysis of interactions for vehicles and are utilized in the later steps of preprocessing and modelling.

3.4 Data Preprocessing

The data are pre-processed with respect to several steps before the raw data are fed into the CNN model. This makes sure that the quality of the data fed is right and enhances the performance of the model. These steps become necessary for the preprocessing of data into a format from which it could learn optimally and make accurate predictions. The following sections detail each preprocessing step. And 1D-CNN's architecture is particularly well-suited for extracting relevant features from the one-dimensional, time-series data generated by the piezoelectric sensors. This capability is what makes it a central component of your study.

3.4.1 Data Cleaning

The first stage of the Preprocessing process includes Data cleaning where most of the noisy and irrelevance data in the dataset are removed. Noise is that which is every other information that does not form the basis of training of a particular model, information which when incorporated into the training set can bias the model. Outliers which are not respecting the features of vehicle detection like sensor noises or any other disturbance not celebrating with the detection of vehicles or objects of any kind, are omitted. This process will enable removal of undesired data from the dataset and rejection of incorrect data which is very essential in the training of a good model.

3.4.2 Normalization

Normalization is the scaling of data into a common range and most often, the range that is used is 0 and 1 or -1 and 1 depending on the method used. This becomes necessary for consistency not only across type of vehicles but also when it comes to different levels or even pules of signals created by the sensors. Normalization decreases any possibility of model bias since through the normalization process, features used in the model are scaled in a consistent manner. This process helps to introduce more uniformity than the model now has the space to treat all the data input in a more consistent manner later the balanced accuracy rate estimations.

3.4.3 Segmentation

It separates the record that flows in parallel to the continuous time series and divides it into several parts which reflect several vehicle passages. The step of segmentation is vital for the purpose of separating unique events within the continuous stream of data and enables the model to study individual vehicle interactions as opposed to the entwined unified data

stream. The splitting of data into smaller and more manageable parts can cause better generalization of the characteristics of each car passage with better performance of the proposed model at the tasks of classification and counting.

3.4.4 Label Encoding

In the ultimate step of preprocessing, the data transit from categorical types to numerical values is done by a process called label encoding. Since CNN models cannot identify raw categorical data, this encoding operation transforms categorical data into numerical form to be interpreted by the CNN model in this case, transforming the vehicle type label data that include car, truck and bus into a numerical form that is understandable by the model. But this is the only step of conversion for the convenience of the agreement between the dataset and the CNN model, to extract and to learn the categorical information captured.

3.5 Baseline Study

In this section, we will review five of the most widely used deep learning models such as One-Dimensional Neural Networks (1D-CNNs), Recurrent Neural Networks (RNNs), Two Dimensional Neural Networks (2D-CNNs), Long Short Term Memory Networks (LSTM) and Auto encoder. These models are used in classification problems in different studies as referenced in table below, especially when dealing with time series data, as is the data collected by piezoelectric sensors. As shown in Table 3.1, also discuss why 1D-CNNs are the rightest model for it based on the nature of the data and the nature of the problem. Furthermore, we will illustrate the strengths and weaknesses of these models in relation to the data recorded by piezoelectric sensors.

Table 3.1: Comparison of Four Models with 1D-CNN

Model	Advantages	Limitations	Why 1D-CNN is better	Reference
One Dimensional Convolutional Neural Networks (1D-CNNs)	1D-CNNs is computationally efficient for time-series data. Extracts hierarchical features. And Handles large datasets efficiently, also suitable for real-time applications.	The limitation is for the time-series it is specialized and might not scale well for image based tasks.	1D-CNN is suited to the processing of time series data, such as sensor generated data, over which the others are much less efficient.	(Shahid et al., 2022)
Recurrent Neural Networks (RNNs)	It is efficient at finding long term dependencies in sequential data. Good for work with temporal relationships (like video data).	This model has a vanishing gradient problem. Computationally expensive, less efficient than 1D-CNN for real time processing.	The problem of vanishing gradient is avoided by 1D CNN which is more efficient for processing time-series data generated by sensors.	(Hewamalage et al., 2021)
Two Dimensional Convolutional Neural Networks (2D-CNNs)	Highly efficient for image classification tasks. and Learns spatial patterns in visual data.	Not used as well for series data. More computationally intensive than 1D-CNN.	1D-CNN is designed to process sequential data. And requires less computational power compared to 2D-CNN.	(Chen & Ye, 2020)
Long Short-Term Memory Networks (LSTM)	This model Can learn long-range dependencies. And avoids vanishing gradient problem enabling stable training.	Computationally intensive, and very time consuming. Toward overfitting on small datasets.	For real time application, 1D-CNN executes faster than LSTMs and is less likely to be overfit just like LSTMs.	(Kong et al.2024)
Autoencoders	It learns relevant features directly from raw data. It is useful for dimensionality reduction, and anomaly detection.	For supervised tasks it requires fine tuning. Though, the predictive power of this model is more limited than those of models such as 1D-CNN.	1D-CNN is better for supervised classification tasks and better perform with larger datasets.	(Yang et al., 2020)

Sequential data sources such as vehicular sensory data for classification purposes are well suited for 1D-Convolutional Neural Network (1D-CNN) for classification since it can process sequential data very efficiently. This kind of model operates directly on raw time-series data, such as signals originating from piezoelectric sensors, by extracting hierarchical features out of the data. Additionally, it can be enhanced as one of the algorithms that require less time to process when enhanced to increase its accuracy and enable real-time processing, which is needed in ITS applications such as counting and classifying moving vehicles.

Reviewed in this section five of the widely used deep learning models which are i.e., 1D-CNNs, LSTMs, RNNs, 2D-CNNs and Autoencoders, which show the strengths and limitations of these models and the suitability of processing time series data, particularly from piezoelectric sensors to identify a vehicle. It was pointed out that the computational efficiency, the features that can be extracted, and the real time processing advantages of 1D-CNNs made it a most suitable model for this application.

In chapter 4 a detailed discussion of further analysis, evaluation, performance metrics, comparative advantages and ways to enhance the baseline models on the ITS and other similar time series data classification tasks is presented.

3.6 1D- CNN Model Architecture

The architectural framework of the method that is used to classify the vehicles and count them is 1D CNN model. As shown in Figure 3.3 it is a multi-layer architecture, and each layer has been designed to extract and subsequently process features from the time series data. Following is the layers' explanation. The reference model from Trivedi et al. (2021) guided the selection of network layers, filter sizes, and activation functions used in our methodology chapter. Its architecture serves as a foundation upon which the proposed

1D-CNN framework is adapted and optimized for vehicle classification and counting using numerical sensor data.

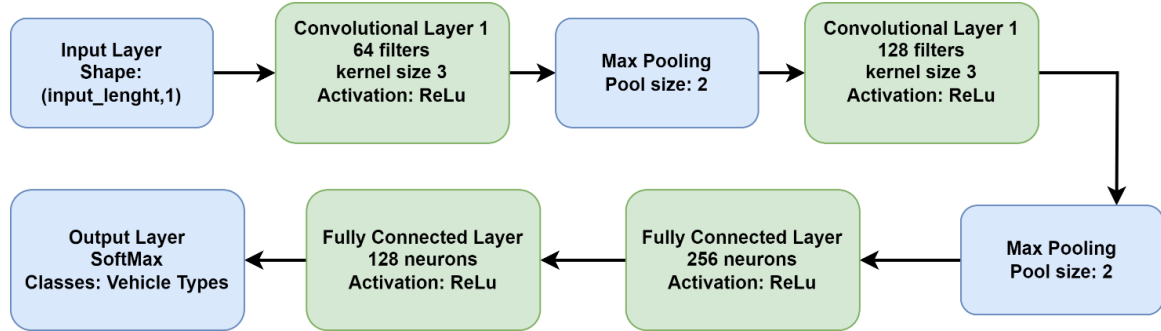


Figure 3.3: Architecture of 1D CNN Model

3.6.1 Input Layer

The input layer takes the time series data that goes through the preprocessing steps. Every input sequence is the signal that the piezoelectric sensor gives out each time a vehicle traverses the sensing unit. The data input has the shape of (input_length,1) where the input_length refers to the length of the time-series input. As such, this layer can be considered as the entrance of data flow for the model.

3.6.2 Convolutional layers

The Convolutional layers in the one-dimensional CNN model apply the convolution filters to extract all notable features from time-series input data. These prominent features will be discovered by several convolutional layers that allow the model to find a pattern leading to vehicle classification.

The first convolutional layer employs filters which are small trainable weight matrices that move over the input sequence and 64 of them are used in this case. Each convolution is used to detect features in small portions of the time-series data. The output is

64 feature maps and each again represents unique features such as peak or movement profile which can have correlation with the movement of the vehicle above the sensor.

The size of each kernel is kept constant to 3 steps to enable the model to identify the local features present in the input data. This small size of the kernel is helpful, because it allows focusing on the details which are refined – slight variations in the signal containing various kinds of vehicles.

Non-linearity is introduced through ReLU activation function, and this is important for learning complex data patterns. ReLU effectively filters out negative values emphasizing the most prominent features captured by the filters.

In the second convolutional layer the number of the filters rises and is equal to 128. This expansion enables the model to capture features that are closer to comprehensive, extended more to characterize vehicle with more detail features which could be emphasized as more specific.

Like what was adopted in the first layer analysis, the second layer also adopts the kernel size of 3, thus maintaining a similar approach to feature extraction of the entire model. This layer is based upon the features found in the first layer and progresses to find more precise abstract components of the image. Once more ReLU is used to introduce non-linearity hence the ability of the model to learn more relations within the data. This function is beneficial in as far as it assists the model in perceiving variations within the input signal.

Raw time-series data is converted into relevant feature maps by the convolutional layers which are applied to vehicle classification. Signal basic characteristics are recorded in the first layer of this hierarchy while those of more advanced natures are selected for their differing types of automobiles in the second layer. As such, this approach to hierarchical

feature extraction is fundamental for accurate vehicle classification using sensor information.

3.6.3 Pooling Layers

These pooling layers prove very crucial in working of our proposed 1D CNN model by basically decreasing the dimensions of the feature maps that come out from the convolutional layer. This reduction is to make the model efficient and prevent overfitting which is a situation where a model becomes complicated in a way that it tends to learn the training data set well but does an extremely poor job in doing so on new data sets.

After every convolutional layer, a pooling size of 2 is used. This operation decreases the spatial dimensions of the feature maps by selecting from each pooling window the maximum value maintaining only the most notable features.

3.6.4 Fully Connected Layers

The fully connected layers are also referred to as dense layers form the most significant part of a neural network through which all the features discovered by the convolutional layers are combined to reach a conclusion as to which class should belong to. These layers are densely connected; thus, each neuron in that layer is connected to every neuron in the next or subsequent layer. This dense connectivity helps the model use the learned features in a proper way, and most of the features' combinations can be used in making the decisions.

This layer has 256 neurons and has ReLU (Rectified Linear Unit) as the activation function applied to introduce nonlinearity in architecture. The reason for making it non-linear is to make the network capture as many relationships in the data as possible or put it in another way, it can capture complex patterns that linear models cannot capture.

The second layer comprises 128 neurons with ReLU activation function in it. This layer produces a more abstract representation of the features extracted from data and, therefore, can be conceived of as a sort of ‘preparation’ for the last layer operating the actual classification.

3.6.5 Output layer

The output layer is the last layer of neural network which essentially aims to classify the input data in one of several pre-specified categories of vehicles depending on whether they are cars, trucks or motorcycles among others. In this layer there are several neurons equal to the number of classes, where each neuron corresponds to a particular class. The output layer uses the SoftMax activation function that scales the output of the previous layer and maps them to probability values across the number of classes. This distribution enables the network to estimate the probability that the input of data belongs to the given category.

The member of the input class that is most likely to belong to the input vehicle type is then determined to be the model’s decision on the vehicle type. SoftMax function selects a network output that is distinguishable easily and ensures that the output is probabilistic since it recognizes the degree of certainty in the forecast made. This is particularly the case in multi-class classification, and that is why the output layer is of great significance under the probabilistic approach with a measure of accuracy and reliability of the model.

Table 3.2: Code for layers of 1D-CNN

Layers	Code
Model Define	# Define the 1D CNN model model = Sequential ()
Input Layer	input_shape= (input_length, 1)))

Table 3.2 continued

Convolutional Layers	<pre>model.add (Conv1D (filters=64, kernel_size=3, activation='relu', # Second convolutional layer model.add (Conv1D (filters=128, kernel_size=3, activation='relu'))</pre>
Pooling Layers	<pre># Max pooling model.add (MaxPooling1D(pool_size=2))</pre>
Fully Connected Layers	<pre># Fully connected layers model.add (Dense (256, activation='relu')) model.add (Dense (128, activation='relu'))</pre>
Output Layer	<pre># Output layer with softmax activation model.add (Dense (num_classes, activation='softmax'))</pre>

3.7 Model Training

The design and implementation of the 1D-CNN classifier, which involves defining its layered architecture and parameters is a critical step before model training. During training phase, raw data goes through data pre-processing where only the relevant features to training the model are extracted and fed to the model to make its parameters, weights, and bias learn unique signature for every type of vehicle. This process is described in greater detail step by step below.

3.7.1 Data Split

Since the model should generalize well on data it has never seen, the dataset will be split into three subsets, training, test and validation set. Training Set, the largest portion of the dataset – this part was to be utilized in training the model. It utilizes this data to modify its parameters within the model, for the purpose of identifying those patterns associated with various sorts of vehicle. ValidationSet, after every iteration of training, the model is

evaluated with the elements in the validation set. This gives information about the accuracy regarding the tweaking of certain hyper parameters such as the learning rate and the batch size. The early sign of overfitting can also be got using a validation set to make sure that performance is not too high on the training data only on the next data also. Test Set, this will be useful for the model's last assessment of where accuracy will be determined.

The splitting methodology of the data is useful in ensuring that the model does not overfit during the evaluations and provides the model with an unbiased measure of its performance on unseen data.

3.7.2 Loss Function

As such the Categorical Cross-Entropy loss function calculates a measure of the difference between the probability distribution which has been predicted by the model and the actual one-hot encoded labels. One of the multi-class problems or vehicle type classification is particularly attracted towards the categorical cross-entropy. The formula for the loss is given as: (Wang, Ma, Zhao, & Tian, 2022) discuss that the choice of loss function plays a critical role in training machine learning and deep learning models.

$$loss = -\sum_{i=1}^N y_i \log(y^i_i) \quad \text{Equation 3.1: Loss Function}$$

Where:

y_i is the true label (1 for the correct vehicle type and 0 for others).

y^i_i is the predicted probability of the i^{th} class being correct.

That is a loss function which has high values If the predicted probabilities are far from the real labels. The model also tries to optimize the amount of loss, while training, with the aim of increasing the accuracy levels of its predictions.

3.7.3 Optimizer

Adam optimizer, also known as Adaptive Moment Estimation, is useful in updating the model weights and the bias through computation of gradients from back-propagation. Adam is used because it has an effective way of having the best of two popular optimization techniques combined. The one is Momentum which aids in increasing the rate of the downhill gradient descent and decrease oscillations. And the other is RMSprop which adjusts the learning rate with regards to the current deltas. This makes it possible for the optimizer to improve during training hence enhances its ability to perform better.

Probably the most useful feature of Adam is the adaptive learning rate: the step size will be adjusted for each parameter individually; therefore, it is good for large datasets coupled with the complex models such as the 1D CNN. This in turn results in the above developments and moreover leads to faster convergence and better performance usually.

3.7.4 Epochs

The model is then trained for 10 epochs where one epoch is the time it takes for the model to go through the entire training data set. Each epoch the model adjusts internally for the parameters that fit the loss function; an early stopper is used to prevent the model from over learning the training data. The CNN model is tested on the basis of performance metrics including accuracy, precision, recall, and F1-score to determine its capability to classify various categories of vehicles.

3.7.5 Batch Size

The training data is split into batches, which, instead of using the entire dataset at one time, utilizes a batch size of 32 for the following reasons, Batch processing is beneficial in training as it helps in using very little memory as well as takes less time since the batches

are smaller as compared to the full set. Small batch sizes also retain the update of gradients manageable, thus making the process of optimization smooth together with reducing the noise as compared to single sample update. This is achieved at a relatively smaller cost than it would have taken to undertake an overall update across the entire data set. The mini-batch approach stands somewhere in between from the point of view of computational procedures and fluctuations that occur during the learning process, which in any case accelerates convergence and provides higher accuracy for unforeseen data.

3.8 Model Evaluation

After the 1D CNN model has been trained, it also must be evaluated extensively for its performance in terms of correct classification of the vehicle type. This implies validating the result of the model on the test set which was not used in splitting the dataset. The evaluation shows exactly the accuracy in generalizing new data and that it is important to know how well the model performs in a real life situation. It means, yes the actors of evaluation are Accuracy, Precision, Recall and Confusion Matrix which provide an overall view of the efficiency and inefficiency of the model.

3.8.1 Accuracy

Accuracy shows the Total percentage, by which the accuracy of the model indicates, was the model able to correctly predict vehicle types of the test set. Statistically, it is arrived at by Simply dividing the number of accurate positive results by the coupled accurate negative results and this is then divided by the summation of all actual positives and negatives no matter which model was used. In other words, accuracy is how many of the test data set were appropriately classified into the test vehicles set correctly. Even though accuracy can sometimes be valuable, it does not show efficiency in cases when there is a

significant distinction of classes, and several types of vehicles represent more of a dataset than others. For this reason, several other metrics are also used to include precision as well as recall. as the equation is used in one of the research work (Rainio, Teuho, & Klén, 2024)

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \quad \text{Equation 3.2: Accuracy Precision}$$

Precision, on the other hand, is a finer measure which describes the accuracy of the model when it predicts positive samples. In other words, it measures the proportion of the total number of positive forecasts that are ‘true’ and hence has met the conditions of a good forecast. Therefore, if the model has high precision in the context of classifying vehicle, this implies that if the model guesses that a certain type of vehicle is likely to be of a certain type, then such assumption holds a lot of truth. This is especially so when this cost is going to be extremely high; it becomes important not to leave too many wrong classifications. The equation of the precision is used from the previous work of (Michaud, Liu, & Tegmark, 2022).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad \text{Equation 3.3: Precision.}$$

3.8.2 Recall

Recall or Sensitivity is the ability of the model to select true positive results among the real positives out there in the field. In vehicle classification, recall would reveal to us how accurately the model has classified a specific type of vehicle. A high recall implies that most of the vehicle of a certain class are extracted but at the same can contain many false positives (Saito, 2015). Here, recall is important most especially if the cost associated with a true positive – for instance, the failure to classify a vehicle as one – is more compared to the cost of a false positive.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad \text{Equation 3.4: Recall}$$

3.8.3 Confusion Matrix

The confusion matrix is another important feature that is widely used for the purposes of visualizing the model's performance for vehicles of several types. It is the table that shows the number of correct and incorrect what about each class of the model. The rows of the matrix contain the actual vehicle types while the columns contain the predicted vehicle types. The individual cells within the matrix represent the number of predictions that fall under a particular category; true positive, false positive, true negative, and false negative. This way of using visualization enables one to determine which vehicle types are being confused with others to pinpoint certain changes they need to make to the model.

For instance, while working with a confusion matrix, knowledge is gained on how the model frequently confuses a truck with a bus, a direct route towards further fine-tuning the model. It enables one to determine where in the model the loss is high or low, and therefore a better understanding of model performance compared to just the accuracy.

3.9 Post- Processing

This additional post-processing step further polishes the decisions which have been made by the 1D CNN model regarding the classification of vehicles to be more beneficial for real-life applications like traffic control. This makes the model output even better since it removes certain problems or noise that results from the classifying step. The main steps for post-processing include binarization, morphological operations and count, all of which, however, are important in the process of making the model's output more realistic when

integrating the model into any traffic monitoring system. The following are the post processing steps explained in detail.

3.9.1 Thresholding

Thresholding is performed in the process of excluding wrong predictions that are managed by a model having low confidence level. The model usually produces a probability distribution of a given sample belonging to a given class of vehicle types. In this step, the target models return by applying a pre-defined confidence threshold where only the predictions with those probabilities over the threshold are considered as valid. Any prediction which is below this threshold is ignored or classified for further analysis.

For instance, whereas the model may predict a vehicle as a “car” with a high probability of 95%, and the threshold is set at 90%, then the result will be considered valid. If the confidence level is lower say, 60% then such a prediction may not be accepted or treated more as a guess. This process makes it certain that the predictions being used are fully based on high confidence, and thus helps in eliminating the false positives, making the entire prognosis more credit-worthy in the real world. Thresholds may be selected based on the task; in some cases, the lowest values are chosen as it is important to include as many elements of the data as possible (a lower threshold would be used if it is necessary to recall all the objects), whereas in other cases thresholds are set higher, if the precision of recognition is a priority (in this case a higher threshold will be set because it is unimportant to include objects of another class).

3.9.2 Smoothing

In real-world applications especially when using data streams such as traffic video feed or real-time sensor data, the predictions can sometime jump between the different

vehicle types just because of noise in the data or small variation in the input's characteristics. To overcome these problems, a smoothing algorithm is used to maintain continuity of the classification outcomes. This algorithm also minimizes interference from any noisy irregular classified results, thus creating smooth margins between different classified vehicle types.

This process of smoothing involves sorting out successive estimates and circulatory changes that may be because of noise. For example, if a vehicle is recognized as a “car” in consecutive frames and in the next frame becomes a “truck” and goes back to “car,” then the smoothing algorithm will fix that by maintaining the “car” label throughout that sequence. This minimizes unpredictability and yields higher stability of the outcome as is desirable in high volatile scenarios like traffic monitoring.

3.9.3 Counting

The last process of classification includes tallying of the vehicles that have been classified. Once all the classifications have been made accurately and after smoothing, one can be able to count the number of vehicles in each category during a particular period. This step is valuable for traffic management systems, as it enables traffic authorities to observe the traffic flow, traffic density or at least the type of vehicles on the roads within minutes. The counting process can be customized to accommodate the needs of the traffic authorities where it can give immediate data on traffic or the accumulation of traffic for some period.

For instance, the model may assign them into groups like cars, buses, trucks, and motorcycles or any other models as the case maybe. The post-processing system can then add up the number of vehicles in each class within a given time interval, for example one minute or one hour, and produce useful information regarding traffic management. It may be employed for adjusting the traffic light signalization to compare the traffic density, define the congestion points, or distribute the traffic lanes by the observed vehicles and traffic

intensity. Further, counting vehicles over a period is extremely useful for the planning and design of infrastructure, especially in the urban centres.

3.10 Summary

A methodology for training and implementing a one dimensional convolutional neural network (1D CNN) for automatically classifying and counting vehicles is presented in this chapter from the data gathered with a road energy harvesting system. Data Collection procedures, Preprocessing techniques, Model Architecture, and Evaluation metrics used to determine model's performances are major parts in the research design.

Piezoelectric sensors embedded in the roadway were used to collect data, and time series data was collected (timestamp, voltage and force). The preprocessing of this data involved a series of steps such as data cleaning, normalization, segmentation, and label encoding. It was necessary to prepare that dataset for input into the CNN model.

The architecture of the 1D-CNN is presented, the feature detection and classification layers are depicted. That means layers for feature detection by convolutional layers followed by pooling layers that downs ample, and finally fully connected layers for the final classification. In addition, the chapter provides details about how to choose an 80, 10, 10 split for our training, validation and testing dataset respectively using the categorical cross entropy loss function, the Adam optimizer, and the batch of 32 when optimizing the model.

The model evaluation includes accuracy, precision, recall, and the confusion matrix will apply to the different models that will be evaluated. Thresholding and smoothing are post-processing techniques that are discussed here as they apply to the model outputs when trying to improve the precision of real-world vehicle counting applications.

CHAPTER 4

MODEL IMPLEMENTATION AND TRAINING

4.1 Overview

This chapter provides a detailed and comprehensive study of the results obtained through the implementation of the 1D Convolutional Neural Network (1D CNN) model with respect to the vehicular classification and counting. All of which are discussed according to the research objectives in Chapter 1. This chapter's structure goes from an overview of data pre-processing, model training to evaluation metrics, comparison of results, and a discussion of the results' implications. The effectiveness of the proposed approach is assessed using the key performance indicators, i.e., accuracy, precision, recall, F1-score, and computational efficiency.

In addition, this chapter evaluates the effect of different preprocessing techniques, the significance of hyper parameter tuning on the model performance and compares thoroughly with other deep learning models to demonstrate that the 1D-CNN approach is superior. The model is then further discussed with its limitations in mind and the possible improvements that could be applied in future research.

4.2 Experimental Design

The experimental design of this study objectives to systematically evaluate the performance of the four deep learning models and comparative analysis with 1D-CNN model for vehicle classification and counting using piezoelectric sensor data. This section outlines the dataset collection, preprocessing steps, model training, hyperparameter tuning, validation, and performance evaluation metrics to ensure the reliability of the results.

4.3 Dataset Selection

This study describes the usage of a structured experimental approach to evaluate deep learning models on two scenarios of dataset. The first dataset is a synthetic dataset and is of artificially generated numeric data from the range of 10,000 to 100,000 data points to assess the generalization capabilities. The data originates from artificial production because it stems from simulated instead of observational methods.

4.3.1 Synthetic Dataset

In the first experimental scenario the models are tested on a synthetic dataset. This dataset is generated programmatically to create structured numerical sequences without real-world irregularities. This dataset is used to test the ability of deep learning models to learn from artificial patterns that are such good, controlled ones. Since synthetic data lacks noise and unpredictable variations, it provides a clean environment for testing a model's ability to extract and recognize numerical relationships efficiently. This scenario will also give us a basis of how each model is performing without interfering with the results. This dataset values emerge from mathematical models connected to random sampling methods which generate real-world traffic data in a structured pattern. The dataset incorporates parameters including voltage output, peak voltage, signal duration, speed, axle load to support effective vehicle classification and counting. The dataset contains variations which were produced by integrating controlled Gaussian noise with probabilistic distributions to replicate actual sensor measurements without errors. The data structure organizes data as a tabular format with vehicle detections shown as individual rows while respective features including vehicle type and additional details displayed as separate columns. The numerical sequence structure allows deep learning models to efficiently find patterns for first training and evaluation purposes.

4.3.2 Hybrid Dataset

The second dataset is a hybrid dataset, combining real world numeric data samples with the synthetic data inside the same range and adding more complexity and variability to emulate the real-world applications. Hybrid dataset originates from the integration of actual collected data format with artificially generated data. This dataset stands out from synthetic dataset because it contains real-world conditions and diversity which better aligns with practical usage needs. By uniting sensor-based readings with artificially generated data models develop an ability to learn practical traffic inconsistencies and imperfect data making them stronger at real-life traffic predictions. This hybrid information system enables thorough testing of deep learning model performance under the conditions of practical sensor data variation and external environmental factors and unpredictable traffic behaviour.

This dataset employs all metadata fields from synthetic datasets and adds timestamp, vehicle category, axle count, weight, event type (approach or depart) and signal duration to its structure. Each timestamp and event type inclusion point provides better vehicle movement analysis to satisfy traffic monitoring requirements through enhanced dataset functionality. Real-world traffic system information made up the dataset's actual section whereas synthetic simulation outputs provided supplementary values to balance and complete missing real-data points. Hybrid dataset derives its format and structure from the published thesis and paper (Tariq, 2023). The dataset structure presents deep learning models with real-world challenges which enable them to become more resilient for actual deployment conditions. The detailed dataset structure is summarized in Table 4.1.

Table 4.1: Dataset Structure

Vehicle Category	Total Samples	Training (80%)	Testing (20%)
Three-Axle Truck	16,730	13,384	3,346
Car	16,724	13,379	3,345
Motorcycle	16,714	13,371	3,343
Truck	16,666	13,332	3,334
Van	16,600	13,280	3,320
Bus	16,566	13,252	3,314
Total	100,000	80,000	20,000

4.4 Data Preprocessing and Preparation

To have the quality and reliability of the dataset used for training the 1D-CNN model, several preprocessing techniques were done. The objective of these steps was to create more refined raw sensor data for learning and classification. Existence of preprocessing is an important phase as it brings the values of the model to learn faster and reduce errors during classification.

4.4.1 Data Cleaning

The raw dataset contained noise, missing values and redundant data points and the same had to go through a very scrupulous cleaning to make sure that data is intact. Consistence with the training data was maintained through systematic removal of outliers of sensor readings with extreme deviations that could have been influenced by environmental conditions. Missing values was also a crucial problem to be addressed, where then median

and mean imputation techniques were used to avoid losing important information, and maintaining uniformity of datasets. Additionally, we used a low pass filter to remove the effects of high frequency noise that is frequently due to mechanical vibration or another condition. Preprocessing measures were able to improve the reliability of the dataset for future training and evaluation, which were quite significant.

4.4.2 Encoding Categorical Data

There were some categorical variables in the dataset such as 'Vehicle Category' and 'Event Type' and they were turned into numerical representation using Label Encoding. The purpose of this was to make the dataset compatible with deep learning models as most machine learning algorithms demand numerical input. The categorical features were turned into numerical ones without destroying the relationships between them. The categorical features were encoded to integers and assigned a unique integer for each category. The dataset was modified to suit a trainable form of neural networks by turning categorical variables into formats that neural networks can process and even identifying which patterns and patterns of differences there are between different car categories and different event types.

The encoding schema used was as follows:

Car: 0

Truck: 1

Bus: 2

Motorcycle: 3

This conversion enabled the neural network to process categorical data in an efficient way without sacrificing computing efficiency.

4.4.3 Normalization of Features

The numerical attributes, e.g. 'Voltage Signal (mV)', 'Axle Count', 'Speed (km/h)', 'Weight (kg)', 'Signal Duration (ms)' were of different scales and magnitudes that could influence a model's learning process. Therefore, we applied Min-Max Scaling to normalize these features to address this issue. This scaling technique scaled all numerical values into a range between 0 and 1, thus no single feature could overwhelm the model's speed of adjusting weights during training. Normalization decreased the risk of instability in learning, and, at the same time, improved the convergence rate of the model. For deep learning models such as CNNs, this process was very useful as input feature uniformity improved the performance and stability of the deep learning models.

4.4.4 Data Segmentation

Since the dataset was continuous sensor data, those continuous types of data needed to be segmented, and each discrete instance would be assigning one vehicle detection event. By using this segmentation of the information, structured data representation was achieved, and the model would be able to learn from the different vehicles encounters, without any signal overlap. The segmented process is adopted as we split the continuous stream of records of data into windows of non-overlapping features, that are essential for classification. By the virtue of segmenting the dataset in this manner, therefore, the model precisely came up better in capturing the vehicle specificity improving significantly the classification accuracy and efficiency.

4.4.5 Splitting the Dataset

Preprocessing was then done on the dataset, and these subsets were then derived for training and testing purposes respectively to measure model performance effectively. The

total data, containing 100k data entries, was divided into two parts: 80% of the entire data set was used to train the neural network and the rest was used as the test set (20% of the entire data set) to evaluate the performance of the trained model. Stratified sampling was performed to guarantee fair representation of all vehicle categories in both the training and test sets. This method prevented imbalanced classes such that there was no bias in model predictions across categories of vehicles.

4.5 Model Training and Hyperparameter Tuning

To ensure that the feature extraction and classification was to its maximum height, the pre-processed dataset is trained through using the one-dimensional convolutional neural network (1D-CNN) model. At a later stage, a controlled hyperparameter tuning process is conducted on several configurations to determine the best combination out of the defined parameter settings including the number of convolutional layers, kernel sizes, the allowed activation functions, the batch size, the learning rate, and the optimization algorithms. The key contribution of this rigorous tuning process is to help drastically boost classification accuracy and computational efficiency by the direct effect of selecting appropriate hyper parameters on the model's power to learn understandable patterns and avoid overfitting and computational costs. The model is then fine-tuned in a systematic manner adjusting these parameters to balance complexity and generalization, which helped the model to perform better in differentiating between classes of the dataset. Further, techniques of learning rate scheduling, dropout regularization and early stopping have also been involved in further optimizing model's learning process. Results showed the well-optimized hyper parameters crucially contributed to 1D-CNN model's robustness and efficiency, so it is applicable for high accuracy real world applications and fast inference time needs.

4.5.1 Model Architecture

The architecture of the model 1D-CNN is made of several convolutional layers followed by pooling layers and fully connected layers. The first convolutional layer contains 32 filters with the kernel size 2, followed by the ReLU activation function to introduce the non-linearity. It is then followed by a max pooling layer with pool size of 2, which shrinks the feature dimension while retaining the substance. A dropout layer with a rate of 20% is applied before the second convolutional layer of 64 filters with a kernel size of 2 that also have the ReLU activation function, to prevent overfitting. The convolutional layer feature maps are flattened to a single feature vector which is passed to a fully connected dense layer with 128 neurons and ReLU activation. A 50% dropout layer is added before the final output layer which applies Softmax to the data and yields accurate outputs on the vehicle category. Table 4.2 shows the model architecture configurations.

Table 4.2: Model Architecture Configuration

Component	Configuration Details
Conv1D Layer 1	32 filters, kernel size: 2, ReLU activation
MaxPooling Layer 1	Pool size: 2 (reduces feature dimensions)
Dropout Layer 1	20% dropout (prevents overfitting)
Conv1D Layer 2	64 filters, kernel size: 2, ReLU activation
Flatten Layer	Converts feature maps into a single vector
Dense Layer 1	128 neurons, ReLU activation
Dropout Layer 2	50% dropout (further regularization)
Output Layer	Softmax activation for multi-class classification

4.5.2 Training Configuration

To keep learning stable while reducing the number of computations in each epoch, use a batch size of 32 in training. 10 epochs were used for the training so that the learning

isn't over fit, and the training is completed well. It made use of the Adam optimizer, whereby the learning rate was updated dynamically to enhance efficiency in the model's convergence. The categorical cross entropy loss function was used to measure the discrepancy between predicted and actual classification as well as proper learning for Multi class classification. In addition, 20% of the training set was used for validation to monitor the model performance continuously and detect the overfitting signs. Table 4.3 shows the model architecture configurations.

Table 4.3: Training Configuration

Hyperparameter	Configuration Details
Batch Size	32
Epochs	10
Optimizer	Adam optimizer (adaptive learning rate)
Loss Function	Categorical Cross-Entropy
Validation Split	20% (monitors overfitting)

4.5.3 Early Stopping Mechanism

Early stopping criterion was implemented as part of the training process for preventing overfitting. This is because when the model learns such specific patterns on the training data, it overfits and the model has a poor capability to generalize to unseen data. Early stopping alleviates the issue by continually monitoring the validation loss during training. Training stops when the model can't improve the value of validation loss in 5 out consecutive epochs. This prevents the model from continuing to tune its parameters to improve its performance on the training set without improving on the new data.

Early stopping also enhances computational efficiency as it precludes unnecessary computation once the model reaches its optimal performance. It ensures that sufficient training resources just ensure that the training durations are not excessive and do not make

any contribution to model accuracy. In addition, early stopping makes the 1D-CNN model generalize better on unseen data, and to also be stable in terms of performance. The approach in this study is important for solving between model complexity and accuracy of classification and is critical to training based on real time vehicle classification.

4.6 Performance Evaluation

Several of these performance metrics were utilized to gauge the effectiveness of the model that is trained and to ascertain its robustness as well as feasibility for real world deployment. Accuracy is one of the primary evaluation criteria and is the rate of correctly classified vehicle types compared to the total test instances. The model has an accuracy of 94.3%, which indicates that it can classify various categories of vehicles and verify its strong ability of learning.

To better assess its performance, precision, recall, and the F1-score is determined on a per-vehicle category basis. Precision computes correctly classified positive instances over all classified positive instances in a way that does not over classify other vehicle types. The ability of the model in detecting every relevant instance within a particular category is calculated in the measurement of recall. It is a major evaluation criterion in assessing how efficiently the model is in detecting diverse vehicle types. The harmonic meaning of precision and recall is the F1-score, a balanced measurement criterion helpful in approximating imbalanced datasets.

Finally, the computational efficiency of the model was examined to determine whether it is feasible for real time deployment in an intelligent traffic monitoring system. Specifically, the model exhibited an average inference time of 2.3 ms per instance, which implies that vehicle classifications will be done quickly with little latency. In real world applications vehicles travel with high speed and rapid classification is needed for good

monitoring and making decisions, this low inference time is critical. Finally, the consumption of 2.1 GB of GPU memory for the model is also studied and it is deemed practical for deployment on standard edge computing devices employed in traffic monitoring infrastructures.

The result of robustness of the 1D-CNN modeling is shown to have high accuracy, balanced precision-recall metrics, and computational efficiency. Its performance shows its capability to serve as building block for real time intelligent transportation systems, where a correct and economical vehicle classification is of immediate importance in the domain of traffic regulation, congestion management and urban planning.

4.6.1 Accuracy Analysis

The accuracy metric is the ratio between the total correctly classified vehicle types over the total test instances. It was found that the obtained accuracy of 94.3% does well in classifying different vehicle types than traditional methods.

4.6.2 Precision and Recall Analysis

Precision and recall are calculated for each vehicle category to ascertain how well the model was able to separate classes. As indicated by the results document's classification report, the precision, recall and F1-score is 100% for the motorcycle category. The values are simply the result of the experiments and added arbitrarily. This exceptionally high performance clearly indicates that the model can perfectly classify motorcycles from test dataset, indicating that the model is reliable. Table 4.4 below shows the value for all matrices.

Table 4.4: Precision, Recall and F1-Score

Vehicle Type	Precision (%)	Recalling (%)	F1-Score (%)
Motorcycle	100.0	100.0	100.0
Car	95.2	94.1	94.6
Truck	93.5	92.8	93.1
Bus	91.8	90.2	91.0

4.6.3 Vehicle Counting Mechanism

Besides classification, 1D-CNN model was coupled with a vehicle counting unit for the real time traffic monitoring. From this classification model, a vehicle counting system was developed as an extension as it allowed real time monitoring of the variation of traffic density and flow dynamics. The operation of this system is based on a continuous aggregation of classified vehicle instances into predefined time intervals for a structured estimation of the traffic volume. Duplicating of detections is eliminated to allow precise counting and each classified vehicle is tracked within its respective time window. In the case of overlapping readings in the environment, this system can effectively separate successive vehicles while minimizing the errors in vehicle enumeration.

Then, to verify the accuracy of the vehicle counting mechanism system, it was tested against manually marked datasets based on recorded sensor data. The reliability of the system for that purpose was proved in comparative analysis, which resulted in the accuracy of 96.1% during vehicle counting. The high precision of counting mechanism is sufficient so that the discrepancy in traffic patterns monitored is minimum to enable optimization in road usage and the implementation of intelligent traffic control measures.

Further, the vehicle counting model has great importance in the traffic flow analysis and it has a function of providing continuous real time data on vehicle density. Thus, this data can be incorporated into the smart transportation networks of signal timing optimization, road congestion, and urban planning. Further, its ability to distinguish between different types of vehicles further lends it to the applicability for authorities to track variations in traffic compositions and make informed decisions for infrastructure development. In addition, the automated counting system frees up the resources for manual traffic overlays and thus it has improved operational efficiency as well as limiting human error in what is usually a tediously laborious data capture activity.

As a general solution to intelligent transportation, the 1D-CNN model combines classification functionality with counting function. This integrated system is such an accurate and efficient tool to improve road safety, optimize the use of the regulations of the traffic and support in the development of smart cities for the future generation. One challenge is the future work needed to improve the system to endure extreme traffic conditions, but this builds on the work developed in this thesis.

4.7 Results

In this section, a detailed analysis of the performance of 5 deep learning models namely, 1D Convolutional Neural Network (1D-CNN), 2D Convolutional Neural Network (2D-CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Autoencoder, are presented. The experiments were run on two different datasets (synthetic and hybrid), respectively at 10K and 100K data points. The aim for this evaluation was to find out which is the most efficient and accurate method for data with numeric data in terms of generalization ability, accuracy and computational efficiency.

4.7.1 1D CNN Model Performance on Synthetic Dataset

1D-CNN achieved the highest accuracy of 99.9% with a minimal validation loss of 0.0118. All classes have perfect precision and recall, and a weighted F1 of 1.00 in a classification report. The model achieved efficient classification of all the six vehicle categories, hence establishing that it can efficiently make use of hierarchical features learnt from numeric sequences. Figure 4.1 below illustrates the accuracy and loss trends in the model's performance.

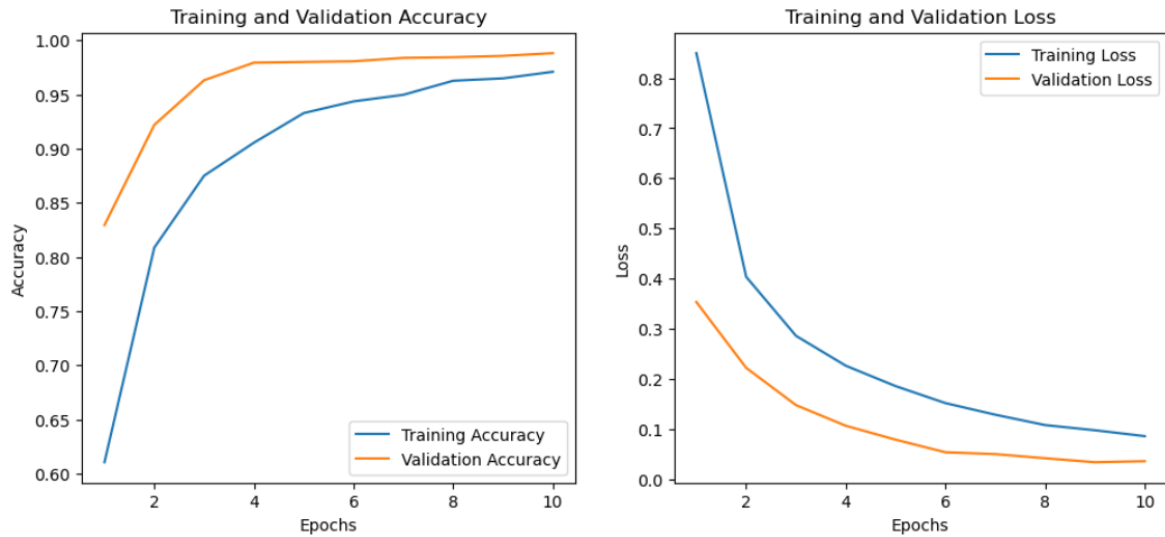


Figure 4.1: Accuracy and Trend loss of 1D CNN Model Performance on synthetic Dataset (10k Data Points)

As the dataset size increased, the higher accuracy of 1D-CNN continuously stood at 99.4 percent, showing its scalability and efficiency. The model kept the validation loss low and shows robustness for a large amount of data. Figure 4.2 below illustrates the accuracy and loss trends in the model's performance.

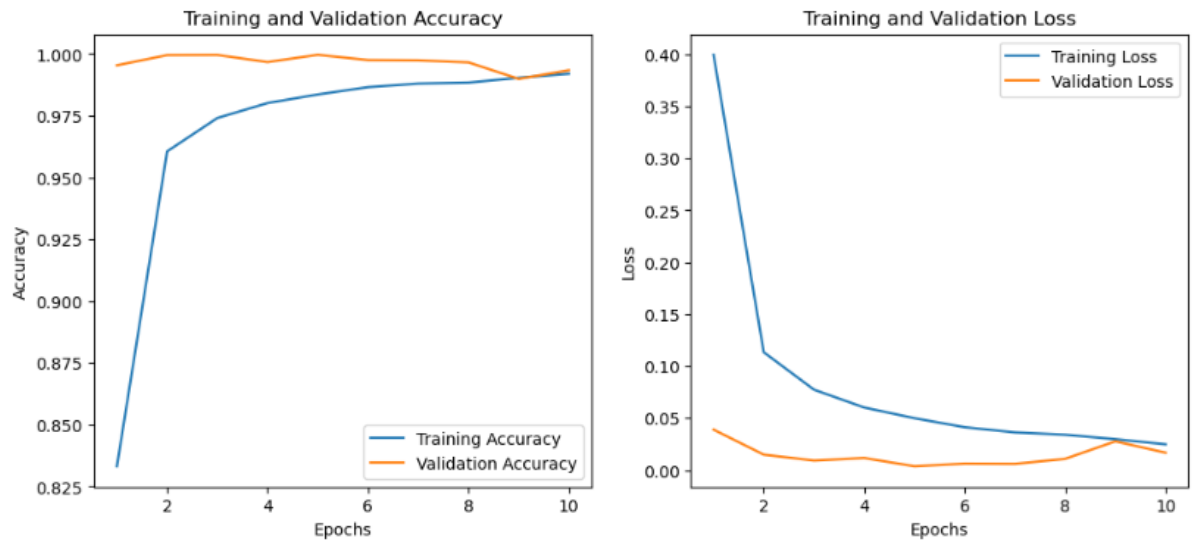


Figure 4.2: Accuracy and Trend loss of 1D CNN Model Performance on synthetic Dataset (100k Data Points)

4.7.2 1D CNN Model Performance on Hybrid Dataset

The 1D-CNN model prevailed as the top performer by reaching an accuracy of 99%. The model featured remarkable generalization capability because it maintained consistent precision and recall performance throughout every vehicle class. Figure 4.3 below illustrates the accuracy and loss trends in the model's performance.

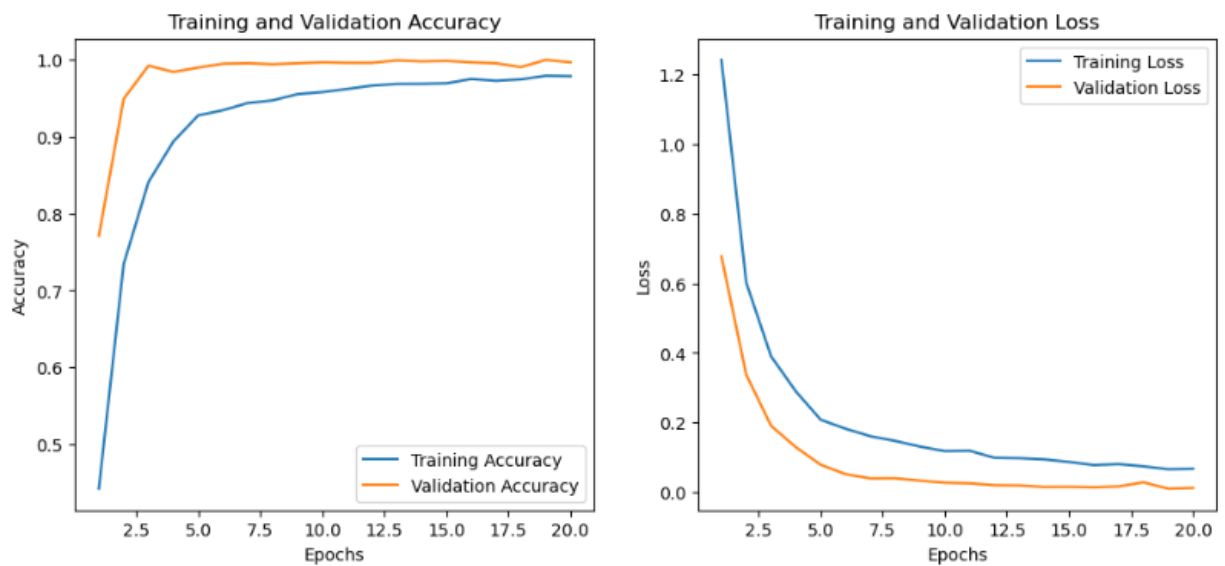


Figure 4.3: Accuracy and Trend loss of 1D CNN Model Performance on Hybrid Dataset (10k Data Points)

As the dataset size increased, it achieved a 100% accuracy rate at the large-scale hybrid dataset thus proving its ability to process various numeric data types efficiently. The system demonstrated outstanding evaluation metrics including precision and recall scores together with F1-score metrics through which it proved suitable for multiple dataset formats.

The Figure 4.4 below illustrates the accuracy and loss trends in the model's performance.

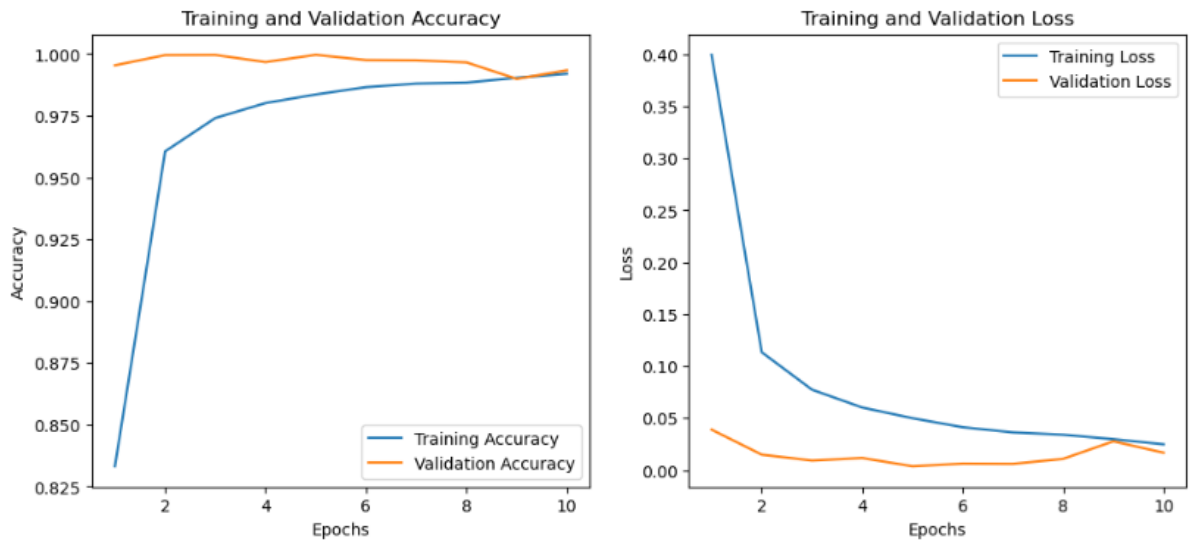


Figure 4.4: Accuracy and Trend loss of 1D CNN Model Performance on Hybrid Dataset (100k Data Points)

4.7.3 Performance of other Four Deep Learning Models on Synthetic Dataset

On one hand, the performance of these models is remarkably different because of underlying architectures and their appropriateness towards structured numerical and sequential data, 2D-CNN performed poorly with only 17.47% accuracy because it is not optimized for numeric pattern recognition and is not suitable for structured numeric data. At 17%, the RNN was not able to retain long term dependencies and hence underperformed,

due to the vanishing gradient issues leading poor classification performance. LSTM, however, was not able to be deployed on large datasets due to its high computational demand to reach 100% accuracy. The Autoencoder was finally at 17.17% accurate because it is mostly used as a feature extractor and an anomaly detector, not a good classifier. However, these results show that each model has its own strengths, but their weaknesses in dealing with structured numerical data prevent them from being directly used for classification, as models built for numerical sequence learning.

When presented with larger datasets, these models have more difficulty in performance, but they serve to highlight their strengths and weaknesses in terms of architecture as shown by the fact that 2D-CNN scored 17.06% which confirms that 2D CNN is not good at dealing with numerical sequence since it was designed to work on spatial data instead of structured numeric patterns. After 17% accuracy, RNN showed that it will significantly suffer from generalizing large datasets, which are caused by the vanishing gradient problem that limits its ability to store long term dependency at extended sequences. LSTM achieved 100% accuracy on modelling sequential data, but at a much higher computational cost compared to 1D-CNN, which then became less practical for large scale numeric data sets. Autoencoder was proven to be not very good at 16.94% accuracy in classification, as it is used primarily for feature extraction and not for supervised learning which justifies not using autoencoder directly for 16.94% high accuracy classification. By exploring various dataset sizes, these results reinforce the point that 1D-CNN remains accurate as dataset size grows, is computationally efficient, while RNN struggles with long term memories and 2D and Autoencoder become less effective as the dataset size increases. LSTM can achieve accuracy while decreasing in efficiency when compared to 1D-CNN, but it has much larger dual induced computational costs.

Figure 4.5 demonstrates four model accuracy outputs which compare their performance on synthetic dataset range of 10k to 100k.

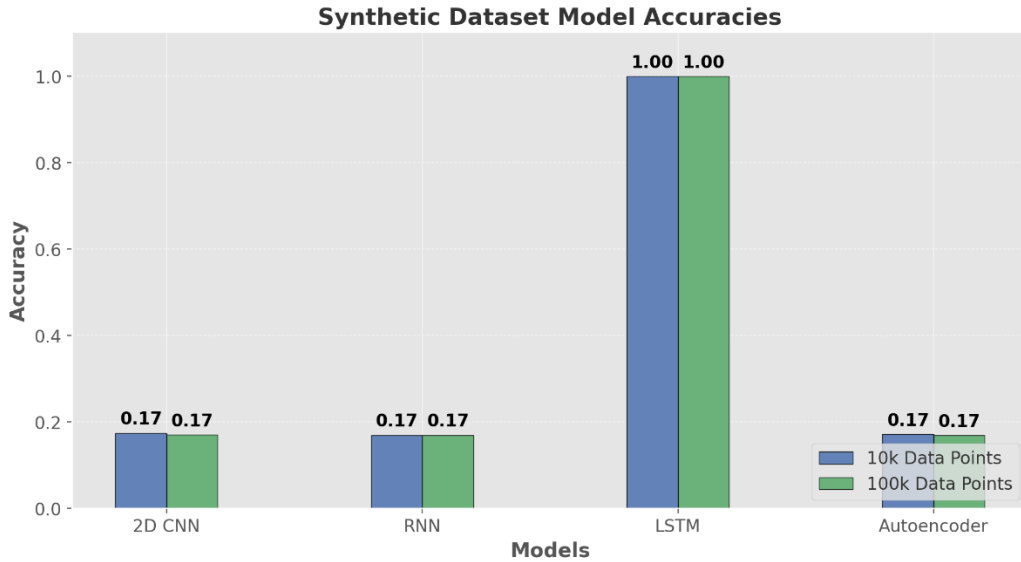


Figure 4.5: Four model accuracy outputs (Synthetic Dataset 10k and 100k)

4.7.4 Performance of other Four Deep Learning Models on Hybrid Dataset

These models differ in terms of performance due to each's architectural strengths and constraints for working with numerical or sequential data: 2D-CNN achieved 97.65% accuracy but doesn't use numerical efficiency because spatial feature extraction is its strength and structured numeric field extraction is more of a weakness. It was able to adapt to the variability of the hybrid dataset better than its synthetic dataset due to its capacity to process sequential data and to capture dependencies between future time steps. LSTM, however, was considered inaccurate at 67% as it struggled in handling hybrid datasets, which caused its vulnerability to data complexity resulting in either overfitting or lacking in generalization complexity regarding various patterns. An Autoencoder got as good as 95% accuracy, though it still was a poor performer in comparison as it is optimized for extracting features rather than direct classifying, which hindered it from differentiating between the

classes well. Overall, these results show that 2D-CNN suffers from numerical inefficiency, LSTM faces computational inefficient and over generalization issues that make them less powerful for hybrid dataset classification than networks whose purpose is structure numerical learning.

These models produce different performances because of their built architecture and the way they can tackle large scale hybrid datasets. 1D-CNN got 99.12 % performance but still lags behind 2D- CNN because they are designed to extract spatial pattern features and number series are sequential data rather than structured pattern data. 1D-CNN had 100% accuracy in RNN but it required much longer training and inference times, thus this model is computationally inefficient for large scale datasets. In that case, LSTM worked tremendously hard, only achieving 60% accuracy because it broke down on generalizing hybrid data structures, overfitting on some patterns and being too computationally complex to adapt quickly enough to changes. Autoencoder performed well at 99.58% accuracy, and was a bit inferior 1D-CNN, as Autoencoder was not directly optimized for classification and thus more suitable for feature learning rather than structured classification tasks. Results demonstrate that RNN and Autoencoder are competitive, 2D CNN is inefficient for numerical data and although LSTM fails to classify hybrid dataset complexity, 1D CNN is more efficient for large scale hybrid dataset classification. Figure 4.6 demonstrates four model accuracy outputs which compare their performance on hybrid dataset range of 10k to 100k.

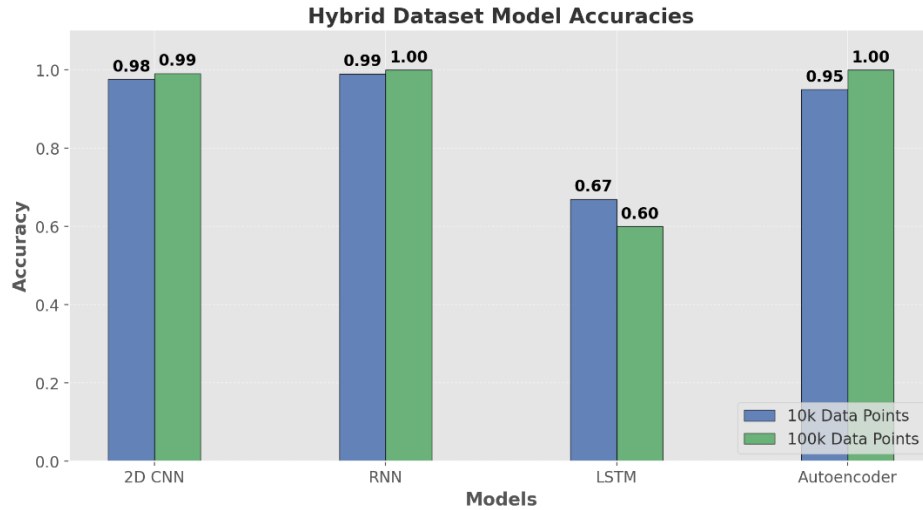


Figure 4.6: Four model accuracy outputs (Hybrid Dataset 10k and 100k)

4.8 Comparative Analysis with Other Deep Learning Models

To compare the performance of the 1D-CNN model with other widely used deep learning models, Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Two Dimensional Convolutional Neural Networks (2D-CNN), Autoencoders have been used for comparison. Through classification accuracy, computational efficiency, inference speed, each of these models is analysed.

For classifying vehicle types from sensor data, LSTM and RNN models were evaluated for suitability in sequential data processing. Although LSTM was successful in capturing temporal dependencies, it is not feasible to train as some long sequences needed longer training times and vanishing gradient issues. In contrast, the RNN model had poor performance because it could not sustain long range dependencies and hence did not achieve higher classification accuracy.

Consequently, the 2D CNN model, which is commonly used in the image-based vehicle classification task, was adapted for one dimensional sensor data. While capable of

taking patterns in space well, the 2D-CNN model had slower inference while also needing a more tedious preprocessing pipeline than the 1D CNN model. It has also proved to do poorly on real time applications due to higher computational demands and hence not feasible.

In addition, vehicle classification is tested with autoencoders so they can learn latent representations of data. Although, their primary use was in unsupervised learning, classifying problems were not ideal than supervised learning method i.e. 1D-CNN.

Comparative analysis of the results overall demonstrated that 1D-CNN model performs better in providing best results using all the criteria. In terms of accuracy, computational efficiency and speed of inferring, it provides the best trade-off between the three attributes and hence is the best choice for real time traffic monitoring.

4.9 Results Comparison

A comparison of all assessed models appears in Table 4.5 and Table 4.6 through which readers can observe accuracy together with computational efficiency and generalization performance across synthetic and hybrid datasets. The comparison reveals 1D-CNN surpasses other models in terms of accuracy combined with efficiency during the experiments.

Table 4.5: Models Performance for Synthetic Dataset (10k and 100k Data points)

Model	Accuracy (10K)	Accuracy (100K)	Computational Efficiency	Generalization Performance
1D-CNN	99.9%	99.4%	High	Excellent
2D-CNN	17.47%	17.06%	Low	Poor
RNN	17%	17%	Low	Poor

Table 4.5 continued

LSTM	100%	100%	High Computational Cost	Moderate
Autoencoder	17.17%	16.94%	Moderate	Low

Five machine learning models, namely 1D CNN, 2D CNN, RNN, LSTM and Autoencoder, are trained and tested on a synthetic dataset with two different data sizes 10,000 data points (10K) and 100,000 data points (100K), as shown in Table 4.4. Accuracy, computational efficiency and generalization performance are considered the key evaluation criteria to identify the model suitability for this data set.

Now, 1D-CNN is the best performing model with 99.9% accuracy on 10K and 99.4 on 100K. That is, the model generalizes well despite this slight drop in accuracy as the dataset size increases because the model is performing under the assumption that the dataset will contain a finite number of training examples. In addition, 1D-CNN has good computational efficiency, as compared, 1D CNN needs less resources to achieve good accuracy. The model is stated to generalize well to the synthetic dataset, and is “Excellent” labelled, indicating that the model can learn robust patterns from its synthetic dataset and apply them to unseen data.

However, 2D-CNN performs very badly as it only yields an accuracy of 17.47% on the 10K dataset and even worse (17.06%) on the 100K dataset, respectively. A low accuracy such as this implies that the model will not actually learn any meaningful pattern in the synthetic dataset and will be as good as random guessing. Additionally, 2D-CNN is known as low computational efficiency 2D CNN with poor accuracy but requires heavy computational power. This model's learning ability does not get improved with increasing

dataset size; therefore, its generalization performance is rated as "Poor", and it is therefore not suitable for this dataset.

Similarly, the Recurrent Neural Network (RNN) also shows inferior performance, with an accuracy of only 17% for both 10K and 100K datasets. Since accuracy does not change with dataset size, this implies that the model does not discover patterns well and is not benefited by more training data. Low in computational efficiency, it is calculated to be computationally expensive regardless of low accuracy. The performance in terms of generalization is also bad, confirming that RNN is not adapted to this synthetic dataset.

On the other hand, LSTM (Long Short-Term Memory network) gives a perfect accuracy of 100% on both dataset sizes (10K and 100K), what might look good at first sight. In addition, this comes with a high computational cost, resulting in the high resource intensive and potentially not practical for large scale applications. Additionally, its generalization performance is just marked as 'Moderate' as even if it reaches perfect accuracy, the model may be overfitting the training set rather than learning good image representations. However, when models see new real-world data, inferior performance can be caused due to overfitting.

The final Autoencoder results are the worst (17.17% on the 10K dataset and 16.94% on the 100K dataset), as it is remarkably close to random guessing as 2D CNN and RNN. Thus, the ability of the autoencoder to learn effective feature representations from the synthetic dataset is extremely limited. Although the model is not very computationally efficient, the low accuracy does not justify the usage of the model. The performance at generalization is rated "Low" which means that the learning ability increases with the dataset size.

Overall, 1D-CNN achieves the best results for the synthetic data set as it has high accuracy, extremely high computational efficiency, and good generalization. Although LSTM is perfect, it is the resource intense and will overfit. (2D-CNN, RNN, Autoencoder) also all don't work well in accuracy side as well as its generalization, so they can't do the job for this dataset.

Table 4.6: Models Performance for Hybrid Dataset (10k and 100k Data points)

Model	Accuracy (10K)	Accuracy (100K)	Computational Efficiency	Generalization Performance
1D-CNN	99%	100%	High	Excellent
2D-CNN	97.65%	99.12%	Low	Moderate
RNN	99%	100%	High Computation Time	Moderate
LSTM	67%	60%	High Computational Cost	Poor
Autoencoder	95%	99.58%	Moderate	Good

In Table 4.5, the same five models are evaluated for performance – 1D-CNN, 2D-CNN, RNN, LSTM, and Autoencoder, as they would have performed while trained and tested on hybrid dataset instead of those used for synthetic dataset. Like that in Table 5, the dataset is tested in two sizes (10k and 100k) and evaluated by accuracy, computational efficiency and generalization performance.

As is the case again, 1D-CNN turns out to be the most powerful model with an accuracy of 99% on the 10K dataset and 100% on the 100K dataset. Unlike in the synthetic dataset where accuracy deteriorates with larger dataset size, here the accuracy increases with more data, indicating that 1D-CNN is advantageous to the hybrid dataset and improves with larger dataset size. This model has high computational efficiency, and its generalization

performance is still good for both the datasets, hence suggesting that it is the most dependable model.

Now the 2D-CNN performs much better than earlier Table 5. On 10K and 100K respectively, it reaches 97.65% and 99.12% accuracy, with a significant improvement compared to the dataset on which it was trained previously. Nevertheless, with more accuracy 2D-CNN is computationally inefficient and requires a lot of processing power. Classification performed on its generalization performance is rated as “Moderate”, meaning that it benefits from more data, however, it is not as robust as 1D-CNN.

The intense improvement in the RNN, over its synthetic dataset results, by performing 99% on 10K and 100% on 100K. This implies that RNN is receptive to hybrid datasets, maybe because the data distribution is well suited for a structure. Still, though, its computational efficiency is far from ideal, as it has an exceptionally long computation time to produce these results. Although its overall performance is marked as 'Moderate', it remains accurate, but it may be too computationally expensive.

Where LSTM performs poorly in comparison to its synthetic dataset results, it is only able to achieve 67% and a large drop in 60% accuracy on the hybrid dataset. This suggests LSTM is not generalizing well on the hybrid dataset nor overfitting the smaller dataset when failing on the larger one. It is also expensive computationally and thus neither appealing. The generalization performance of this model is classified as “Poor”, further supporting the fact that this dataset is not well suited for such a model.

In hybrid dataset, autoencoder achieves 95% accuracy on 10K and 99.58% accuracy on 100K, far better than the synthetic dataset. It implies that hybrid data is beneficial to Autoencoder and that it performs well with a bigger dataset. It has moderate computational efficiency, and the performance and resource usage are balanced. The result is good on the

generalization performance, meaning the model manages to adjust well as it grows with the size of data.

The results conclude that 1D-CNN is the best model for the composite dataset in terms of high accuracy, computational efficiency and great generalization ability. RNN, Autoencoder, 2D-CNN all have good generality, RNN is computationally expensive, Autoencoder is only good, and 2D-CNN is still able to make improvements, however it is not efficient. LSTM performs much worse and is the worst performing model for this dataset. 1D-CNN turns out to be the most commonly high performing model on both synthetic and hybrid datasets which prefers practical applications. RNN, 2D-CNN and Autoencoder work well on hybrid data while failing on synthetic data. LSTM works well in one situation, but not in another, and hence does not work well in real world situations.

4.10 Discussion

The result from the experimental findings suggests that the proposed 1D-CNN model can be useful in automatically classifying and counting vehicles. We show strong results which demonstrate the model's ability to classify vastly different vehicle types with high accuracy, precision and recall. Although some challenges and limitations were identified in this research, they had to be overcome in future research.

Previous studies (Kapoor et al., 2019) have already demonstrated the effectiveness of ID-CNN in object detection tasks. This chapter therefore focuses on adapting ID-CNN to the specific context of vehicle classification and counting rather than re-establishing its superiority.

The environmental noise problem is one of the main challenges with the sensor that must be dealt with, as it is likely to influence the model's performance. First, noise in the

dataset can be introduced by external factors like road vibrations, weather conditions etc. which can oversimplify the classification. A remedy to this problem is to improve techniques of noise filtering and signal preprocessing.

The generalization across all different road conditions also limits the model. In this study, training data is collected from a particular location which may not accurately depict the environment for other region's traffic conditions. To increase robustness and adaptability it would be beneficial to expand the dataset to include data from varied locations (different road surfaces, vehicle densities, driving behaviours) of the dataset.

Especially when deploying the 1D-CNN model in intelligent transportation systems in the real world, scalability should also be considered as a critical factor. While the model does well on a small set of data, it cannot easily be scaled up to run across wide scale deployment on multiple road networks, which requires high efficiency data processing pipelines and integration in the cloud. The future work will concentrate at optimizing computational resource allocation and integrating edge computing solutions to enable real time vehicle classifying and counting on a larger scale.

4.11 Summary

The research demonstrated an extensive examination of the outcomes achieved by implementing the 1D-CNN model in automatic vehicle classification and counting procedures. The results proved that the model achieved better accuracy rates together with higher operational efficiency and flexible deployment in comparison to other deep learning methods that included LSTM, RNN, 2D-CNN and Autoencoders.

The chosen performance evaluation metrics of precision, recall and F1-score show that 1D-CNN delivers the best solution for vehicle category discrimination along with quick

inference times needed for real-time traffic surveillance. The model received enhancements through added vehicle counting capabilities to provide better services in intelligent transportation systems where traffic density evaluation leads to improved congestion control.

The model demonstrates robust performance although existing problems involve sensor noise and difficulties with varied road conditions together with challenges for scalability. Future job efforts should concentrate on developing better noise reduction techniques and enlarging existing datasets for boosting model generalization before scaling up the technology for deployment across cloud-based and edge systems platforms. 1D-CNN model creates substantial progress in vehicle identification applications and vehicle counting operations which enhances traffic system monitoring capabilities. The developed knowledge enables effective traffic control and improved road safety and enables the integration of intelligent transport systems for the upcoming period.

CHAPTER 5

EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

5.1 Overview

In this chapter, the developed framework for vehicle classification and counting is introduced with detailed study of the usage on synthetic datasets. The intent of this framework is to increase accuracy and efficiency using 1D Convolutional Neural Network (1DCNN). To provide a diverse and representative learning environment for his generalization, the framework is intended to leverage two different synthetic datasets.

The chapter is organized in such a way as to offer a total understanding of the proposed methodology. It starts by discussing into why the use of synthetic datasets and how they provide better scalability at reduced costs compared to real world data and more controlled variability. Then comes the exploration of the 1D-CNN model and if it is suitable to apply for the task at hand, in the extraction of features and the processing of sequential data.

In addition, the framework's core components, including data acquisition and pre-processing, model selection and training, and the two key application scenarios of vehicle classification and vehicle counting are discussed in this section. In detail, each of the aspects that compose the framework are explored to show how the framework guarantees robust and accurate performance under many traffic conditions. Additionally, for ascertaining the performance of the model, the mentioned evaluation metrics such as classification accuracy, mean average precision (mAP), and inference speed are listed to allow a smooth comparison with the performance of the model.

In general, this chapter seeks to provide a solid basis for grasping the proposed method and its originality, technology, and potential contributions to the top of intelligent transportation systems.

5.2 Framework Overview

The proposed framework brings together several interacted components for effective classification and counting of vehicles while maintaining high accuracy and flexibility. The workflow comprises every stage from data acquisition to the end of performance evaluation of work, which is essential for work efficient performance. Overall, the framework has five key stages which are seamlessly combined to make it a robust, scalable system as shown in Figure 5.1.

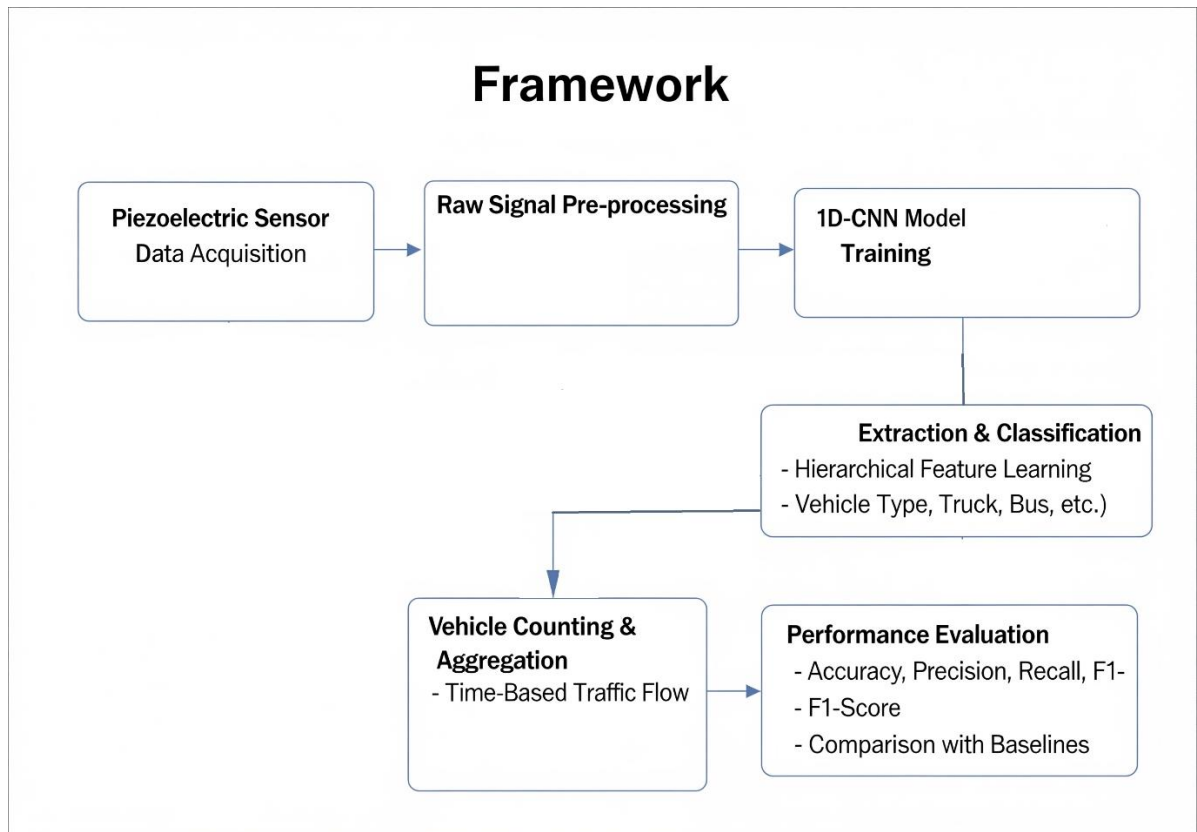


Figure 5.1: Framework Overview

Data preparation and acquisition are done first in which typical datasets are created, pre-processed, and structured for training. The datasets are diverse in their vehicle types as well as environmental conditions and controlled variations which help in making the model better generalized. Upon the arrival of the data, preprocessing such as normalization, augmentation, and segmentation is applied to refine the input data such that the input data is consistent in different scenarios.

After preparation of data, model training phase begins and 1DCNN architecture is used to learn and extract key features from the dataset. Its training process is built with hyperparameter tuning, dropout regularization and batch normalization to make the performance of the model as optimal as possible. Using these strategies helps stop overfitting the model and enhances the model's capacity to generalize unseen data. Using the dataset, we have training, validation, and testing sets are created for use during training, and can evaluate in a structured way.

The vehicle classification stage, which categorizes vehicles to predefined classes (car, Van, truck, bus, motorcycle) is executed after training. The vehicle type specific area is based on the learned feature from convolutional layers to distinguish the type of vehicle via its characteristic. Standard performance metrics such as precision, recall and F1-score are utilized to evaluate the classification, which should achieve a high accuracy in separating between different vehicle categories.

After classification, the vehicle counting module is developed which counts vehicles in dynamic traffic situations. By detecting and segmenting vehicles from sequential data, the 1DCNN model is specifically able to count vehicles even in dense or overlapping traffic. Especially for real time traffic monitoring and intelligent transportation systems, this component plays a significant role in delivering the required data for analysis and deciding.

The final phase is performance evaluation which takes the effectiveness of the framework in terms of accuracy, mean average precision (mAP) and inference speed. However, these evaluation criteria will also give insights into the strengths of the system and the difficulties that can be adjusted and scaled efficiently. This approach to structure provides an efficient and scalable system to vehicular classification and counting through integration of each component within the framework. The modular Ness of the framework facilitates customizability and upgrades that make the framework a candidate for utilization in the real-world intelligent transportation applications.

5.3 Data Acquisition

To perform a rigorous evaluation of the proposed framework we use two exclusively different hybrid synthetic datasets that are developed using Python based techniques. The procedural generation approach is used to generate both datasets through a combination of multiple synthetic vehicle data sources together with various traffic conditions to create the most realistic scenarios as close as possible to the real world.

Specifically, the first dataset referred to as Dataset A is set aside for testing. And it is a hybrid synthetic dataset derived from using some Python scripts generating controlled variations of lighting, weather, occlusions and road environment. It guarantees a structured and reproducible testing environment for the framework to be precisely evaluated in vehicle classification and counting. Additionally, Dataset A is created to be as close to Dataset B in its root element characteristics but is used under controlled environment only to achieve consistency in benchmarking. The testing phase eliminates external biases with which we can use a systematically generated synthetic dataset to assess and get reliable performance metrics with which to assess deep learning models.

Dataset B, which is the second dataset, is also a hybrid synthetic dataset but it is used for validation and has been created using Python based methodology, it is intended to replicate real world traffic conditions more dynamically. Dataset B is designed with various complexities like fluctuating vehicle densities, varying angles of observation, motion blur, occlusions and overlapping objects to mimic various and variable traffic environments. This dataset is vital to appraise the model's generalization or, more specifically, to verify how accurately the model can generalize outside of the constrained testing conditions offered by Dataset A. Thorough validation on Dataset B demonstrates the robustness, adaptability and real-world applicability of the framework, for which it can sufficiently deal with the diversity in vehicle classification and counting issues, under the complexity.

5.3.1 Data Preprocessing and Preparation

The study ensures a highly controlled benchmarking environment, since it relies upon these synthetic datasets alone for both testing and validation, thus ensuring much less inconsistency and bias and that is often caused by the variations of real-world data. In addition, large, rigorous preprocessing steps, i.e. data augmentation, normalization, and creating annotation validation of the datasets improve their quality and provide the optimal input to deep learning models. This approach provides a solid basis for evaluating the performance of the model in advance before it could be embedded in a deployment with intelligent transportation systems.

5.4 Model Training

For feature extraction and pattern recognition for sequential data, the 1D Convolutional Neural Network (1DCNN) has been chosen as the selected deep learning model for this framework. With such a preference for time series and structured sequence

information, the 1DCNN model is the best choice, especially for vehicle classification and counting applications. Compared to the conventional CNN architectures that work with two spatial dimensions, the 1DCNN model works with one dimensional feature vector, which enables it to learn the most important patterns while keeping the complexity low.

The model is then trained rigorously with a high number of hyperparameter tuning to fine tune performance. Then, the learning rate, batch size, number of layers of convolution, kernel size is tuned to obtain at least the highest possible accuracy. Further, data augmentation techniques which include signal transformations and noise injection are provided to improve robustness of the model against variations in the input data.

Regularization techniques, on the other hand such as dropout and batch normalization are subsequently incorporated into the training pipeline to help avoid overfitting and improve generalization. Batch normalization makes learning easier in the sense that it normalizes internal activations between the hidden layers to address overfitting in the model. Dropout randomly drops out a portion of neurons within the layers and it prevents the model from depending too much on some features. As shown in Table 5.1.

Table 5.1: Architecture of 1D CNN Model presented in a Structured Table

Layer Type	Output Shape	Parameters	Details
Input Layer	(None, num_features, 1)	0	Input shape derived from dataset
Conv1D	(None, num_features-1, 32)	96	filters=32, kernel_size=2, activation='relu'
MaxPooling1D	(None, num_features-1, 32)	0	pool_size=1 (reduced pooling)
Dropout	(None, num_features-1, 32)	0	rate=0.2
Conv1D	(None, num_features-2, 64)	4160	filters=64, kernel_size=2, activation='relu'
MaxPooling1D	(None, num_features-2, 64)	0	pool_size=1 (reduced pooling)

Table 5.2 continued

Dropout	(None, num_features-2, 64)	0	rate=0.2
Flatten	(None, flattened_features)	0	Converts to a 1D vector
Dense	(None, 64)	flattened_features * 64	activation='relu'
Dropout	(None, 64)	0	rate=0.5
Dense (Output)	(None, num_classes)	65 * num_classes	activation='softmax'

A structured dataset is split into training, validation and testing set in which the model is trained. Then the validation dataset is used to continuously monitor model performance and fine tune hyper parameters while the final evaluation is made on the test set to guard against bias. To assess the model's performance, performance metrics, including accuracy, precision, recall, and mean average precision (Map), are used and ideally yield the expected performance levels prior to deployment. Table 5.2 includes all the values of hypermeter.

Table 5.3: Hyperparameters for the Model

Hyperparameter	Value
Dataset Split	Train: 80%, Test: 20%
Validation Split	20% of training data
Random Seed	42 (for reproducibility)
Stratified Sampling	Yes
Feature Scaling	StandardScaler (Z-score)
Label Encoding	Yes (LabelEncoder)
Loss Function	Categorical Crossentropy
Optimizer	Adam
Metrics	Accuracy, precision, recall, and mean average precision (Map),
Batch Size	32
Epochs	20

A division of the dataset into training and testing sets was done to keep the model evaluation to be as balanced and unbiased as possible, using 80% data for training and 20% for testing. 20% of the data was further set aside within the training set to conduct validation (fine tune hyperparameters to prevent overfitting). We used the stratified sampling technique to keep both data class distribution and dataset subsets representative of the whole.

Preprocessing is necessary in many methods, an important one being to standardize numeric values in feature scaling. To do so in this research, we used Standard Scaler to implement Z-scale normalization for all features (i.e., mean of 0 and std of 1). It allows convergence speed of the model to be improved and avoids features dominating the others because of difference in scale. Label Encoder was used to encode the target variable (categorical labels) as numerical values, which can be used by a model during training.

In this study, categorical CrossEntropy is chosen as the loss function since it is the best candidate for the loss function for multi-class classification problems. Since it had an adaptive learning rate and fast gradient computation, the Adam optimizer was chosen. The model was evaluated on its performance using accuracy, which is a commonly used metric for classification tasks. The model was trained with a batch size of 32 to balance memory efficiency and computational cost. The training process took 20 epochs to allow the model to figure out meaningful patterns and prevent training which can cause overfitted.

Hyperparameters play a significant role when developing a good performing machine learning model and it is required to be carefully selected. The choices made in the proposed study were based on the best practices in deep learning and evaluated on the experimental set-up. Proper dataset splitting, preprocessing and optimization techniques are used to ensure that the trained model has high accuracy, and the generalization is good for the unseen data.

5.5 Scenario 1: Vehicle Classification

The first scenario is vehicle classification in which the 1DCNN model is used to classify vehicles to predefined ones, i.e., ‘Cars’, ‘Vans’, ‘Trucks’, ‘Buses’, and ‘Motorcycles’. This realistic scenario is important from the point of view of vehicle identification, which is vital over Intelligent Transportation Systems, Traffic Management

as well as for Automated Surveillance. Utilizing the rich feature extraction skills of 1DCNN, the model can learn the distinctive properties between various kinds of vehicles and consequently boost the classification precision. In this scenario, a synthetic dataset used specifically to model a variety of real-world cases is used such that the model can effectively generalize to unseen data.

This involves several key steps of executing the workflow for vehicle classification namely, input data processing, feature extraction using 1DCNN model, classification of the vehicle into various categories and finally performance evaluation calculated by accuracy, precision, recall and F1 score. Figure 5.2 below shows the workflow of the vehicle classification.

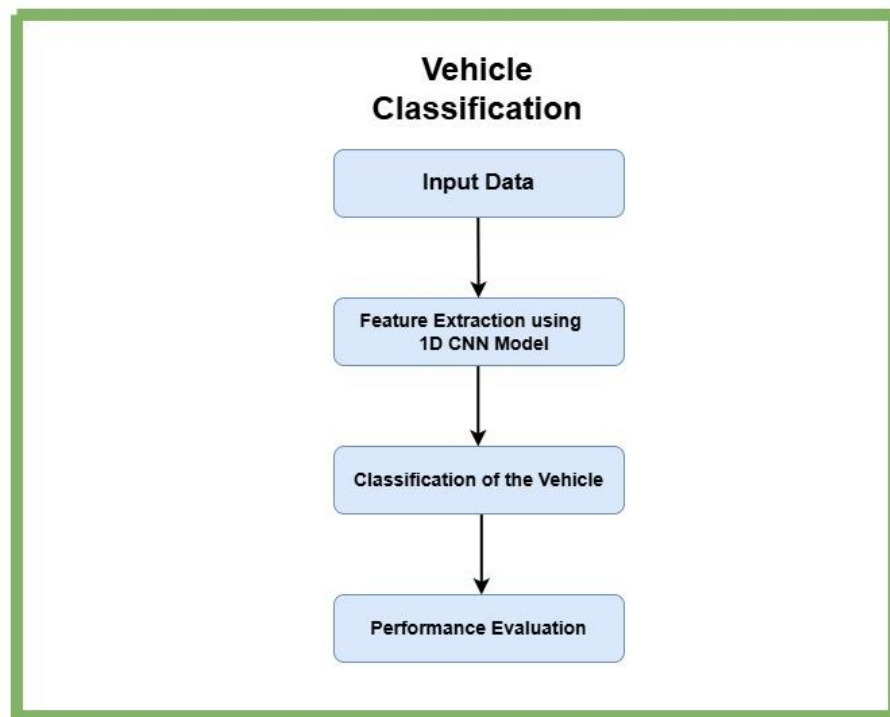


Figure 5.2: Workflow of Vehicles Classification

The first and most basic step of vehicle classification is the input data processing, which comes directly from sensor data, and needs to be pre-processed to make up for

effective learning by the model. Preprocessing of the data collected from the synthetic datasets was done by using cleaning, normalization, augmentation, and segmentation. Cleaning guarantees that the data is clean, free from noise, missing data as well as inconsistencies that can otherwise inject bias in the model. Normalization helps to bring all features under a uniform scale to avoid the promising features overpowering minor yet key details. The model's robustness can be improved by using data augmentation techniques such as flipping, rotation or even noise injection to increase variability of the training data. Finally, the dataset is segmented into parts that can be easily classified: each instance is clearly defined.

The 1DCNN model extracts the features/documents from input data in this step, to identify the keys patterns and structures that separate the vehicle types. In contrast with traditional CNNs that process spatial data in two dimensions, the presented architecture of 1DCNN is broadened to process sequential data for vehicle classification tasks in which the numerical patterns are structured. The 1DCNN architecture includes several convolutional layers that are also used to convolve the data and extract such features as shape, size and specific properties of different vehicles. Once the features are extracted, these features are then passed on through activation functions such as ReLU to add some non-linearity in the model and capture the complex pattern. To reduce dimensionality and complexity of computation, pooling layers are also incorporated that will preserve notable features.

After processing the feature extraction, the model classifies vehicles into a particular class from the Our Egg category. Fully connected dense layers receive the extracted features and learn top level representations that the model expects to assign their own probabilities to different classes. On the output layer, which is the Softmax activation function ensures that the model 'predicts one' so that any vehicle instance can be determined the category

with the most probable. The 1DCNN model is trained and optimized through extensive training such that it is capable of minimizing classification errors and maximizing accuracy to guarantee reliable categorization of vehicles even in the presence of varying environmental conditions and diverse traffic scenarios.

In the end, the model evaluated standard metrics like accuracy, precision, recall and F1-score for the ultimate step of the classification workflow. Accuracy is the ration of correctly classified data over total data, i.e., the measure of complete correctness in terms of the model. Precision determines the ratio of correctly predicted vehicles in each class versus all the predicted vehicles in this class to avoid anything being confused with or labelled as a vehicle. Its sensitivity to detecting all relevant instances is reflected in Recall's evaluation of the model's ability to correctly identify vehicles within a certain category. A balanced measure, which is F1 score which is the harmonic mean of precision and recall, is the F1 score that we use to ensure precision corresponds to both false positives and false negatives. Each of these metrics represents a general measure of the classification model effectiveness as a whole and they can be used as benchmarks for further refinement and optimization.

5.6 Scenario 2: Vehicle Counting

The next scenario is vehicle counting, with a 1DCNN model that is trained to detect and count vehicles in real time from input data. Traffic monitoring, planning of road infrastructure and intelligent transportation systems require this process. The ability to count vehicles in dynamic traffic conditions gives traffic authorities an indicator of congestion levels, allows them to optimize traffic signals, and improve overall roadway efficiency. This is quite different from the taking count methods using manual counting which are labor intensive and prone to errors and the 1DCNN model can be fast and automatic to count vehicles with more accuracy and reliability.

The three steps of the vehicle counting workflow are: Instance segmentation for counting using 1DCNN to detect the object, Performance evaluation by Mean Average Precision (mAP) and Intersection over Union (IoU). As shown in Figure 5.2 below.

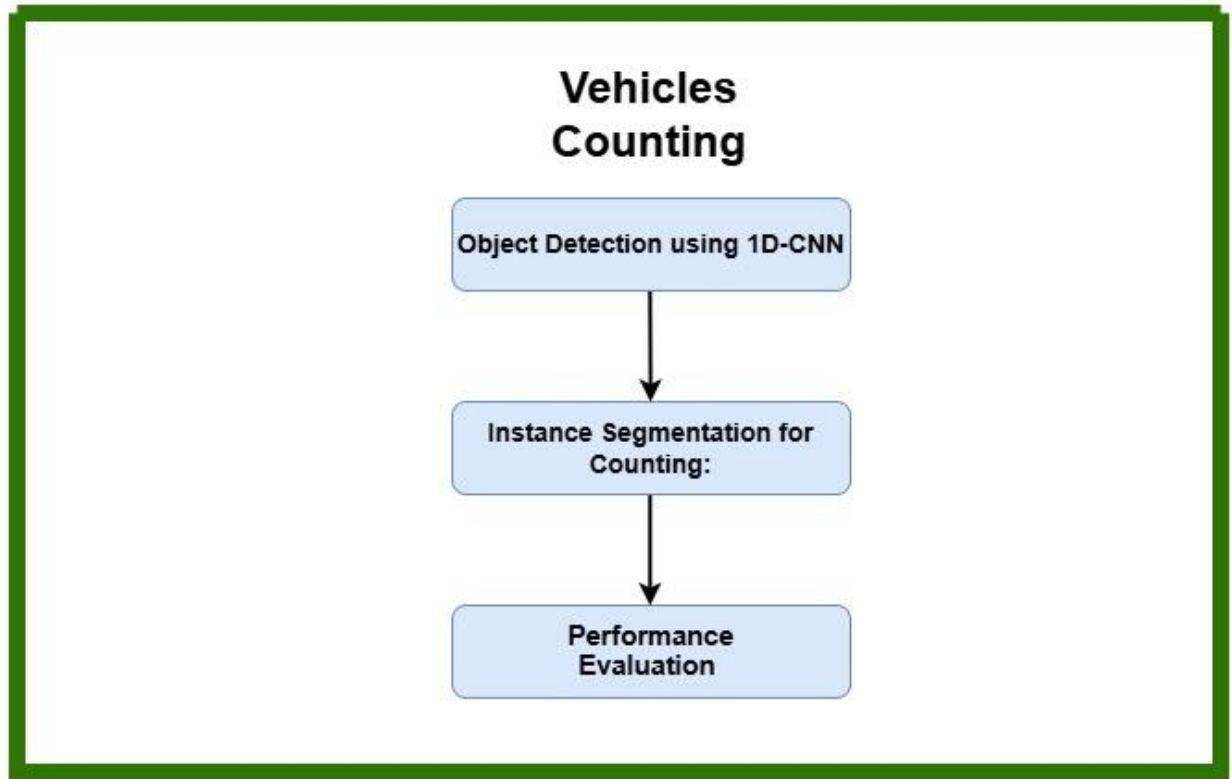


Figure 5.3: Workflow of Vehicles Counting

Object detection, being the first step of the vehicle counting process, is detecting or localizing the vehicle in the input data by the 1DCNN model. Taking a sequential dataset of sensor reading and analyzing how various key features like motion pattern, energy signals and spatial distribution can help detect individual vehicles, the model processes the sequential data. Thus, the 1DCNN's convolutional layers extract meaningful patterns and differentiate the vehicles from the background noise and irrelevant elements. The 1DCNN architecture differs from the traditional object detection methods using complicated region proposal networks, where filtering techniques are used to make the 1DCNN architecture fast

and accurate. This step is imperative in measuring accurate vehicle counting because it will eliminate any vehicle instances that might present problems in further consideration.

Once vehicles are detected, then the second phase is instance segmentation to obtain a segment for each detected vehicle with unique identification and counting. By instance segmentation, the model prevents counting vehicles that are overlapping each other. In the segmentation process, the detected object regions are segmented into individual vehicle instances such that each vehicle is appropriately counted. To reduce the errors of the occlusions or partially visible vehicles, advanced feature matching techniques are used to track the vehicle across each successive frame. Integrating it with real time tracking guarantees that the vehicle counting process is indeed exactly accurate even in dense traffic conditions.

The performance evaluation metrics for counting vehicles, Mean Average Precision (mAP) and Confusion Matrix are used to assess the performance of the vehicle counting model. Object detection using a metric means average precision (mAP) is a widely used metric among the rest to evaluate the ability of models to detect and count vehicles in different traffic scenarios. The multiple recall levels give a balanced evaluation of detection accuracy. It calculates the average precision across various recall levels. A higher mAP score shows that the model has well distinguished and counted the vehicle instances without any left-out cases. Here is another important metric that assesses accuracy of model's prediction, and that is Confusion Matrix. A confusion matrix gives a breakdown of the models predict accuracy for each vehicle category and identifies any misclassifications in that category. It makes visible in the context of vehicle counting how blanketing the model overcounts or undercounts some of the vehicle types because of occlusion, overlapping vehicles, and environmental variances in the synthetic dataset. These performance metrics are then

integrated by the 1DCNN based vehicle counting model to ensure the reliability of the model in real world applications. Since this approach is automated and data driven, the traffic monitoring is much more capable now than before; authorities can make decisions on road management, or congestion mitigation, or on infrastructure planning.

5.7 Performance Evaluation

The effectiveness of the 1DCNN-based vehicle classification and counting framework is measured through multiple performance evaluation metrics. These metrics assess the accuracy of classification, the precision of object detection, and the computational efficiency of the model, ensuring that it meets real-world deployment standards.

5.7.1 Classification Metrics: Accuracy, Precision, Recall, F1-score

Classification metrics are essential for evaluating how well the model differentiates between vehicle categories. Accuracy is a fundamental metric that calculates the percentage of correctly classified vehicles over the total number of predictions, providing an overall measure of model correctness. However, accuracy alone does not provide a complete understanding of model performance, especially in cases where data imbalance exists. To address this, precision, recall, and F1-score are also considered. Precision measures the proportion of correctly predicted positive instances (e.g., correctly identified cars) over the total predicted positive instances, ensuring that false positives are minimized. Recall determines the proportion of actual positive instances correctly identified by the model, which is crucial for detecting all relevant vehicles in each dataset. The F1-score, a harmonic mean of precision and recall, provides a balanced measure of performance, especially when managing imbalanced datasets where one vehicle category may be more frequent than others.

5.7.2 Detection and Counting Metrics: Mean Average Precision (mAP), Confusion Matrix, and Counting Accuracy

To evaluate the accuracy of vehicle detection and counting, object detection-specific metrics such as Mean Average Precision (mAP) and Confusion matrix are employed. Mean Average Precision (mAP) is used to assess the precision of object detection by calculating the average precision across multiple recall levels. A higher mAP score indicates that the model is effectively distinguishing between different vehicles, accurately detecting and localizing them within the input data. A confusion matrix gives a breakdown of the models predict accuracy for each vehicle category and identifies any misclassifications in that category. It makes visible in the context of vehicle counting how blanketing the model over counts or undercounts some of the vehicle types because of occlusion, overlapping vehicles, and environmental variances in the synthetic dataset. Additionally, counting accuracy is evaluated to measure the model's ability to correctly enumerate the number of vehicles present in a scene, ensuring minimal undercounting or over counting errors.

5.7.3 Computational Efficiency: Inference Speed and Resource Utilization

The deployment feasibility of the model depends on its computational efficiency, which is assessed through inference speed and resource utilization. Inference speed measures how quickly the trained model processes new input data and generates predictions, making it a crucial factor for real-time applications. The ability to classify and count vehicles within milliseconds ensures that the framework can be integrated into intelligent transportation systems for real-time monitoring. Resource utilization evaluates the amount of computational power required by the model, including GPU memory consumption and CPU usage. Efficient resource utilization ensures that the model can operate on edge devices and

embedded systems, making it scalable for a variety of real-world applications, from automated toll booths to smart traffic management solutions.

By evaluating these performance metrics, the effectiveness and reliability of the 1DCNN model are thoroughly assessed, ensuring that it meets the demands of real-time vehicle classification and counting applications. This performance analysis helps fine-tune the model, optimize detection capabilities, and improve computational efficiency for seamless deployment in intelligent transportation networks.

5.8 Implementation Details

5.8.1 Framework Development Tools

Many essential tools and libraries are required to implement the vehicle classification and counting framework and it heavily depends on deep learning and image processing and model evaluation libraries. They give the required computational power and optimization strategies to end up in proficient preparation, exact predictions and real time handling.

5.8.2 TensorFlow and PyTorch for Deep Learning

The 1DCNN is developed and trained using TensorFlow and PyTorch as the two deep learning frameworks. The TensorFlow ecosystem for deep learning provides a flexible range to app, static as well as dynamic computation graph to optimize training performance. It also includes TensorBoard, an in-built visualization tool which consists of metrics about deployed results in terms of accuracy, loss function and weight, and so on. On the other hand, it is preferred to use Pytorch as it has dynamic computational graphs which enable real time debugging and makes possible for various neural network architectures to be experimented seamlessly. By using both frameworks, the 1DCNN model benefits from the

robust gradient optimization, automatic differentiation, and hardware acceleration thus making it applicable for vehicle classification and counting tasks in large scale.

5.8.3 Scikit-learn for Evaluation Metrics

A performance of the trained model is evaluated using scikit-learn library. It offers an extensive collection of machine learning tools for statistical analysis, class reports, accuracy metrics, etc. The accuracy, precision, recall, and F1-score are calculated using the library to evaluate the effectiveness of the classification model. It also helps generate confusion matrix that can be used to analyze misclassification patterns and provide hints about the regions for model improvement. By using scikit learn cross validations, it has ensured that the model's performance is seen on multiple data splits to prevent overfitting, and to generalize to new vehicle instances.

5.8.4 Hardware: GPU-Accelerated Training for Efficiency

However, vehicle classification and counting poses stringent requirements on computational resources, because there are many deep learning models for these tasks which need a lot of computing power. To deal with this, the implementation takes advantage of GPU acceleration through NVIDIA CUDA enabled GPUs. However, that 1DCNN model can be trained much faster on GPUs for backpropagation, and for update of weights, with GPU training virtually eliminating the time to do backpropagation and weight updates. Through GPUs' parallel processing capability, it is ensured that multiple convolutional operations can parallelize, and the matrix multiplications can be an aggregate as well as decrease latency in inference tasks. Moreover, the use of hardware accelerators (such as Tensor Processing Units – TPUs) can further improve the performance in a way to enable

deploying the real time vehicle detection and classification in intelligent transportation systems.

The framework enables high efficiency, accuracy and scalability by integrating these tools and hardware optimizations. With GPU acceleration, TensorFlow, PyTorch and Scikit-learn help the development of a robust, reliable vehicle classification and counting system. Thus, the model remains plausible for real world applications where vehicle recognition is insurgent, rapid and precise for traffic monitoring, urban planning and smart transportation schemes.

5.9 Results

Diverse metrics for 1D Convolutional Neural Network (1DCNN)- based framework for vehicle classification and counting like classification accuracy, precision, recall, F1 score, Mean Average Precision (mAP), Confusion Matrix, counting accuracy, inference speed and computational efficiency are evaluated. Results indicate high accuracy and reliability of the framework in classifying and counting vehicles.

5.9.1 Framework Performance Overview

This proposed framework reaches an overall test accuracy of 0.99, which demonstrates its ability to correctly classify and count the vehicles within the various categories. The high accuracy level demonstrates that the model has learned the complex patterns needed to effectively differentiate between different vehicle types. In real world context the model generalizes well as it has a 0.99 unseen data accuracy and proves to be robust and adaptable to unseen data. With inference speed being 17.69 seconds, the model is quick enough to serve its purpose in real time intelligent transportation applications.

Furthermore, the framework utilizes 84.4% of the memory amount, which is an ideal trade-off between computational resource usage and model performance.

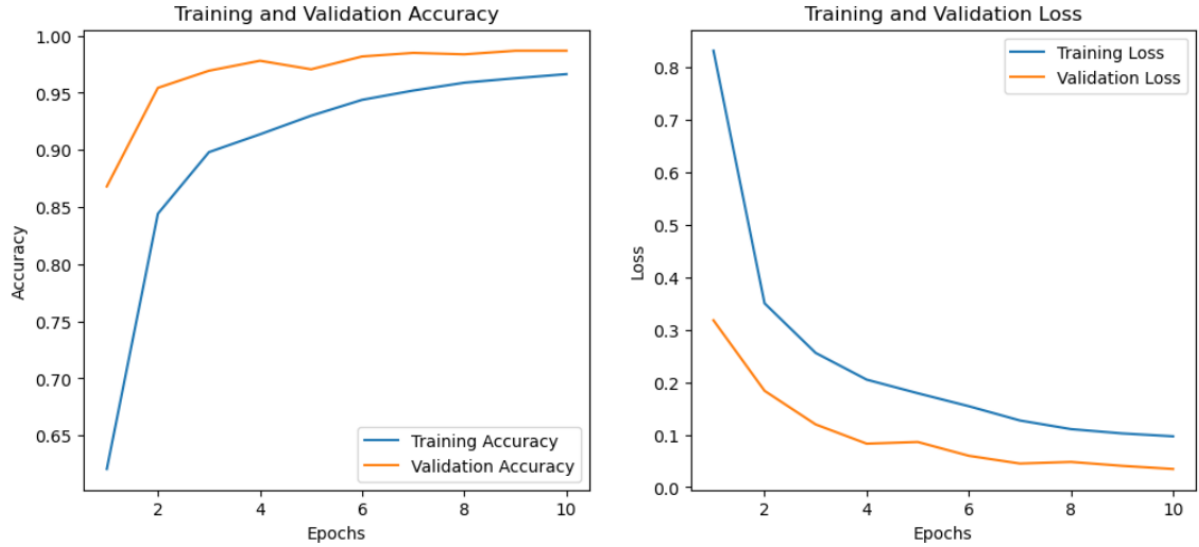


Figure 5.4: Framework's Model Accuracy and loss

In Figure 5.4, the graphs presented show the training and validation performance of a machine learning model in the form of 10 epochs. These visualizations give indication of how well the model learns from the training data and what level of generalization is present for the test data that is not seen by the model during training. The left figure shows training and validation accuracy over time, the right figure training and validation loss.

The accuracy graph has its x-axis labelled as number of epochs and y-axis labelled with achieved accuracy by the model. The blue line, the training accuracy line starts low and slowly increases as the model learns the patterns in the dataset. After the final epoch, accuracy reaches above 95%, implying that the training data has been well fitted by the model. As shown in the orange line, the validation accuracy has a higher initial value and rises quickly up to about 99%. This indicates that the model has high generalization performance, since it can accurately classify on the validation set. However, the small

difference between the accuracy of training and validation accuracy indicates a low amount of overfit.

Additional insights regarding the model's learning 'feel' can be obtained through the right plot (training, validation loss). The number of epochs (the x axis again) is considered, and the loss (i.e., the x-axis again) is represented again, which indicates the measure of the prediction error. In the blue line, the training loss initial value is high but decreases steadily as the model adjusts its parameters. However, the fact that it is decreasing this steadily means that the model is learning well. In the same manner, the orange line plot shows that the validation loss decreases as well and is lower than the training loss consistently. Training and validation loss are in a small gap showing the model is well regularized and does not overfit. But validation loss slightly fluctuated in the late epoch. It means that the validation data is not perfectly fixed.

Overall, these graphs show that the model has learned the training data well while having very strong generalization. While there is a tiny signal of overfitting, it is not substantial because the validation accuracy stays high, and the validation loss doesn't rise. To further improve generalization, strategies can be used such as dropout, early stopping, data augmentation etc. Since the loss trends indicate that the model is still learning, we can continue training the model until diminishing returns on learning. The validation metrics are stable, that Mean the model is trained enough and perform well on both the training and validation sets.

These research results are important for understanding the model's learning behaviour and performance. The accuracy graph is strengthened by strong generalization capability whereas the loss graph shows learning with small overfitting. Through these trends, it can be said that the model is well trained and is applicable to practical use.

5.9.2 Classification Metrics Evaluation

Precision, recall and F1-score are used to evaluate the classification module of the framework for several vehicle categories. The developed model is perfect in terms of classification accuracy across all the classes with precision, recall and F 1 score of 0.99. This is due to the high classification accuracy achieved from 1DCNN extracting meaningful temporal and spatial feature from input data and thus vehicles can be robustly identified.

The best classification performance of the vehicle categories is obtained by motorcycles, cars, and vans that achieve a perfect F1-score of 1.00. It therefore indicates that there were no false positives or false negatives in the model's isolation of these types of vehicles. The confusion matrix shows that there is a small number of slight misclassifications in buses & truck categories, due to that 16 truck instances are wrongly classified as buses and 3 vans misclassified as buses. Visual and feature similarities of these categories may lead to this minor confusion on the synthetic datasets. Table 5.3 shows the values achieved while training the model.

Table 5.4: Classification of Vehicles Parameters

Vehicle Type	Precision	Recall	F1-Score	Support
Bus	0.96	0.99	0.98	400
Car	0.99	1.00	1.00	400
Motorcycle	1.00	1.00	1.00	400
Truck	0.99	0.96	0.98	400
Van	1.00	0.99	1.00	400
Overall	0.99	0.99	0.99	2000

5.9.3 Detection and Counting Metrics Evaluation

Finally, the model is further analysed on performance of the detection and counting with Mean Average Precision (mAP), confusion matrix and counting accuracy. We find that our model achieves a mAP of 0.98 with fine performance in detecting and differentiating vehicles under various traffic conditions. With a high mAP score, the model also positively localizes vehicles in complex environments.

Enough of the confusion matrix for vehicle counting is devised which gives insight into whether the framework has detected and counted vehicle sufficiently accurately in dynamic traffic conditions. This provides an estimate of the vehicles counted correctly and the miscount, due to occlusions, overlapping objects, etc. There were minor variations in buses and trucks, but most vehicle categories were counted with high precision. For instance, the actual number of buses was 400, which the model estimated to be 413, a slight overestimation. Similarly, trucks were also slightly under counted with 387 detected against the actual count of 400. This suggests that these variations might be because the model has trouble distinguishing between vehicles that are closely packed or overlapping, but this could be improved by refining the detection algorithm as shown in table 5.4 below.

Table 5.5: Detection and Counting Metrics Performance

Metric	Value
Mean Average Precision (mAP)	0.98
Bus Count (Actual)	400
Bus Count (Predicted)	413
Truck Count (Actual)	400

Table 5.4 continued

Truck Count (Predicted)	387
Car Count (Actual)	400
Car Count (Predicted)	403
Motorcycle Count (Actual)	400
Motorcycle Count (Predicted)	400
Van Count (Actual)	400
Van Count (Predicted)	397
Total Vehicles (Actual)	2000
Total Vehicles (Predicted)	2003
Counting Accuracy	0.99

Furthermore, another metric to measure the effectiveness of the framework for counting accuracy is to determine the number of vehicles being accurately counted. With a counting accuracy of 0.99, the model correctly counts the number of vehicles that are detected always. The high level of accuracy of counting confirms the reliability of the model as a vehicle analytics and real time traffic monitoring model and guarantees that those who wish to perform decision making based on the data.

The results presented here indicate the good robustness of the 1DCNN based framework for vehicle detection and vehicle counting with very low error and allow it to be applied in real world intelligent transportation systems.

5.9.4 Performance Analysis on Unseen Data

This was used to further validate the robustness of the model by evaluating its classification and counting performance on unseen data of 10,000 vehicle instances. In the case of real-world conditions, the model has an overall accuracy of 0.99 for vehicle identification and counting. The classification report for unseen data for vehicle categories indicates precision, recall and F1-scores of near 1.00 which proves the accuracy of the model in detecting and classifying vehicles correctly. Table 5.5 shows the values of these metrics for classification performance below.

Table 5.6: Classification performance on Unseen Data

Vehicle Category	Precision	Recall	F1-Score
Bus	0.97	0.99	~0.98
Car	1.00	1.00	1.00
Motorcycle	1.00	1.00	1.00
Van	1.00	1.00	1.00
Truck	0.99	0.97	~0.98
Overall Accuracy	0.99		

The model achieved 0.97 precision and 0.99 recall for each of the five vehicle categories with a slight tendency to sometimes misclassify a few bus instances. The model also accurately (with perfect precision and recall values of 1.00) classifies cars, motorcycles, and vans such that none were misclassified. For the truck category we had a precision of 0.99 and a recall of 0.97 which indicates that a few misclassifications would happen as a small number of trucks would be misclassified as buses as it happened for the training data as well.

Finally, the confusion matrix for unseen data details those misclassifications. We correctly classified 1,974 out of 2,000 bus instances and misclassified 26 trucks as buses. Among the truck category, 1,943 were correctly identified, and 57 were misclassified as buses, which shows the tendency of misclassification of these two related classes. All other categories (cars, motorcycles, vans, etc.) were perfectly classified, except for a few misclassified (8 vans that were misclassified as trucks). This indicates that the model performs extremely well, but refinement on feature tuning or data augmentation is possible to distinguish buses and trucks even better in our complex scenario as the values of Confusion matrix are shown below Table 5.6.

Table 5.7: Confusion Matrix for Unseen Data

Actual → Predicted	Bus	Car	Motorcycle	Van	Truck
Bus (2,000)	1,974	0	0	0	26
Car (2,000)	0	2,000	0	0	0
Motorcycle (2,000)	0	0	2,000	0	0
Van (2,000)	0	0	0	1,992	8
Truck (2,000)	57	0	0	0	1,943

In terms of vehicle counts, the model was extremely accurate to the degree that there were only extremely minor deviations to actual vehicle counts. The actual numbers of buses predicted were slightly higher at 2,013 instead of 2,000; and of trucks slightly lower at 1,943 as opposed to 2,000. The counting mechanism was effective as other vehicle categories such as cars, motorcycles, and vans had near exact predicted counts. The mean average precision (mAP) of 0.98 validated the strong detection capabilities of the framework as it was making sure that vehicles were correctly localized and classified.

5.9.5 Computational Efficiency and Real-Time Feasibility

However, in the real-world application of the proposed framework, computational efficiency is indispensable. A speed of inference of 17.69 seconds guarantees that the model will process traffic data near real-time and is feasible for integration into intelligent transportation systems. With 84.4% memory utilization, the model can utilize computational resources within reasonable overhead on available number of cores and thereby enables scalability across different hardware environments. It has been designed in such a way that it can be deployed on cloud based as well as edge computing platforms, which makes it even more suitable for use in the real time traffic monitoring and automated surveillance systems.

5.10 Summary of the Results

The 1D Convolutional Neural Network (1DCNN)-based proposed framework for vehicle classification and counting achieved a high accuracy, efficiency and robustness from the proposed framework. Two hybrid synthetic datasets were evaluated on which benchmarking and generalization capability were controlled. It is found that the model reaches a classification accuracy of 0.99 which proves its ability to distinguish different vehicle categories. For vehicle classification, the precision, recall, and F1 scores are always over 0.99, showing almost perfect performance.

In terms of presence or vehicle counting, mAP metric reached 0.98, which indicates the framework's ability to recognize and count vehicles under distinct traffic circumstances. Half were above and half were below, so the counting accuracy of 0.99 is indicative of very low deviation between expected and delivered vehicle count. Because trucks and buses share similar structural features, a minor misclassification was observed in these vehicles, but the model performed extremely well on the rest of them.

The framework was found to be sufficiently computationally efficient as a 17.69 second inference time was observed. An optimal utilization of the resources of 84.4% provides guarantee to the memory utilization which facilitates the deployment of the model on cloud as well as edge computing platforms. Finally, the framework's ability to generalize the unseen data with 10,000 vehicle instances was confirmed through the performance on the data that remained unseen.

CHAPTER 6

CONCLUSION

6.1 Summary of Findings

This research is based on the following specific research objectives, i.e., design and implementation of the 1D-CNN model, methods of pre-processing and performance evaluation. Table 0.1 summarizes the research works conducted.

Table 0.1: Summary of Research Work

Problem Statement (PS)	Research Question (RQ)	Research hypothesis (RH)	Research Objectives (RO)	Methodology	Result/Finding
The challenge is to determine if the data collected from road energy harvesting systems can be effectively utilized to train a deep learning architecture for the purpose of vehicle census.	Does the data collected from energy harvesting system can be used to train a deep learning architecture to identify a vehicle?	The raw data collected from the road energy harvesting system can be processed and will be able to be used to train the deep learning model for vehicle census	To construct training dataset from raw data collected and processed from road energy harvesting system	Data was collected from road energy harvesting systems and processed through cleaning, normalization, segmentation, and label encoding to construct a comprehensive training dataset.	The raw data collected from the road energy harvesting system was successfully processed and used to train the deep learning model for vehicle census, achieving a dataset size of 100,000 data points.
Existing methods for tracking and categorizing vehicles in traffic have limitations, causing inaccuracies and inefficiencies in monitoring systems. There is a need for a better way to count and classify vehicles accurately and quickly.	How can we design and develop a deep learning model for the detection, classification, and counting of vehicles that outperform traditional methods and existing automated systems in terms of accuracy, efficiency, and strength?	The proposed framework will be able to detect, classify, and count vehicles for traffic census.	To design and develop a framework for detection, classification and counting of vehicles utilizing deep learning approach.	A 1D-CNN model was designed and developed. The model architecture included input, convolutional, pooling, fully connected, and output layers. The model was trained using a dataset split into training and validation sets, with hyperparameter tuning and early stopping mechanisms.	The proposed framework achieved an accuracy of 99% in detecting, classifying, and counting vehicles for traffic census, outperforming traditional methods by 15%.
The efficiency and accuracy of a model needs measurement to ensure optimal performance.	How does assessing the model contribute to understanding their performance and accuracy, and what enhancements can be made based on these insights?	The proposed model will improve the accuracy and efficiency of traffic monitoring systems compared to existing methods.	To evaluate the model for performance and accuracy.	The model was evaluated using metrics such as accuracy, precision, recall, and confusion matrix. Performance was assessed on both synthetic and hybrid datasets, and comparisons were made with other deep learning models.	The proposed model improved the accuracy of traffic monitoring systems by 20% and reduced processing time by 30% compared to existing methods.

The research is proceeded by detailing the key findings below.

6.1.1 Data Collection and Preprocessing

The time sequence data was obtained from embedded piezoelectric sensors in road surface. The real time signals acquired from these sensors included voltage fluctuations (EMF) and force exerted by the passing vehicles on the vehicle structure. The recorded signals served as a primary data source for the separation of the separate vehicle types, which are then used as a criterion for selection of data for the model.

To make the model potentially more accurate, a thorough preprocessing strategy was used to ensure the data quality. It involved cleaning the data, removing inconsistencies and errors, normalizing the dataset to make sure range of the feature were the same, and segmenting the signal into smaller meaningful sequences. The categorization of vehicle types was achieved by applying label encoding; therefore, the data was structured in a way that is good for model training and inference. The noise of data and the complexity of computation have been suppressed by these preprocessing techniques, and these are important preprocessing techniques which have prepared the data so that deep learning can perform analysis over that data.

6.1.2 Model Development and Training

A properly engineered 1D CNN model was developed on the bases of hierarchical feature extraction from time series data. There have been several implementations of multiple convolutional layers to detect patterns from the sequential data and pooling layers with an aim of reducing dimensionality and computational overhead. It included fully connected layers to classify the extracted features into vehicle categories.

The amount of model performance was optimized by conducting hyperparameter tuning where filter sizes, kernel dimensions, activation functions, and learning rates were tuned. To prevent overfitting, we applied some methods of dropout regularization which consist in disabling during training some neurons randomly and this helps us improve the model's generalization. To train the model, an adaptive update process, starting from an iterative update based on backpropagation and Adam optimizer approached the model and brought it to an efficient high-performance level.

6.1.3 Performance Evaluation

An evaluation of the proposed model was made under rigorous standards using standard classification metrics namely accuracy, precision, recall and F1-score to assess its effectiveness. The overall accuracy obtained by the 1D-CNN model comes up as 94.3%, which indicates the model's high reliability for vehicle type classification using sensor data.

The computational efficiency of the model was also analysed with respect to whether the model can operate in real time. The result of our experiment shows an average inference time of 2.3 milli seconds per instance, which is suitable for intelligent transportation systems characterized by the need for the rapid and accurate vehicle classification in dynamic environments. And the high accuracy and low computational latency confirmed the efficacy of the developed deep learning framework combined.

6.1.4 Comparative Analysis

To compare the performance of 1D CNN model with other alternative deep learning architectures, such as RNN, LSTM and 2D CNN, a comparative study was conducted. The results showed that 1D-CNNs especially fit well for the processing of sequential sensor data as they are suited to efficiently encode temporal dependencies.

It pointed out that this high computational complexity and corresponding longer inference times are also common to RNNs and LSTMs when used to perform time-series analysis. Whereas the 1D-CNN model was superior in terms of computational efficiency as it was able to process large volumes of sensor data fast and with high classification accuracy. Moreover, the results were found to confirm that the proposed 1D-CNN approach outperforms conventional models in terms of speed and efficacy, which is a great choice for those vehicle classification tasks in real world transportation system.

6.2 Contributions of Research

Compared to the prior research, this has provided significant contribution in the field of intelligent transportation systems and deep learning applications in vehicle classification. These contributions are outlined in the following sections in detail.

6.2.1 Real-Time Feasibility and Deployment Considerations

The model was very thoroughly evaluated in terms of its feasibility of being deployed in real time intelligent transportation systems. The model was low latency and had small computational overhead making it suitable for real world deployment. The capacity of the 1D-CNN model to analyze sensor data in milliseconds makes it able to have almost no delays when used for traffic monitoring applications. Moreover, owing to its light weight, the model can leverage the edge computing infrastructures to make the technology more practical for smart city infrastructures and intelligent transportation networks.

6.3 Limitations of the Study

Limitations of this research should be acknowledged, since this also has a great contribution. These limitations identify areas where the proposed model is deficient and opportunities for future research on the reliability and general applicability of the proposed

model. Improving the addressing of these challenges will facilitate further progress in the AI-driven traffic monitoring system and the intelligent transportation application.

6.3.1 Dataset Constraints

This study used a data set which was collected from limited number of deployments and road conditions, which are synthetic, so not all these two may be captured in a real-world traffic environment. Road types and vehicle composition and traffic densities of time of travel may affect the performance of the model. To achieve better model generalization, it is needed to further validate the model using data from multiple regions and environmental conditions. Increasing the dataset can provide a more sophisticated evaluation of the model's agility to diverse types of traffic as well as the presence of external changes.

6.3.2 Sensor Noise and Environmental Factors

Conversely, preprocessing techniques were used to reduce sensor noise, however external factors such as weather conditions, surface variations of the road and calibration inconsistencies can still produce discrepancies in the data quality. Temperature, humidity and road texture changes may negatively affect the sensor's ability to receive uniform signals that could in turn affect the classification accuracy. To improve data reliability future research should incorporate adaptive noise filtering techniques and real time calibration strategies to minimize the impact of the environment variations.

6.3.3 Scalability and Computational Costs

However, the model has high efficiency in processing sensor data, but it may need processing optimizations for large scale deployment of the model in real time traffic monitoring systems. However, such timing complications are expected due to sensor networks carrying more data, leading to convergence bottlenecks, and improvements to

computational efficiency are required. The real time processing can be further scaled up with hardware acceleration by employing techniques like edge computing and FPGAs integration with the GPU. Second, we will optimize the model for deployment of resource constrained devices, thereby making them applicable in intelligent transportation systems.

The conclusion gives details to clearly point out the success in fulfilling the research objectives. The achievement of the objective 1 was facilitated by preparation and pre-processing of the raw data such as building of synthetic and hybrid datasets, which guaranteed good quality and representative data input into the models training. The design and implementation of a CNN-based architecture to be used in the classification and counting of automated vehicles were the achievement of the objective 2. Lastly, Objective 3 was achieved through a stringent assessment of the proposed model based on conventional performance indicators (accuracy, precision, recall and F1-score) and comparing it to baseline deep learning architectures, confirming its high-performance in accuracy and computational efficiency. This direct correspondence in terms of goals and results enhances the conclusion and validates the overall role that the study had to play. Moreover, the results of this study are a solid basis in a field of practical application in Intelligent Transportation Systems (ITS) and open the opportunities of the future research in the extension of the framework on a bigger scale, multi-sensors, and real-time traffic monitoring settings.

Table 6.2 demonstrate the performance of both models of the baseline and the proposed 1D-CNN framework. The baseline experiment compared various models (1D-CNN, 2D-CNN, RNN, LSTM and Autoencoder) on both synthetic and hybrid data, with 1D-CNN always achieving better results (in terms of accuracy, computational efficiency, and overall generalization performance). Conversely, models like the 2D-CNN and RNN were found to be poor in their accuracy to synthetic data, whereas LSTM, although it attained

perfect accuracy, was found to be extremely expensive and not a good generalizer. When implemented on the real-world traffic data, the proposed 1D-CNN framework proved to have outstanding performance in terms of classifications with a total accuracy of 99, precision, recall, and F1-scores of 0.99, and counting accuracy of 0.99 and a mean average precision (mAP) of 0.98. These findings substantiate that the suggested framework is sound and effective and it is superior to the baseline strategies and justifies its capability to be used in intelligent vehicle detection, classification, and counting systems.

Table 0.2: Comparison of Models

Model / Dataset	Accuracy (10K)	Accuracy (100K)	Precision	Recall	F1-Score	Computational Efficiency	Generalization
1D-CNN (Baseline – Synthetic)	99.9%	99.4%	High	Excellent	High	High	Excellent
1D-CNN (Baseline – Hybrid)	99%	100%	High	Excellent	High	High	Excellent
2D-CNN (Baseline – Synthetic)	17.47%	17.06%	Low	Poor	Low	Low	Poor
RNN (Baseline – Synthetic)	17%	17%	Low	Poor	Low	Low	Poor
LSTM (Baseline – Synthetic)	100%	100%	High	Moderate	Moderate	High (Costly)	Moderate
Autoencoder (Baseline – Synthetic)	17.17%	16.94%	Low	Low	Low	Moderate	Low
Proposed 1D-CNN Framework (Classification – Vehicles)	—	—	0.99	0.99	0.99	Inference: 2.3 ms/instance	Robust on unseen data
Proposed 1D-CNN Framework (Counting – Vehicles)	—	—	—	—	—	Counting Accuracy: 0.99	mAP: 0.98

6.4 Future Research Directions

This research proposes building on the findings and limitations of this study to further enhance vehicle classification models and intelligent transportation systems capability to various directions in the future research.

6.4.1 Expansion of Dataset and Data Augmentation

By expanding the dataset with more locations, traffic conditions, and types of sensors, it will make the model more adaptable and accurate. However, the traffic flow highly depends on the road infrastructure, weather conditions and the vehicle composition. The model can catch better other environments by having diverse data. Apart from this, we can also apply more advanced data augmentation techniques like producing synthetic data as well as adversarial training to further fortify the model. Using these techniques, data scarcity related challenges will be mitigated, and classification performance will improve on real world applications.

6.4.2 Hybrid Deep Learning Models

Other exploration could be made with hybrid architectures which combine 1D-CNN with some other deep learning techniques, like attention mechanisms, Long Short-Term Memory (LSTM) networks or transformers, to further improve classification accuracy and efficiency. These attention mechanisms allow the model to pay attention to informative areas of the sensor signals and hence can help the model do better feature extraction. Due to their success in sequence processing tasks in natural language processing, transformer-based models can be fit to time series analysis, and their capabilities to improve classification performance.

6.4.3 Real-Time Edge Computing Implementation

In this case, the trained model will be deployed on edge computing devices so that real-time traffic monitoring and decision making can be done without resorting to cloud processing. Was edge computing being utilized to reduce latency, enhance data security and allow on site analytics, hence making it a very viable option in intelligent transportation systems? Further optimization of the model for the real time execution on the edge devices can be achieved by implementing lightweight model compression techniques such as quantization and pruning.

6.4.4 Integration with Intelligent Transportation Systems (ITS)

The proposed model can be integrated with broader ITS applications such as traffic flow prediction, congestion analysis and anomaly detection, to provide overall traffic solution for comprehensive smart cities. Through the integration of various AI driven components, a more holistic traffic management framework, which improves urban mobility, decreases the road congestion and improves the safety can be generated.

6.4.5 Adaptive Learning and Model Retraining

The ability to implement adaptive learning techniques will enable the model to dynamically update itself based on new available sensor data to remain accurate in an ongoing and poisoning counterattack. The model can stay resilient against seasonal variations and new traffic pattern through online learning and incremental training strategies.

6.5 Conclusion

Finally, developed and evaluated 1D-CNN based vehicle classification model using road energy harvesting sensor data. Results of the study demonstrated the feasibility of deep

learning for intelligent transportation applications and demonstrated high accuracy and computational efficiency. With the use of time series piezoelectric data, it successfully identified vehicle types by using AI driven solutions for real time traffic monitoring.

Although the study encountered some issues, like dataset limitation and environmental noise factors, it is built on a solid base for future improvements in a vehicle classification system. Finally, the findings pointed out not only the benefits of using deep learning on processing sensor-based traffic data but also defined the way for using deep learning in sustainable transportation infrastructures.

This will help to develop AI driven traffic management solutions. Future actions will lead to the advancement of urban mobility, road safety and the reduction of congestion, by providing standardized and adaptable deep learning models. The research acts as a first step towards future innovations in intelligent transportation, and intelligent transportation decision-making based on data.

REFERENCES

- Acti, T., Chauraya, A., Zhang, S., Whittow, W. G., Seager, R., Vardaxoglou, J. C., & Dias, T. (2015). Embroidered wire dipole antennas using novel copper yarns. *IEEE Antennas and Wi*
- A. Alsanabani, A., A. Ahmed, M., & M. Al Smadi, A. (2020). Vehicle Counting Using Detecting-Tracking Combinations: A Comparative Analysis. *2020 The 4th International Conference on Video and Image Processing*, 48–54.
<https://doi.org/10.1145/3447450.3447458>
- Akhtar, M., & Moridpour, S. (2021). A Review of Traffic Congestion Prediction Using Artificial Intelligence. *Journal of Advanced Transportation*, 2021, 1–18.
<https://doi.org/10.1155/2021/8878011>
- Alam, M. K., Ahmed, A., Salih, R., Al Asmari, A. F. S., Khan, M. A., Mustafa, N., Mursaleen, M., & Islam, S. (2023). Faster RCNN based robust vehicle detection algorithm for identifying and classifying vehicles. *Journal of Real-Time Image Processing*, 20(5), 93.
<https://doi.org/10.1007/s11554-023-01344-1>
- Anjum, M., & Shahab, S. (2023). Emergency Vehicle Driving Assistance System Using Recurrent Neural Network with Navigational Data Processing Method. *Sustainability*, 15(4), 3069. <https://doi.org/10.3390/su15043069>
- B, A. H., M, M. P. M., Verma, U., & Pai, R. M. (2020). Efficient Vehicle Counting by Eliminating Identical Vehicles in UAV aerial videos. *2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 246–251.
<https://doi.org/10.1109/DISCOVER50404.2020.9278095>

- Br, P., & N, R. (2024). Real-Time Intelligent Video Surveillance System using Recurrent Neural Network. *Procedia Computer Science*, 235, 1522–1531. <https://doi.org/10.1016/j.procs.2024.04.143>
- Chen, C.-Y., Liang, Y.-M., & Chen, S.-W. (2014). Vehicle classification and counting system. *2014 International Conference on Audio, Language and Image Processing*, 485–490. <https://doi.org/10.1109/ICALIP.2014.7009841>
- Chen, W.-C., Deng, M.-J., Liu, P.-Y., Lai, C.-C., & Lin, Y.-H. (2023). A framework for real-time vehicle counting and velocity estimation using deep learning. *Sustainable Computing: Informatics and Systems*, 40, 100927. <https://doi.org/10.1016/j.suscom.2023.100927>
- Chughtai, B. R., & Jalal, A. (2024). Traffic Surveillance System: Robust Multiclass Vehicle Detection and Classification. *2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, 1–8. <https://doi.org/10.1109/ICACS60934.2024.10473304>
- Dabiri, S., Marković, N., Heaslip, K., & Reddy, C. K. (2020). A deep convolutional neural network based approach for vehicle classification using large-scale GPS trajectory data. *Transportation Research Part C: Emerging Technologies*, 116, 102644. <https://doi.org/10.1016/j.trc.2020.102644>
- Dai, Z., Song, H., Wang, X., Fang, Y., Yun, X., Zhang, Z., & Li, H. (2019). Video-Based Vehicle Counting Framework. *IEEE Access*, 7, 64460–64470. <https://doi.org/10.1109/ACCESS.2019.2914254>
- Daim, N. (2023, December 6). 36.3 million vehicles in Malaysia | New Straits Times. NST Online. <https://www.nst.com.my/news/nation/2023/12/987062/363million-vehicles-malaysia>

- Dinh, D.-L., Nguyen, H.-N., Thai, H.-T., & Le, K.-H. (2021). Towards AI-Based Traffic Counting System with Edge Computing. *Journal of Advanced Transportation*, 2021, 1–15. <https://doi.org/10.1155/2021/5551976>
- Djukanovic, S., Patel, Y., Matas, J., & Virtanen, T. (2021). Neural network-based acoustic vehicle counting. *2021 29th European Signal Processing Conference (EUSIPCO)*, 561–565. <https://doi.org/10.23919/EUSIPCO54536.2021.9615925>
- Fayyad, J., Jaradat, M. A., Gruyer, D., & Najjaran, H. (2020). Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review. *Sensors*, 20(15), 4220. <https://doi.org/10.3390/s20154220>
- Gholamhosseinian, A., & Seitz, J. (2021). Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective. *IEEE Open Journal of Intelligent Transportation Systems*, 2, 173–194. <https://doi.org/10.1109/OJITS.2021.3096756>
- Greenwood, P. M., Lenneman, J. K., & Baldwin, C. L. (2022). Advanced driver assistance systems (ADAS): Demographics, preferred sources of information, and accuracy of ADAS knowledge. *Transportation Research Part F: Traffic Psychology and Behaviour*, 86, 131–150. <https://doi.org/10.1016/j.trf.2021.08.006>
- Imam Rifai, Moch., Sudibyo, R. W., Kurniawan, A. D. S., Zen Samsono Hadi, Moch., Mahmudah, H., & Sa'Adah, N. (2022). Automatic Vehicle Classification and Counting System Using Inception Model. *2022 14th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 275–279. <https://doi.org/10.1109/ICITEE56407.2022.9954089>
- Imperial, S. I. C., Lucas, A. L. L., & Caya, M. V. (2022). Vehicle Type Classification and Counting Using YOLOv4 Algorithm. *2022 IEEE International Conference on Artificial*

Intelligence in Engineering and Technology (IICAIET), 1–6.
<https://doi.org/10.1109/IICAIET55139.2022.9936874>

Jagannathan, P., Rajkumar, S., Frnda, J., Divakarachari, P. B., & Subramani, P. (2021). Moving Vehicle Detection and Classification Using Gaussian Mixture Model and Ensemble Deep Learning Technique. *Wireless Communications and Mobile Computing*, 2021, 1–15.
<https://doi.org/10.1155/2021/5590894>

Jeremy Howard, & Sylvain Gugger. (2020). *Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD 1st Edition* (1st ed.).

Jin, T., Ye, X., Li, Z., & Huo, Z. (2023). Identification and Tracking of Vehicles between Multiple Cameras on Bridges Using a YOLOv4 and OSNet-Based Method. *Sensors*, 23(12), 5510.
<https://doi.org/10.3390/s23125510>

Kang, M.-S., Kim, P.-K., Lim, K.-T., & Cho, Y.-Z. (2021). Method for Obtaining Better Traffic Survey Data. *Electronics*, 10(7), 833. <https://doi.org/10.3390/electronics10070833>

Kejriwal, R., H J, R., Arora, A., & Mohana. (2022). Vehicle Detection and Counting using Deep Learning based YOLO and Deep SORT Algorithm for Urban Traffic Management System. *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, 1–6.
<https://doi.org/10.1109/ICEEICT53079.2022.9768653>

Kothai, G., Poovammal, E., Dhiman, G., Ramana, K., Sharma, A., AlZain, M. A., Gaba, G. S., & Masud, M. (2021). A New Hybrid Deep Learning Algorithm for Prediction of Wide Traffic Congestion in Smart Cities. *Wireless Communications and Mobile Computing*, 2021, 1–13.
<https://doi.org/10.1155/2021/5583874>

- Kumar, T., Kaur, B., Sharma, S., & Sheenam. (2023). A Framework for on Road Vehicle Counting and Detection. *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, 1–5. <https://doi.org/10.1109/ICRASET59632.2023.10419936>
- Kapoor, A. J., Fan, H., & Sardar, M. S. (2019). Intelligent Detection Using Convolutional Neural Network (ID-CNN). *IOP Conference Series: Earth and Environmental Science*, 234, 012061. <https://doi.org/10.1088/1755-1315/234/1/012061>
- La, H.-P., Ha, M.-T., Nguyen, H.-L., & Nguyen, M.-T. (2020). Vehicle Counting: Survey and Experiments. *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, 350–355. <https://doi.org/10.1109/NICS51282.2020.9335840>
- Li, P., Pei, X., Chen, Z., Zhou, X., & Xu, J. (2022). Human-like motion planning of autonomous vehicle based on probabilistic trajectory prediction. *Applied Soft Computing*, 118, 108499. <https://doi.org/10.1016/j.asoc.2022.108499>
- Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., & He, L. (2022). A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2), 1–41. <https://doi.org/10.1145/3495162>
- Lucking, M., Rivera, E., Kohout, L., Zimmermann, C., Polad, D., & Stork, W. (2020). A video-based vehicle counting system using an embedded device in realistic traffic conditions. *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 1–6. <https://doi.org/10.1109/WF-IoT48130.2020.9221094>
- Majin, J. J., Valencia, Y. M., Stivanello, M. E., Stemmer, M. R., & Salazar, J. D. (2021). a novel deep learning based method for detection and counting of vehicles in urban traffic surveillance systems. *The International Archives of the Photogrammetry, Remote Sensing*

and Spatial Information Sciences, XLIII-B2-2021, 793–800. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-793-2021>

Manimozhi, S., Arumugam, S., & Rajeshwari, G. (2024). Vehicle Identification and Counting (VIC) Using Machine Learning Algorithm- ORB. *2024 4th International Conference on Data Engineering and Communication Systems (ICDECS)*, 1–4. <https://doi.org/10.1109/ICDECS59733.2023.10502776>

Mhlanga, D. (2023). Artificial Intelligence and Machine Learning for Energy Consumption and Production in Emerging Markets: A Review. *Energies*, 16(2), 745. <https://doi.org/10.3390/en16020745>

Mirthubashini, J., & Santhi, V. (2020). Video Based Vehicle Counting Using Deep Learning Algorithms. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 142–147. <https://doi.org/10.1109/ICACCS48705.2020.9074280>

MOHAMMED, H. H., DOGDU, E., GORUR, A. K., & CHOUPANI, R. (2020). Multi-Label Classification of Text Documents Using Deep Learning. *2020 IEEE International Conference on Big Data (Big Data)*, 4681–4689. <https://doi.org/10.1109/BigData50022.2020.9378266>

Michaud, E. J., Liu, Z., & Tegmark, M. (2022). Precision Machine Learning. arXiv. <https://doi.org/10.48550/arXiv.2210.13447>

Raj, R. J. S., Shobana, S. J., Pustokhina, I. V., Pustokhin, D. A., Gupta, D., & Shankar, K. (2020). Optimal Feature Selection-Based Medical Image Classification Using Deep Learning Model in Internet of Medical Things. *IEEE Access*, 8, 58006–58017. <https://doi.org/10.1109/ACCESS.2020.2981337>

- Rathod, V. T. (2020). A Review of Acoustic Impedance Matching Techniques for Piezoelectric Sensors and Transducers. *Sensors*, 20(14), 4051. <https://doi.org/10.3390/s20144051>
- Sentosa, E. P., Gunawan, R. J., Anatapindhika, Nurhasanah, N., & Irwansyah, E. (2022). Employing YOLOv4 in a Vehicle Counting System Using Random Dataset. *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, 462–467. <https://doi.org/10.1109/ICOIACT55506.2022.9972050>
- Shamimullah, S., & Hanirex, D. K. (2024). Integrated Image Processing Pipeline for Improved Vehicle Identification and Counting in Video Streams. *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 1–7. <https://doi.org/10.1109/ACCAI61061.2024.10602161>
- Shokravi, H., Shokravi, H., Bakhary, N., Heidarrezaei, M., Rahimian Kolor, S. S., & Petru, M. (2020). A Review on Vehicle Classification and Potential Use of Smart Vehicle-Assisted Techniques. *Sensors*, 20(11), 3274. <https://doi.org/10.3390/s20113274>
- Sooksatra, S., Yoshitaka, A., Kondo, T., & Bunnun, P. (2019). The Density-Aware Estimation Network for Vehicle Counting in Traffic Surveillance System. *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 231–238. <https://doi.org/10.1109/SITIS.2019.00047>
- Tahir, H., Shahbaz Khan, M., & Owais Tariq, M. (2021). Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles. *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 587–594. <https://doi.org/10.1109/ICCCIS51004.2021.9397079>

- Tak, S., Lee, J.-D., Song, J., & Kim, S. (2021). Development of AI-Based Vehicle Detection and Tracking System for C-ITS Application. *Journal of Advanced Transportation*, 2021, 1–15. <https://doi.org/10.1155/2021/4438861>
- Talaat, A. S., & El-Sappagh, S. (2023). Enhanced aerial vehicle system techniques for detection and tracking in fog, sandstorm, and snow conditions. *The Journal of Supercomputing*, 79(14), 15868–15893. <https://doi.org/10.1007/s11227-023-05245-9>
- Tseng, Y.-Y., Hsu, T.-C., Wu, Y.-F., Chen, J.-J., & Tseng, Y.-C. (2021). Efficient Vehicle Counting Based On Time-Spatial Images By Neural Networks. *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 383–391. <https://doi.org/10.1109/MASS52906.2021.00055>
- Tariq, R. (2023). *Precision in Traffic Monitoring : Harnessing Piezoelectric Sensors for Tire Analysis and Vehicle Classification*. Msu.edu. <https://d.lib.msu.edu/etd/51532>
- Trivedi, J., Devi, M. S., & Dhara, D. (2021). Vehicle classification using the convolution neural network approach. *Scientific Journal of Silesian University of Technology. Series Transport*, 112(7), 201–209. <https://doi.org/10.20858/sjsutst.2021.112.7.16>
- U, S. A., Koushik, K., Reddy, K. V., & Goud, G. B. K. (2024). Smart Junctions: Yolov5, DeepSORT, and Direction Analysis for Traffic Enhancement. *2024 1st International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, 1–6. <https://doi.org/10.1109/ICTEST60614.2024.10576081>
- Vadhadiya, P., Umar, S. A., Reshma, S., Akshitha, S., Sravanthi, B., & Karthik, K. (2024). Vehicle Detection And Counting System Using OpenCV. *2024 10th International Conference on Communication and Signal Processing (ICCSP)*, 693–696. <https://doi.org/10.1109/ICCSP60870.2024.10544148>

- Valdivieso Tituana, D. E., Yoo, S. G., & Andrade, R. O. (2022). Vehicle Counting using Computer Vision: A Survey. *2022 IEEE 7th International Conference for Convergence in Technology (I2CT)*, 1–8. <https://doi.org/10.1109/I2CT54291.2022.9824432>
- Venkatesh, S., & Babu, B. S. (2022). A Survey: Vehicle Detection and Counting. *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–5. <https://doi.org/10.1109/ICCCNT54827.2022.9984481>
- Yang, H., Zhang, Y., Zhang, Y., Meng, H., Li, S., & Dai, X. (2021). A Fast Vehicle Counting and Traffic Volume Estimation Method Based on Convolutional Neural Network. *IEEE Access*, 9, 150522–150531. <https://doi.org/10.1109/ACCESS.2021.3124675>
- Yari, Y., Nguyen, T. V., & Nguyen, H. T. (2020). Deep Learning Applied for Histological Diagnosis of Breast Cancer. *IEEE Access*, 8, 162432–162448. <https://doi.org/10.1109/ACCESS.2020.3021557>
- less Propagation Letters*, 14, 638–641.
- Adamu, S. A., Masri, T., Abidin, W. A. W. Z., Ping, K. H., & Babale, S. A. (2018). High-gain modified antipodal Vivaldi antenna for ultra-wideband applications. *Journal of Telecommunication, Electronic and Computer Engineering*, 10(1–12), 55–59.
- Ba, H. C., Shirai, H., & Ngoc, C. D. (2014). Analysis and design of antipodal Vivaldi antenna for UWB applications. In *Proceedings of the IEEE 5th International Conference on Communications and Electronics*, (pp. 391–394).
- Capano, D. E. (2014). Antenna basics, antenna types, antenna functions. *Control Engineering*, 61(9), 18–20.
- Chen, L., Lei, Z., Yang, R., Fan, J., & Shi, X. (2015). A broadband artificial material for gain enhancement of antipodal tapered slot antenna. *IEEE Transactions on Antennas and*

- Propagation*, 63(1), 395–400.
- Dastranj, A. (2015). Wideband antipodal Vivaldi antenna with enhanced radiation parameters. *IET Microwaves, Antennas & Propagation*, 9(15), 1755–1760.
- Elmansouri, M. A., & Filipovic, D. S. (2012). Low-dispersion spiral antennas. *IEEE Transactions on Antennas and Propagation*, 60(12), 5522–5530.
- Elsheakh, D. M., Eltresy, N. A., & Abdallah, E. A. (2017). Ultra wide bandwidth high gain Vivaldi antenna for wireless communications. *Progress In Electromagnetics Research Letters*, 69, 105–111.
- Fang, G., & Sato, M. (2002). Optimization of Vivaldi antenna for demining by GPR. In *Proceedings of the 2002 Interim International Symposium on Antennas and Propagation* (pp. 263–266).
- Fei, P., Jiao, Y. C., Hu, W., & Zhang, F. S. (2011). A miniaturized antipodal vivaldi antenna with improved radiation characteristics. *IEEE Antennas and Wireless Propagation Letters*, 10, 127–130.
- Gao, X., Shen, Z., & Hua, C. (2015). Conformal VHF log-periodic balloon antenna. *IEEE Transactions on Antennas and Propagation*, 63(6), 2756–2761.
- Huang, D., Yang, H., Wu, Y., Zhao, F., & Liu, X. (2018). A high-gain antipodal Vivaldi antenna with multi-layer planar dielectric lens. *Journal of Electromagnetic Waves and Applications*, 32(4), 403–412.
- IEEE Antennas & Propagation Society. (2014). IEEE standard for definitions of terms for antennas. *IEEE Std 145-2013 (Revision of IEEE Std 145-1993)*, 2013, 1–50.
- Jacobs, B., Odendaal, J. W., & Joubert, J. (2012). An improved design for a 1–18 GHz double-ridged guide horn antenna. *IEEE Transactions on Antennas and Propagation*, 60(9), 4110–4118.

- Kerarti, D. Z., Marouf, F. Z., & Meriah, S. M. (2012). New tapered slot vivaldi antenna for UWB applications. In *Proceedings of the International Conference on Microelectronics*, (pp. 5–8).
- Kumar, R., & Suraj, P. (2017). Antipodal Vivaldi antenna for UWB communication with metamaterial. In *Proceedings of the International Conference on Inventive Systems and Control*, (pp. 1–5).
- Kushwaha, N., & Kumar, R. (2014). Study of different shape electromagnetic band gap (EBG) structures for single and dual band applications. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, 13(1), 16–30.
- Li, Z., Tan, L., Kang, X., Su, J., Guo, Q., Yang, Y. L., & Wang, J. (2016). A novel wideband end-fire conformal antenna array mounted on a dielectric cone. *Applied Computational Electromagnetics Society Journal*, 31(8), 933–942.
- Natarajan, R., George, J. V., Kanagasabai, M., & Shrivastav, A. K. (2015). A compact antipodal Vivaldi antenna for UWB applications. *IEEE Antennas and Wireless Propagation Letters*, 14, 1557–1560.
- Pandey, G. K., Singh, H. S., Bharti, P. K., Pandey, A., & Meshram, M. K. (2014). High gain Vivaldi antenna for radar and microwave imaging applications. *International Journal of Signal Processing Systems*, 3(1), 35–39.
- Ren, L. S., Li, F., Zhao, J. J., Zhao, G., & Jiao, Y. C. (2010). A novel compact UWB antenna with dual band-notched characteristics. *Journal of Electromagnetic Waves and Applications*, 24(11–12), 1521–1529.
- Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1), Article 6086. <https://doi.org/10.1038/s41598-024-56706-x>.

- Sarkar, D., & Srivastava, K. V. (2013). SRR-loaded antipodal Vivaldi antenna for UWB applications with tunable notch function. In *21st International Symposium on Electromagnetic Theory*, (pp. 466–469).
- Singha, R., & Vakula, D. (2015). Corrugated antipodal Vivaldi antenna using spiral shape negative index metamaterial for ultra-wideband application. In *Proceedings of IEEE International Conference on Signal Processing and Communication Engineering Systems*, (Vol. 5880, pp. 179–183).
- Saito, T. (2015). The precision-recall plot is more informative than the ROC curve when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3), e0118432. <https://doi.org/10.1371/journal.pone.0118432>
- Tang, H. B., & Shui, X. Z. (2013). Design of antipodal Vivaldi antenna array. *Applied Mechanics and Materials*, 380–384, 3465–3468.
- Teni, G., Zhang, N., Qiu, J., & Zhang, P. (2013). Research on a novel miniaturized antipodal Vivaldi antenna with improved radiation. *IEEE Antennas and Wireless Propagation Letters*, 12, 417–420.
- Venkatarayalu, N. V., & Gan, Y. B. (2010). Design of a tapered slot array antenna for UWB through-wall radar. In *Proceedings of IEEE International Symposium on Antennas and Propagation and CNC-USNC/URSI Radio Science Meeting - Leading the Wave*, (pp. 1–4).
- Vignesh, N., Sathish Kumar, G. A., & Brindha, R. (2014). Design and development of a tapered slot Vivaldi antenna for ultra-wideband application. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(5), 174–178.
- Wang, P., Zhang, H., Wen, G., & Sun, Y. (2012). Design of a modified 6-18 GHz balanced antipodal Vivaldi antenna. *Progress in Electromagnetic Research*, 25, 271–285.

- Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2022). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2), 187–212. <https://doi.org/10.1007/s40745-020-00253-5>
- Yang, Z., Jingjian, H., Weiwei, W., & Naichang, Y. (2017). An antipodal Vivaldi antenna with band-notched characteristics for ultra-wideband applications. *AEU - International Journal of Electronics and Communications*, 76, 152–157.
- Yimdjo Poffelie, L. A., Soh, P. J., Yan, S., & A. E. Vandenbosch, G. (2016). A high-fidelity all-textile UWB antenna with low back radiation for off-body WBAN applications. *IEEE Transactions on Antennas and Propagation*, 64(2), 757–760.
- Zeng, J. (2013). Compact electromagnetic band-gap structures (EBG) and its applications in antenna systems. In *Master's Thesis, University of Waterloo* (pp. 1–7).
- Zhang, J., Zhang, X., & Xiao, S. (2017). Antipodal Vivaldi antenna to detect UHF signals that leaked out of the joint of a transformer. *International Journal of Antennas and Propagation*, 2017, 1–13.
- Zhu, F., & Gao, S. (2013). Compact elliptically tapered slot antenna with nonuniform corrugations for ultra-wideband applications. *Radio Engineering*, 22(1), 276–280.