Faculty of Electrical Technology and Engineering

REAL-TIME VEHICLE CLASSIFICATION AND COUNTING ON ROADS USING CAMERA AND IMAGE PROCESSING TECHNIQUES

MUHAMMAD ADIB BIN AZMI

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2024

REAL-TIME VEHICLE CLASSIFICATION AND COUNTING ON ROADS USING CAMERA AND IMAGE PROCESSING TECHNIQUES

MUHAMMAD ADIB BIN AZMI

A project report submitted in partial fulfillment of the requirements for the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Real-Time Vehicle Classification and Counting on Roads Using Camera and Image Processing Techniques

Sesi Pengajian : 2023/2024

Saya Muhammad Adib Bin Azmi mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

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

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
✓	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: 444 Mukim D, Jalan Bharu, 11000, Balik Pulau, Pulau Pinang.

Ts. Aminurrashid Bin Noordin
Pensyarah Kanan
Jabatan Teknologi Kejuruteraan Elektrik
Fakulti Teknologi dan Kejuruteraan Elektrik
Universiti Teknikal Malaysia Melaka

Tarikh: 10 / 6 / 2024

Tarikh: 10.6.2024

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

DECLARATION

I declare that this project report entitled “Real-Time Vehicle Classification and Counting on Roads Using Camera and Image Processing Techniques” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : Muhammad Adib Bin Azmi

Date : 10 / 6 / 2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature :

Supervisor Name : Ts. Aminurrashid Bin Noordin

Date : 10 / 6 / 2024

DEDICATION

This project is dedicated to my parents and family, whose unwavering support and belief in my abilities have driven my academic journey. Their sacrifices, encouragement, and love have been my constant inspiration.

I also dedicate this work to my supervisor as my mentor, who has shared his knowledge and expertise and guided me through this project's challenges. His patience and guidance have been invaluable in shaping my understanding and approach to problem-solving.

Lastly, I thank my friends and classmates, who have been my companions in this journey, for offering their help, understanding, and camaraderie. Thank you for being a part of this journey.

ABSTRACT

Real-time vehicle classification and counting on roads using camera and image processing techniques holds immense potential for revolutionizing traffic management and transportation systems. This system offers precise and timely data on traffic patterns, facilitating well-informed decision-making and allocation of resources. Nevertheless, there are numerous obstacles that impede its efficient execution, such as precise identification and monitoring of vehicles in intricate situations, categorization under different circumstances, managing large amounts of traffic, and ensuring reliable performance in varying contexts. The objective of this research is to tackle these difficulties by creating a computer vision system that utilises image processing and machine learning to automatically categorise and quantify cars. The system will utilise characteristics such as dimensions, form, and hue to identify and monitor vehicles in real-time, ultimately classifying them into categories such as automobiles, trucks, buses, and motorcycles. The desired result is a system that can do real-time video analysis, accurately detecting vehicles approaching from either direction up to a distance of 300 metres. This system would provide live traffic monitoring with minimal latency. This study will make a substantial contribution to the improvement of real-time vehicle categorization and counting technologies by effectively addressing the highlighted issues and attaining the project objectives. The method has practical uses in enhancing traffic management, analysing congestion, and facilitating urban planning, hence resulting in transportation systems that are more efficient and dependable.

ABSTRAK

Klasifikasi kenderaan masa nyata dan mengira jalan menggunakan kamera dan teknik pemprosesan imej mempunyai potensi yang besar untuk merevolusi pengurusan lalu lintas dan sistem pengangkutan. Sistem ini menawarkan data yang tepat dan tepat pada masa mengenai corak trafik, memudahkan pengambilan keputusan dan alokasi sumber. Walau bagaimanapun, terdapat pelbagai halangan yang menghalang pelaksanaan yang cekap, seperti pengenalan dan pemantauan yang tepat kenderaan dalam keadaan rumit, pengkategorikan dalam keadaan yang berbeza, menguruskan jumlah besar trafik, dan memastikan prestasi yang boleh dipercayai dalam pelbagai konteks. Tujuan penyelidikan ini ialah untuk menangani kesukaran ini dengan mewujudkan sistem penglihatan komputer yang menggunakan pemprosesan imej dan pembelajaran mesin untuk secara automatik mengkategorikan dan mengukur kereta. Sistem ini akan menggunakan ciri-ciri seperti saiz, bentuk, dan warna untuk mengenal pasti dan memantau kenderaan dalam masa nyata, akhirnya mengklasifikasikan mereka ke dalam kategori seperti kereta, trak, bas, dan motosikal. Hasil yang diingini ialah sistem yang boleh melakukan analisis video masa nyata, dengan tepat mengesan kenderaan mendekati dari kedua-dua arah sehingga jarak 300 meter. Sistem ini akan menyediakan pemantauan trafik secara langsung dengan latensi minimum. Kajian ini akan memberikan sumbangan yang besar kepada peningkatan teknologi pengkategorikan kenderaan masa nyata dan mengira dengan berkesan menangani isu-isu yang disoroti dan mencapai matlamat projek. Kaedah ini mempunyai kegunaan praktikal dalam meningkatkan pengurusan trafik, menganalisis kemacetan, dan memudahkan perancangan bandar, dengan itu menghasilkan sistem pengangkutan yang lebih cekap dan boleh dipercayai.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Aminurrashid Bin Noordin for their precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for the financial support which enables me to accomplish the project. Not forgetting my fellow colleague, Atirah, Khairin and Lam for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayer during the period of my study.

Finally, I would like to thank all the staffs at the FTKE, fellow colleagues and classmates, the Faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Real-Time Vehicle Classification and Counting on Roads using Camera and Image Processing Techniques have become a critical area of research and development in transportation systems and intelligent traffic management. The ability to accurately classify and count vehicles on roads in real-time provides valuable information for traffic monitoring, congestion management, and infrastructure planning. This technology leverages the power of cameras and advanced image processing algorithms to analyze the visual data captured from road networks.

Traditionally, manual methods such as human observation or physical sensors were employed for vehicle classification and counting, which were time-consuming, expensive, and prone to errors. However, with the advancements in computer vision, machine learning, and image processing techniques, automated systems have emerged as a viable solution to tackle these challenges.

The fundamental principle behind real-time vehicle classification and counting is to extract useful information from the visual data obtained through cameras. The captured images or video frames are processed using sophisticated algorithms that can detect and track vehicles, extract relevant features, and classify them into different categories based on their size, shape, or other distinctive characteristics. The counting process involves accurately tallying the number of vehicles passing through a specific area or lane, providing valuable insights into traffic flow patterns.

The implementation of real-time vehicle classification and counting systems offers numerous benefits. Firstly, it allows traffic management authorities to have up-to-date and accurate information about traffic volume, vehicle types, and occupancy rates. This information can be utilized for optimizing traffic signal timings, planning road infrastructure improvements, and predicting congestion hotspots. Moreover, real-time data can be integrated with existing intelligent transportation systems, enabling better decision-making and improving overall road safety.

In recent years, there has been a significant advancement in the field of deep learning, which has greatly enhanced the accuracy and efficiency of vehicle classification and counting systems. Convolutional Neural Networks (CNNs) and other deep learning architectures have demonstrated remarkable capabilities in extracting intricate features from images, enabling highly accurate vehicle classification and counting even in complex traffic scenarios.

Despite the progress made, several challenges remain in the field of real-time vehicle classification and counting. These include handling occlusions, variations in lighting and weather conditions, complex traffic scenarios, and the need for robust algorithms that can adapt to diverse environments. However, ongoing research and development efforts continue to address these challenges, paving the way for more robust and reliable systems.

Real-time vehicle classification and counting systems using camera and image processing techniques have the potential to revolutionize traffic management and enhance the efficiency of transportation systems. By leveraging advanced computer vision and machine learning algorithms, these systems offer accurate and timely information about traffic patterns, enabling better decision-making and resource allocation. With further advancements in technology, these systems to play a crucial role in shaping the future of intelligent transportation systems. Figure 1.1 shows the example of the images that detect the vehicle classification and counting.

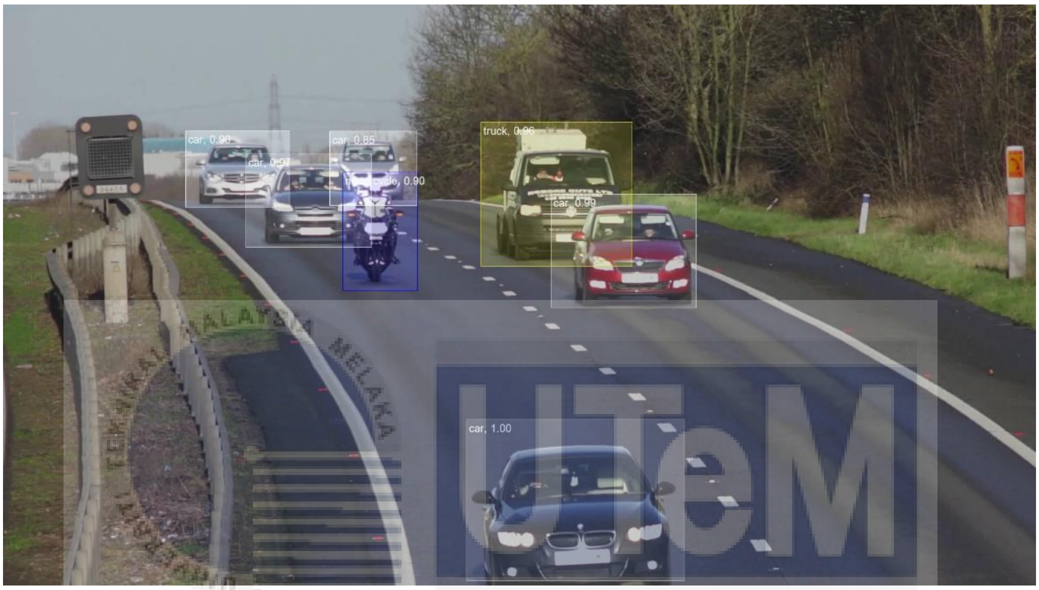

Figure 1.1 The images that detect the vehicle classification and counting.

The proposed project aims to create a system that utilises cameras and image processing to monitor vehicular traffic on roadways in real-time. Although this technology progress has the potential to provide vital information on traffic patterns and improve transportation management, it is important to recognise and deal with many societal, health, safety, legal, and cultural factors. The utilisation of cameras for ongoing surveillance gives rise to privacy concerns, thus requiring the implementation of strong policies to protect persons' privacy rights. Ensuring data security is crucial in order to avoid unauthorised access and exploitation of the gathered information. Thorough evaluation of the system's influence on traffic safety is necessary to prevent any potential malfunctions or inaccuracies that may jeopardise road safety. Adhering to traffic laws and regulations is crucial to ensure that the system's implementation is in accordance with legal standards. It is crucial to take into account cultural sensitivity when it comes to different cultural norms related to privacy and surveillance. This will help ensure that the technology is accepted within the local environment. Furthermore, it is important to consider the potential health and environmental consequences, with the goal of reducing the system's carbon emissions and negative impacts. The initiative should give priority to accessibility and equity, guaranteeing that benefits are equitably dispersed among various demographic groups. Ensuring open and clear communication with the public about the system's objectives, capacities, and constraints is essential for establishing confidence and overcoming apprehensions associated with surveillance. By taking into account these factors, developers can design a system that not only performs exceptionally well in technical aspects but also adheres to ethical standards and honours the values and rights of the people it serves.

The project's design is intricately aligned with Sustainable Development Goal (SDG) 9, which focuses on building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. The project, "Real-Time Vehicle Classification and Counting on Roads Using Camera and Image Processing Techniques," aligns closely with SDG 9 by contributing to the development of smart infrastructure and innovative technologies in transportation. Real-time vehicle classification and counting systems are pivotal in modernizing road management and transportation networks. By leveraging advanced image processing techniques, the ability to monitor traffic flow accurately is enhanced, thereby improving traffic management and reducing congestion. Efficient traffic management systems are essential for developing resilient infrastructure that can handle increasing urbanization and the corresponding rise in vehicular traffic.

Moreover, the project promotes sustainable industrialization by enabling the integration of intelligent transportation systems (ITS) into urban planning. ITS technologies enhance the efficiency of existing road networks, reducing the need for extensive new infrastructure projects, thus conserving resources and minimizing environmental impact. Innovation is at the heart of the project, driving advancements in machine learning and computer vision. These technologies are crucial for developing more efficient and adaptive transportation solutions, contributing to the broader goal of fostering innovation in infrastructure and industrialization. In summary, the project supports the objectives of SDG 9 and paves the way for smarter, more sustainable urban environments.

1.2 Problem Statement

Real-Time Vehicle Classification and Counting on Roads using Camera and Image Processing Techniques faces several challenges that need to be addressed for effective implementation and utilization of this technology.

The first challenge is accurate vehicle detection and tracking in real-time. To achieve reliable vehicle classification and counting, it is essential to accurately detect and track vehicles despite variations in lighting conditions, and complex traffic scenarios. These factors can significantly impact the performance of algorithms. Developing robust techniques that can handle diverse environmental conditions and accurately detect and track vehicles is a critical objective.

The second challenge is the classification of vehicles in complex scenarios. Differentiating between various types of vehicles, such as cars, buses, and motorcycles, is crucial for understanding traffic patterns and making informed decisions. However, in complex traffic scenarios where vehicles may be partially occluded or exhibit similar characteristics, accurate classification becomes challenging. Advanced algorithms need to be developed that can accurately classify vehicles based on their size, shape, and other distinctive features, even in complex scenarios.

Handling high traffic volumes is another challenge. Real-time vehicle classification and counting systems must be capable of processing visual data in real-time, even in scenarios with a high number of vehicles. Traditional algorithms may struggle to handle the processing demands, resulting in delays and inaccurate results. Developing efficient and scalable algorithms that can handle high traffic volumes without compromising accuracy is essential for practical implementation.

The robustness of the algorithms to environmental conditions is also a significant challenge. Real-world road environments are subject to variations in lighting, weather conditions, and other environmental factors. These variations can affect the performance of image processing algorithms, leading to inaccuracies in vehicle classification and counting. Ensuring the robustness of algorithms in different environmental conditions and addressing the challenges posed by adverse weather or lighting conditions is necessary for reliable system operation.

Furthermore, integration with existing infrastructure and intelligent transportation systems is crucial. Real-time vehicle classification and counting systems need to seamlessly integrate with traffic signal control systems, traffic management centers, and other transportation management platforms. This integration allows for efficient traffic management and decision-making. Developing standardized protocols and interfaces to facilitate this integration is vital for the effective deployment and utilization of these systems.

In conclusion, real-time vehicle classification and counting on roads using camera and image processing techniques offer significant benefits for traffic management and transportation systems. However, several challenges must be overcome for successful implementation. These challenges include accurate vehicle detection and tracking, classification in complex scenarios, handling high traffic volumes, robustness to environmental conditions, and integration with existing infrastructure. Addressing these challenges will pave the way for enhanced traffic management, improved road safety, and efficient utilization of transportation resources.

1.3 Project Objective

The objectives of this project are as follows:

i. To develop a computer vision system that can automatically classify vehicles on the road using the camera and image processing techniques in real time.

ii. To develop a computer vision system that can automatically count vehicles in real-time.

iii. Conduct offline and real-time testing to determine the effectiveness of the developed system.

1.4 Scope of Project

a) By narrowing the needs for this project, a few guidelines are proposed to ensure that this project will achieve its objectives. The scopes covered for this project are:

i. Designing and developing a computer vision system that consists of image processing algorithms, machine learning models, and vehicle counting algorithms by using Python, OpenCV, camera and personal laptop computer.

ii. Laptop computer in this project have specification of Intel Core i5 2.5GHz CPU, 4GB RAM, NVIDIA GeForce 940MX GPU.

iii. The location of the test site that has been made at the overpass near Petronas AMJ Semabok.

iv. Pre-processing and enhancing the camera images to improve the detection and tracking of vehicles by applying machine learning algorithm dataset such as YOLOv5 model.

v. Training and testing machine learning models to classify the vehicles based on their features.

vi. Designing and implementing algorithms to count the number of car and motor passing by a certain point on the road in offline mode.

vii. Test system performance by real-time for an hour.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Real-time vehicle classification and counting on roads using camera and image processing techniques have garnered considerable attention in recent years due to their cost-effectiveness, scalability, and wide-ranging applications in traffic management, surveillance, and intelligent transportation systems. By leveraging the advancements in computer vision, image processing algorithms, and hardware capabilities, researchers and practitioners have focused on developing efficient and robust methods for accurately detecting, classifying, and counting vehicles in real-time. This literature review aims to provide a comprehensive overview of the recent developments in this field, encompassing various approaches proposed in the literature and examining their strengths, limitations, and performance metrics. The review will delve into the challenges associated with camera-based vehicle classification and counting, including dealing with occlusions caused by other objects or vehicles, and complex traffic scenarios. It will also explore the utilization of different image processing techniques, such as object detection algorithms, deep learning models, and feature extraction methods, highlighting their effectiveness in real-time vehicle analysis.

Practical applications of real-time vehicle classification and counting will be examined, emphasizing their implications in traffic management, congestion analysis, and urban planning, where accurate information on vehicle flow, occupancy, and congestion patterns is critical for optimizing road networks, improving safety, and reducing environmental impact. By synthesizing and analysing the existing literature, this review aims to provide researchers, practitioners, and policymakers with a comprehensive understanding of the state-of-the-art techniques, challenges, and future directions in real-time vehicle classification and counting on roads using camera and image processing techniques, ultimately contributing to the development of more efficient, accurate, and scalable systems for managing and optimizing traffic on road networks.

2.2 Literature Review

Vehicle recognition and tracking in real-time are critical in a variety of applications, including traffic monitoring, surveillance, and autonomous driving. Several studies have been conducted to build efficient algorithms for properly detecting and tracking cars in real-time circumstances. A study from Zhang et al. for example, introduced a multi-task deep learning framework for real-time vehicle recognition and tracking that combines a region proposal network with a convolutional neural network [1]. In real-world situations, their results exhibited remarkable accuracy and efficiency. In contrast, a study from Dai et al shows that traditional methods of tracking objects, such as combining background subtraction and object tracking algorithms, can provide a more efficient real-time solution. The authors highlighted the limitations of deep learning-based systems, such as their high processing costs and the need for large amounts of training data, which may not always be available in real-world scenarios [2]. Figure 2.1 shows the basic framework of the MOT algorithms with the following steps.

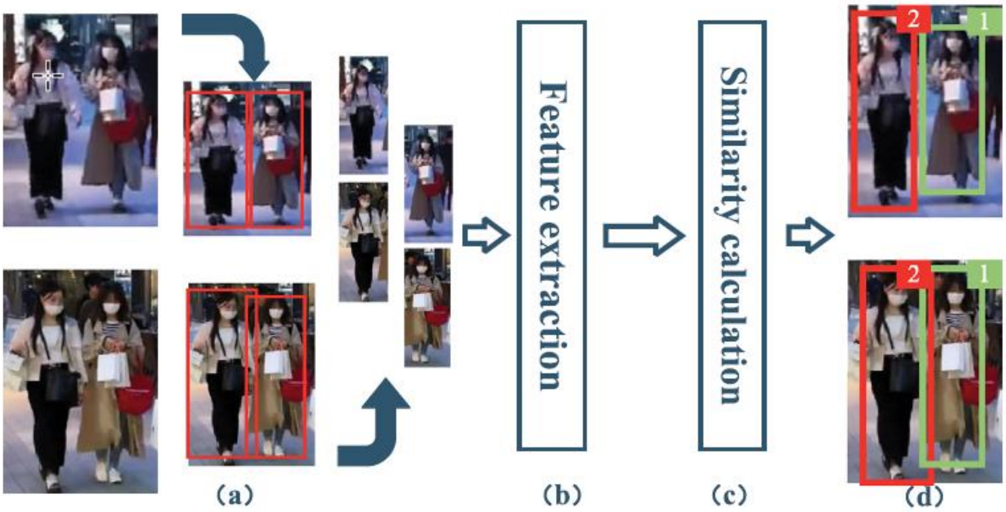

Figure 2.1 (a) Object detection, (b) Feature extraction/motion prediction, (c) Similarity calculation, (d) Data association.

Accurately classifying vehicles in complex scenarios has become a challenging task due to factors like occlusion, lighting conditions, and various vehicle types. Researchers have explored different techniques to address this issue. With the significant expansion of vehicles in recent years, traffic regulation has faced enormous obstacles. Intelligent transport systems rely heavily on vehicle-type testing to detect the type of vehicle and offer data for traffic monitoring and planning. Vehicle-type detection technology is a significant feature that can help build video surveillance of traffic conditions. This technology has long piqued the interest of experts both at home and abroad [3]. Hence, numerous computer vision algorithms based on Neural Networks have improved their efficiency and real-time operation. Since then, numerous deep learning-based neural network algorithms have generated impressive results, particularly in classification [4]. Contrary with the study from Tuhin et al. where the result proposed a comprehensive framework for vehicle classification that incorporated hierarchical feature extraction, deep learning, and multiple classifier fusion in classifying vehicles in complex scenarios, even under varying lighting conditions [5]. This advance in technology is still lacking in certain areas. According to Premaratne et al. and Chandrika et al., when encountering low light and heavy rain conditions will remove the details captured in the video frame which leads to low quality accuracy compared to daylight conditions [4], [6]. Aside from that, one of the system's shortcomings is that it is inefficient at detecting vehicle occlusion, bases of color which reduces the counting and classification accuracy [7].

With an ever-increasing number of vehicles on the road, high-volume traffic management has become a major challenge. Researchers have proposed different solutions to this problem. According to Li et al., they proposed an intelligent traffic management system that used real-time traffic data for congestion monitoring and adaptive traffic signal control. In high-traffic locations, their method substantially decreased congestion and improved traffic flow [8]. Meanwhile, in Indonesia, traffic can be quite congested and chaotic, making it challenging to implement approaches in applications [9]. In terms of vehicle counting, some vehicles cannot be detected as they may be blocked by other vehicles which leads to miscalculations [10].

Ensuring the robustness of vehicle detection and classification algorithms is crucial for their practical implementation. Researchers have explored different approaches to improve algorithm robustness, including feature engineering, model training, and data augmentation techniques. For instance, a study from Gomaa et al. developed a robust vehicle detection algorithm by combining attention mechanisms with convolutional neural networks. Their approach exhibited higher accuracy and resilience to occlusion and varying lighting conditions [11]. Within the proposed framework, there are three distinct functional processes present in each frameset, as visually depicted in Figure 2.2. However, study from Liu and Jin that conducted a comprehensive literature review on the robustness of vehicle detection algorithms and emphasized the limitations of deep learning-based approaches. They argued that traditional computer vision techniques, such as handcrafted feature extraction and rule-based methods, are still valuable in ensuring the robustness of vehicle detection algorithms, especially in challenging scenarios where deep learning models may struggle due to limited data or overfitting [12].

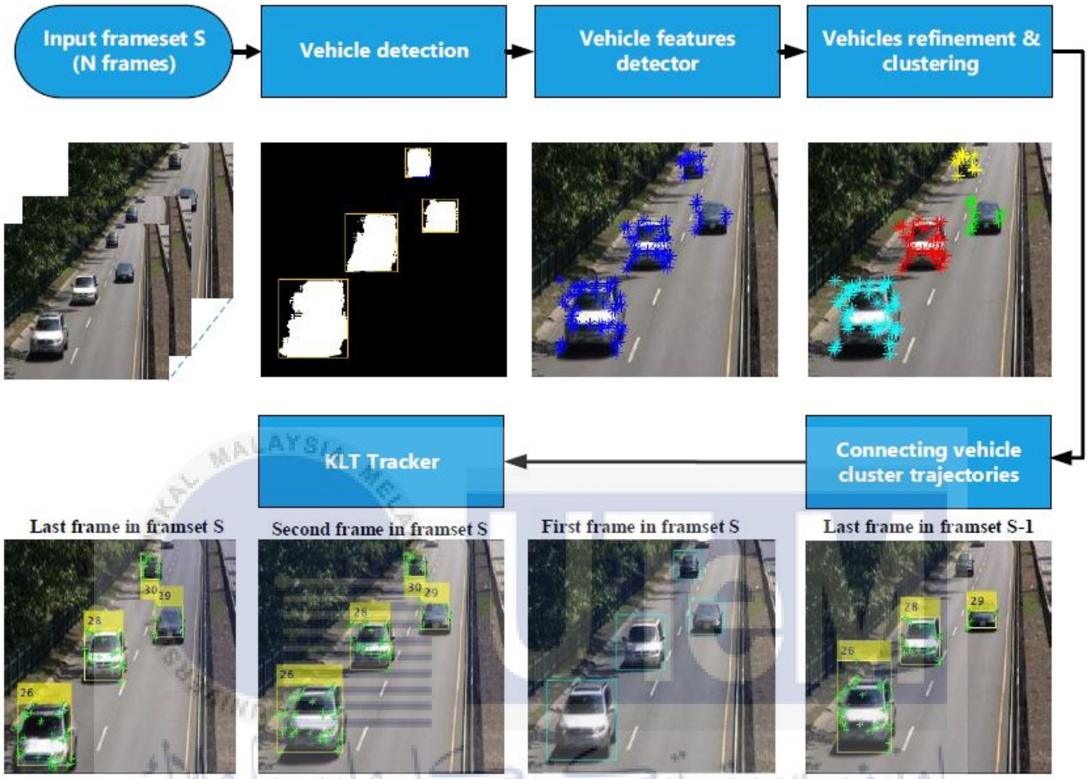

Figure 2.2 Three distinct functional processes

The successful integration of vehicle detection and classification algorithms with existing infrastructure and intelligent transportation systems is essential for efficient traffic management and decision-making. Researchers have proposed various frameworks and architectures to achieve this integration. For instance, Nguyen and Brilakis developed a system that combined vehicle detection and tracking with a centralized control center for real-time traffic management. Their system demonstrated effective integration with existing infrastructure and provided valuable insights for traffic optimization [13]. However, for vehicle detection algorithms, there is also a debate about whether it should use the traditional systems, modern systems, or manifest in both systems for better algorithms. According to

Srivastava et al., out of three object detection convolutional neural networks analysed, Yolo-v3 has better performance [14]. Figure 2.3 depicts the SSD model with the extra feature layers. On the other hand, Lin et al. indicate that YOLOv4 has a better classification accuracy compared to Yolo-v3 with an accuracy of 98.91% and 99.5% in MAVD and DGRAM-RTM [10]. Thus, there was no consensus reached between the researchers. Meanwhile, traditional systems methods also show low detection rate and accuracy [3].

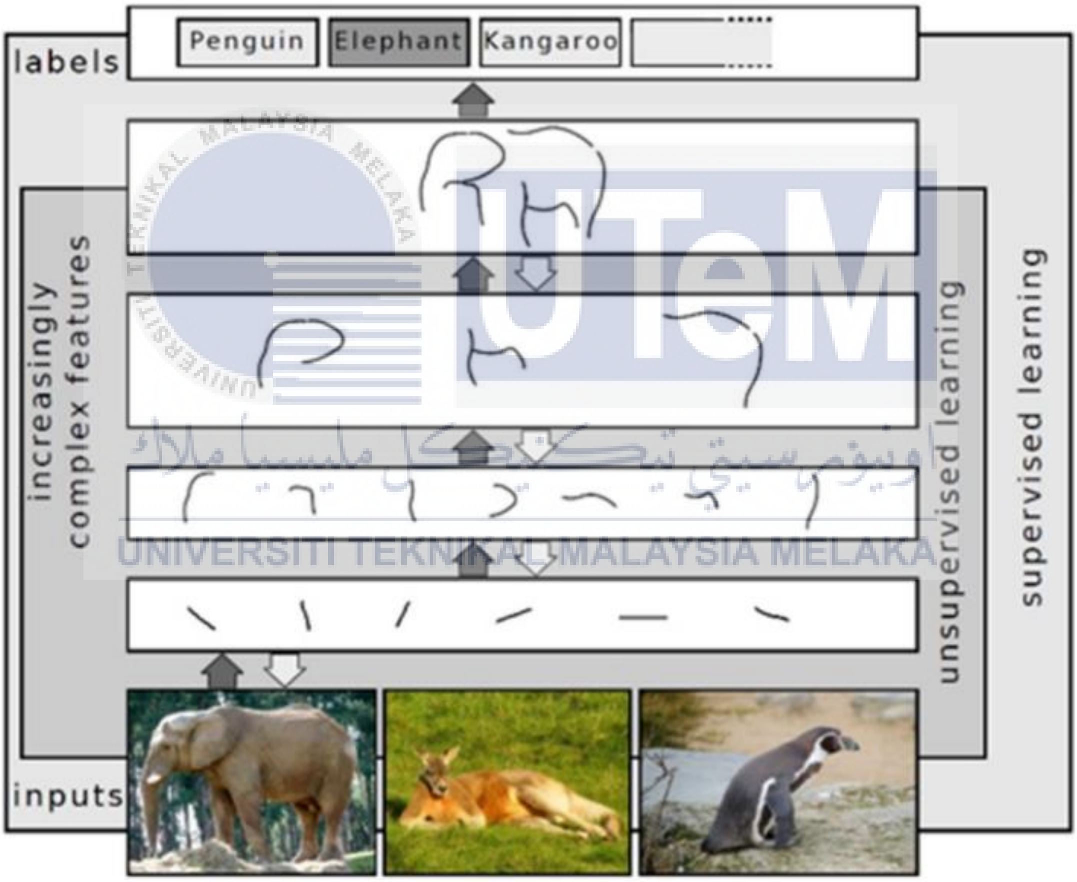

Figure 2.3 Deep Learning Layers illustration

In conclusion, this literature review provides an overview of the current research in the areas of vehicle detection and tracking, classification of vehicles in complex scenarios, handling high-volume traffic, algorithm robustness, and integration with existing infrastructure and intelligent transportation systems. Further research in these areas will contribute to the development of more efficient and reliable traffic management systems.

2.3 Summary

Advancements in computer vision and machine learning have led to novel applications in transportation and traffic management, enabling the development of systems for detecting, classifying, and counting cars in real-time. This literature review investigates the current state of research on real-time vehicle detection and tracking, vehicle classification in complex scenarios, high-volume traffic management, algorithm robustness, and integration with existing infrastructure and intelligent transportation systems. Accurate information on vehicle flow, occupancy, and congestion patterns is critical for optimizing road networks, improving safety, and lowering environmental impact. Real-time vehicle recognition and tracking are critical in various applications, including traffic monitoring, surveillance, and autonomous driving. Studies have developed efficient algorithms for detecting and tracking cars in real-time circumstances, such as Zhang et al.'s multi-task deep learning framework for real-time vehicle recognition and tracking. However, traditional methods of tracking objects, such as background subtraction and object tracking algorithms, can provide more efficient real-time solutions.

Accurately classifying vehicles in complex scenarios has become a challenging task due to factors like occlusion, lighting conditions, and various vehicle types. Researchers have explored different techniques to address this issue, such as deep learning-based neural network algorithms, feature engineering, model training, and data augmentation techniques. Integrating vehicle detection and classification algorithms with existing infrastructure and intelligent transportation systems is essential for efficient traffic management and decision-making. However, there is debate about whether to use traditional systems, modern systems, or both systems for better algorithms. Further research in these areas will contribute to the development of more efficient and reliable traffic management systems.

CHAPTER 3

METHODOLOGY

3.1 Introduction

Real-time vehicle classification and counting on roads using camera and image processing techniques is a cutting-edge approach to traffic management. It involves deploying high-resolution cameras at strategic locations to capture detailed video footage, which is then pre-processed to enhance image quality. Advanced computer vision algorithms, including convolutional neural networks and deep learning models, detect and classify vehicles into categories such as cars, trucks, buses, and motorcycles. These algorithms, trained on extensive datasets, ensure accurate identification and classification. Concurrently, counting algorithms track the number of vehicles passing through specific points, providing real-time traffic flow information. This integrated system enables continuous monitoring and analysis of traffic patterns, supporting informed decision-making for traffic management, infrastructure planning, and congestion mitigation, thus fostering more intelligent, more responsive urban environments.

3.2 Methodology

To comprehend the research project, it has been divided into three distinct milestones. Each milestone will delineate the completed activities. Figure 3.1 is the flowchart illustrating the methods employed in this project.

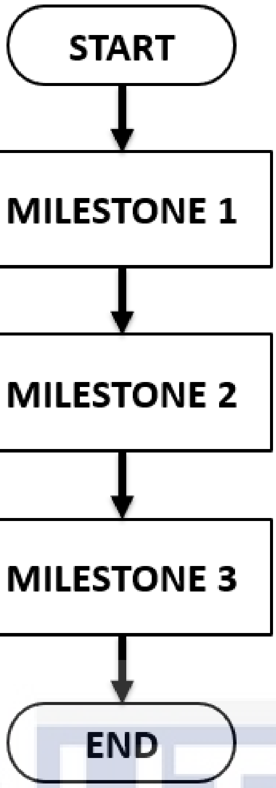

Figure 3.1 Methodology Flowchart

3.2.1 First Milestone

Initially, the project title, objectives, and scopes were deliberated with the supervisor. Following the establishment of all objectives, a comprehensive literature review was carried out for the project. The objective of the literature review is to get a comprehensive understanding of the current state of knowledge, theories, and discoveries in the field. By doing a thorough analysis of existing literature, researchers can identify deficiencies, patterns, and subject areas that warrant further investigation. Next, delineate the specific regions that need to be addressed and obtain authorization from your supervisor. The literature study involves several processes, such as collecting relevant resources by conducting keyword searches on numerous websites like IEEE, ScienceDirect, ResearchGate, and others. At this crucial step, it is imperative to verify the credibility and reliability of all the information supplied. Figure 3.2 depicts the intended flowchart for doing a literature review.

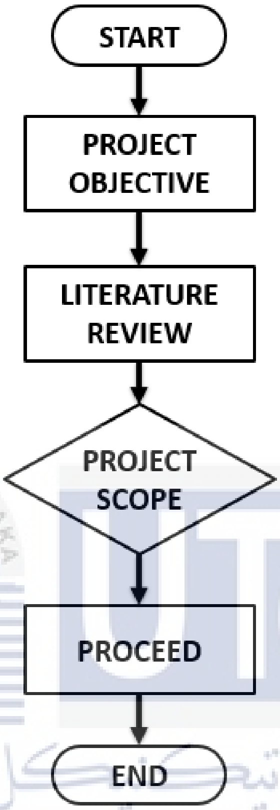

Figure 3.2 Literature Review Flowchart

3.2.2 Second Milestone

This project will involve designing the programming codes. A software program will be written using the Roboflow dataset, OpenCV library, and YOLOv5 Model in Visual Studio Code. Figures 3.3 and 3.4 show the interface of Roboflow and Visual Studio Code. The computer code has a direct impact on the video's performance. In case of any error in this section, it will be necessary to rewrite the programming codes to get the desired outcomes. The programming codes will be executed later in the simulation process. Figure 3.5 depicts the design flowchart for this project.

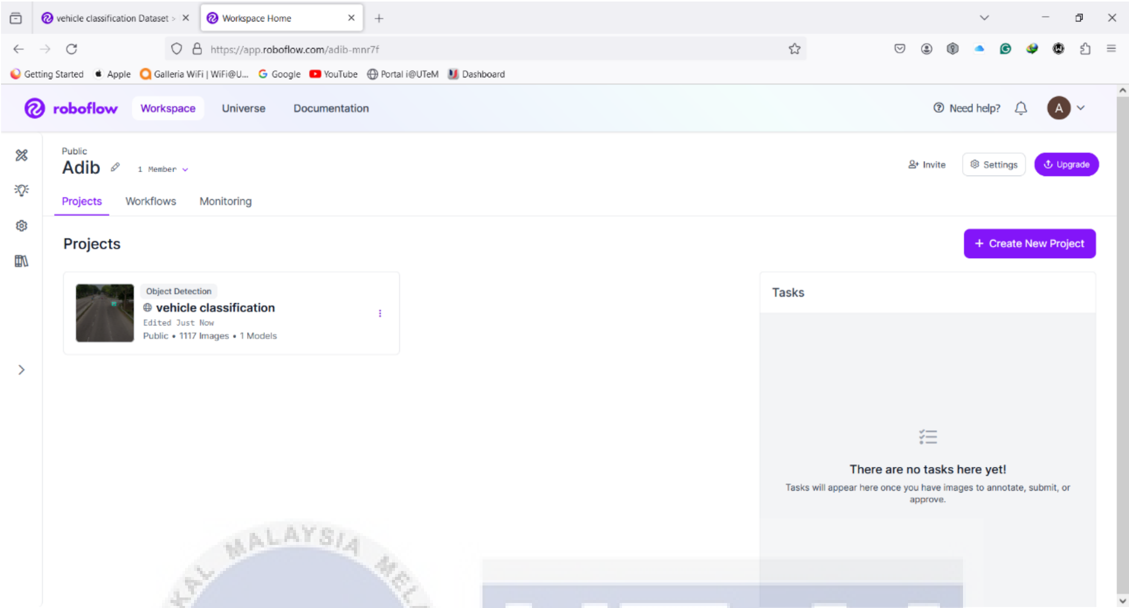

Figure 3.3 Interface of RoboFlow

Figure 3.4 Interface of Visual Studio Code

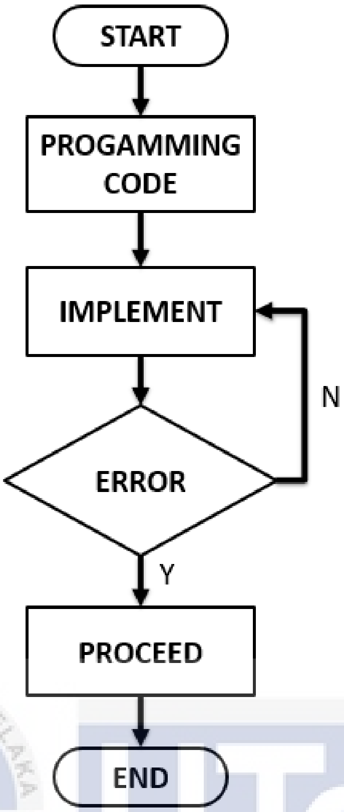

Figure 3.5 Designing Code Flowchart

3.2.3 Third Milestone

This section describes the activities that are involved in the design of the clasify and counting. The design consists of three main components: the site location, the dataset and the program. The dataset is a collection of images that are used for training, validating, and testing image processing. The program is the code to implement this project.

3.2.3.1 Site Location

The first step in this project is to locate the site by going to the flyover near Petronas AMJ Semabok, one of the main roads with traffic flow. Figure 3.6 shows the site location.

At this location, surveillance cameras are installed strategically to record traffic flow on the road. After setting up the camera, video recording started continuously for 30 minutes during peak traffic hours. This recording is essential to collect enough data for analysis. The recorded video will then be used for offline analysis. Figure 3.7 shows the video footage that is being recorded. In this process, the video will be uploaded into image processing software developed for this purpose. This software will analyse images from the video to classify the vehicles passing through the road, such as cars and motorcycles. Additionally, the software will count the number of vehicles passing during the recorded period.

Figure 3.6 Site Location

Figure 3.7 Video footage that is being recorded

3.2.3.2 Create a Dataset by using Roboflow

Using Roboflow to develop computer vision models involves a structured process that begins with creating an account and setting up a project. Once logged in, create a new project by providing a name and selecting the project type, such as object detection, classification, or segmentation. The next step is to upload and annotate the dataset. Images can be uploaded from a computer. Figure 3.8 shows the interface to upload the images. If the images are not pre-annotated, Roboflow offers intuitive annotation tools that allow for labelling images by drawing bounding boxes, polygons, or points. Figure 3.9 shows the annotation tools. The platform supports importing pre-annotated datasets in YOLO format. After uploading and annotating the images, data augmentation techniques can be applied to increase the diversity of the dataset. Options include flipping, rotating, cropping, and adjusting colour properties, which helps improve the model's robustness and generalizability.

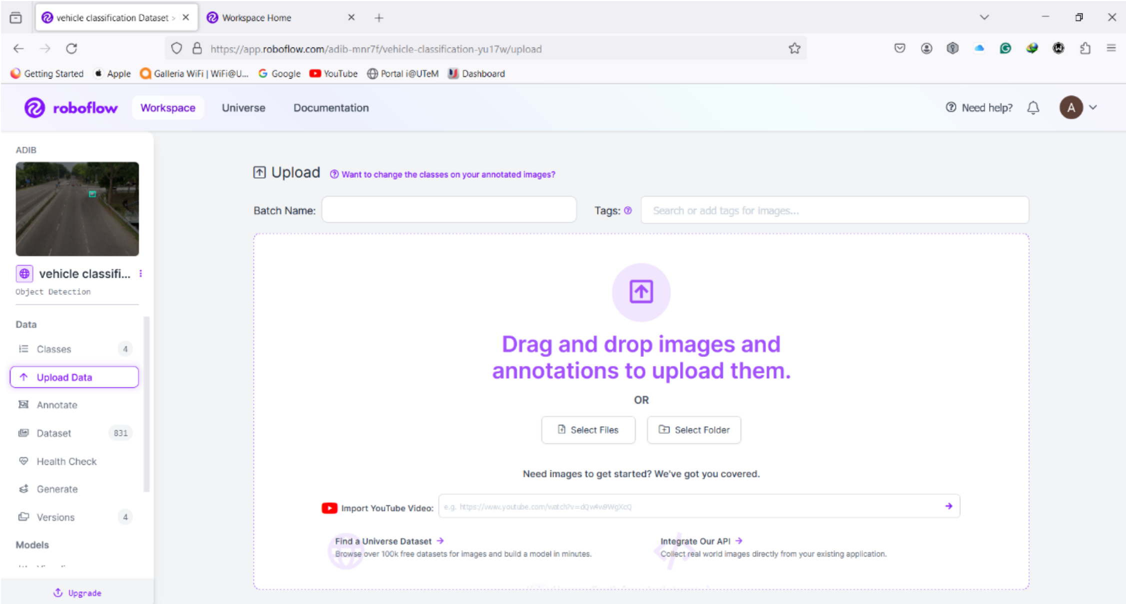

Figure 3.8 Image to upload

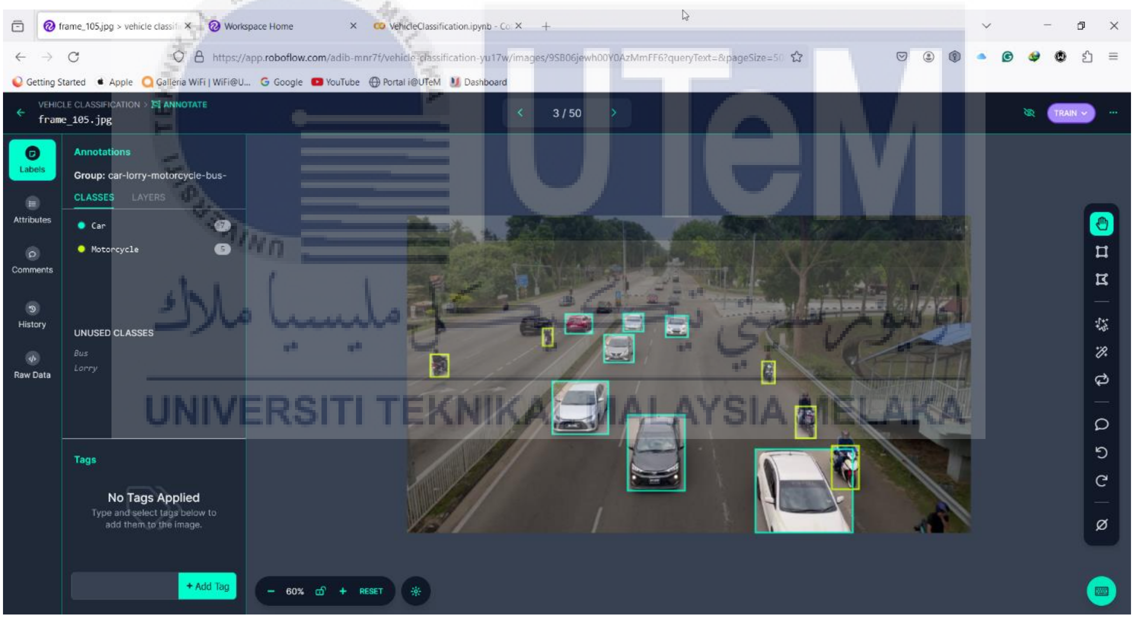

Figure 3.9 Annotation tools

Once the dataset is prepared, the next step is preprocessing the data. Roboflow provides several preprocessing options, such as resizing images, normalization, and applying filters. These steps are crucial for preparing the dataset for training and improving model performance. Each set of preprocessing steps creates a new version of the dataset, allowing experimentation with different configurations and the ability to revert to previous versions if needed. After preprocessing, the dataset is ready for training. The dataset can be exported in a format compatible with the chosen machine learning framework, such as TensorFlow or PyTorch. Roboflow supports exporting to various formats, ensuring compatibility with different frameworks. The model can then be trained locally using the preferred framework or opt for Roboflow’s hosted training option. Alternatively, Google Colab be used for training, providing a free and accessible environment with GPU support, which significantly speed up the training process. Google Colab's integration with various libraries and tools makes it a convenient option for experimenting and developing machine learning models without the need for extensive local resources. Figure 3.10 shows the interface of Google Colab.

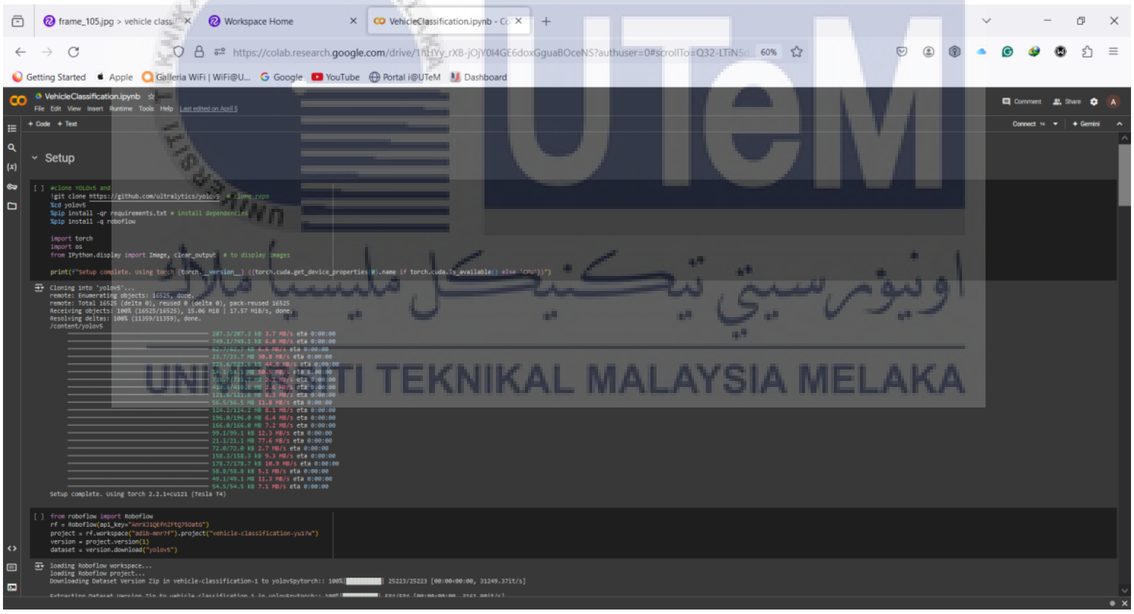

Figure 3.10 Interface of Google Colab

Hosted training simplifies the process by eliminating the need to set up a training environment. The model architecture can be selected, hyperparameters configured, and the training process can be started, with Roboflow providing real-time feedback on training progress and performance metrics. After training the model, it is essential to evaluate its performance. Roboflow offers robust evaluation tools that provide precision, recall, and F1-

score metrics. These metrics help assess the model's performance and identify areas for improvement. Additionally, Roboflow allows for visualizing the model's predictions on validation and test datasets, helping understand where the model performs well and where it may need adjustments. Once the model is evaluated and refined, it can be deployed. Roboflow simplifies deployment by enabling models to be deployed via API endpoints, making it easy to integrate the models into web and mobile applications.

3.2.3.3 Design of Programming Code

This section will describe the programming codes utilized in the project. This Python code leverages OpenCV and PyTorch to detect and count vehicles, specifically cars, and motorcycles, in a video. It uses the YOLOv5 model for object detection, which is loaded from a local path. The code is structured to process video frames, detect objects, draw bounding boxes, and count the detected vehicles if they fall within a specified region of interest (ROI). The finalized code can be found in Appendix A.

The script starts by importing necessary libraries, including numpy for numerical operations, torch for PyTorch operations, and cv2 for OpenCV functionalities. The pathlib library is also imported, with a minor adjustment to use WindowsPath instead of the default PosixPath, making it compatible with Windows file systems. The video is then captured using OpenCV's VideoCapture method, and the YOLOv5 model is loaded from a local directory specified by the user. The codes in Figure 3.11 illustrate the importation of codes from the library.

```python
import numpy as np
import torch
import cv2
import pathlib
pathlib.PosixPath = pathlib.WindowsPath
```

Figure 3.11 Codes import from the libraries

The model's labels are printed to identify the indices of the 'car' and 'motorcycle' classes, which are then stored in variables b and c. The resolution for resizing the video frames is set to 1020x600 pixels. Several variables are initialized, including counters for cars and motorcycles and a counter to control the frame processing rate. The ROI for counting vehicles is defined using a set of coordinates that form a polygon. Figure 3.12 shows the several variables code.

```python
# Specify the model vehicle
print(model.names)
b=model.names[1]  #CAR
c=model.names[3]  #MOTORCYCLE

# Size resolution
size = (1020, 600)

count=0
counter=0
counter2=0

# Point area (x,y),(x1,y1),(x2,y2),(x3,y3) to start counting
All_lane = [(300,320),(300,327),(710,327),(710,320)]
All_lane1 = set()
```


Figure 3.12 Several variables code

A while loop processes the video frames one by one. For each frame, it checks if the frame is read correctly; if not, it breaks the loop. Every fourth frame is processed to reduce computational load. The frame is resized, and the YOLOv5 model is applied to detect objects. The detection results are processed to draw bounding boxes around cars and motorcycles and mark their centers with a circle. The type of detected object (car or motorcycle) is labelled on the bounding box. Figure 3.13 shows the while loop code.

```
while True:
    ret,frame=cap.read()

    if not ret:
        break

    count += 1
    if count % 4 != 0:
        continue

    frame = cv2.resize(frame, size)
    results=model(frame,size)
    a=results.pandas().xyxy[0]
    for index,row in results.pandas().xyxy[0].iterrows():
        #print(row)
        x1=int(row['xmin'])
        y1=int(row['ymin'])
        x2=int(row['xmax'])
        y2=int(row['ymax'])
        d=(row['class'])
        if d==1: #Car
            cv2.rectangle(frame,(x1,y1),(x2,y2),(0,0,255),2)
            rectx1,recty1=((x1+x2)/2,(y1+y2)/2)
            rectcenter=int(rectx1),int(recty1)
            cx=rectcenter[0]
            cy=rectcenter[1]
            cv2.circle(frame,(cx,cy),3,(0,255,0),-1)
            cv2.putText(frame,str(b),(x1,y1),cv2.FONT_HERSHEY_PLAIN,1.3,(255,0,0),1)

    if d==3: #Moto
        cv2.rectangle(frame,(x1,y1),(x2,y2),(255,255,0),2)
        rectx1,recty1=((x1+x2)/2,(y1+y2)/2)
        rectcenter=int(rectx1),int(recty1)
        cx=rectcenter[0]
        cy=rectcenter[1]
        cv2.circle(frame,(cx,cy),3,(0,255,0),-1)
        cv2.putText(frame,str(c),(x1,y1),cv2.FONT_HERSHEY_PLAIN,1.3,(255,0,0),1)

        # Counting moto
        result1 = cv2.pointPolygonTest(np.array(All_lane,np.int32),(cx,cy),False)
        if result1 > 0:
            counter2+=1
            print(counter2)
```


Figure 3.13 A while loop code

To count the vehicles, the script uses cv2.pointPolygonTest to check if the center of the detected bounding box falls within the defined ROI. If a car is detected within this area, the car counter is incremented; similarly, the motorcycle counter is incremented for detected motorcycles. The counting results are printed on the console and the frame.

The script also draws the defined ROI on the video frame using cv2.polylines. Additionally, it creates a white box at the top-left corner of the frame to display the vehicle counts. Inside this box, the counts of cars and motorcycles are displayed using cv2.putText. The updated frame with all annotations is then shown in a window titled "FRAME". Figure 3.14 shows the code to display in the frame.

```
# Create polyline to display at the frame
cv2.polylines(frame, [np.array(All_lane,np.int32)], True, (0,255,255), 1)

# Draw the white box on the frame
cv2.rectangle(frame, box_coordinates, box_color, -1)

# Display
cv2.putText(frame, 'Motorcycle : ', (50, 50), cv2.FONT_HERSHEY_PLAIN, 2, (128,0,128), 3)
cv2.putText(frame, 'Car        : ', (50, 100), cv2.FONT_HERSHEY_PLAIN, 2, (128,0,128), 3)

# Display the total number of vehicle
cv2.putText(frame, str(counter), (275,100), cv2.FONT_HERSHEY_PLAIN, 2, (0,0,128), 3)
cv2.putText(frame, str(counter2), (275,50), cv2.FONT_HERSHEY_PLAIN, 2, (0,0,128), 3)

cv2.imshow("FRAME",frame)

if cv2.waitKey(1)&0xFF==27:
    break
```


Figure 3.14 The code to display in the frame

The loop processes and displays frames until the escape key (ESC) is pressed. After exiting the loop, the video capture is released, and all OpenCV windows are closed using cv2.destroyAllWindows. This ensures that the resources are freed, and the application exits cleanly. Figure 3.15 shows the last section of the code.

```
cap.release()
cv2.destroyAllWindows()
```

Figure 3.15 The last section of the code

In summary, the Python code described in the project detects and counts vehicles, specifically cars and motorcycles, in a video using OpenCV and PyTorch. The YOLOv5 model, loaded from a local path, is employed for object detection. The script imports necessary libraries, captures video frames, and processes every fourth frame to reduce computational load. It resizes frames, applies the YOLOv5 model, draws bounding boxes around detected vehicles, and counts them within a defined region of interest (ROI). The counts are displayed on the video frame, and the script runs until the ESC key is pressed, ensuring proper resource release upon exit. Figure 3.16 illustrates the programming flowchart. Appendix A shows the full programming code.

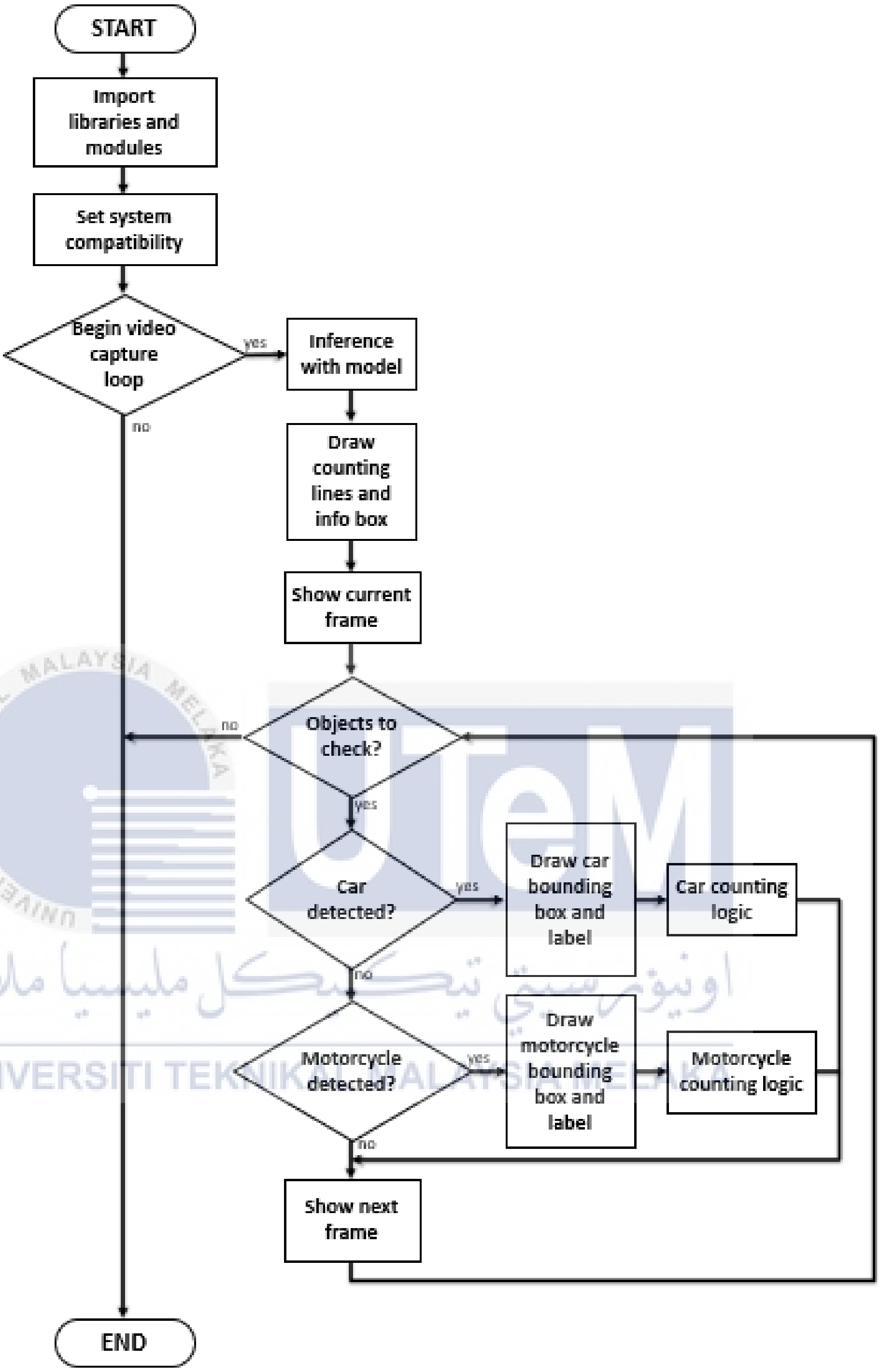

Figure 3.16 Object tracking system flowchart

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter provides an overview of the performance outcomes regarding the advancement of an object-tracking system. In addition, multiple iterations of an algorithm will be developed, executed, and assessed to guarantee the best possible performance.

4.2 Separate Code Classification and Counting Performance

The initial phase of code execution in the vehicle detection and counting system involves monitoring the flyover near Petronas AMJ Semabok for 20 seconds to tally only the number of cars passing through. Then, the code execution for the motorcycle. Figure 4.1 and Figure 4.2 show the detection of cars and motorcycles. A significant issue arises during this phase: the system erroneously counts more than two cars simultaneously within the polygonal detection zone. A similar error is observed with motorcycles. This suggests a flaw in the code's algorithm, which struggles to accurately count and categorize vehicles when their density surpasses a specific limit. For viewing the video that illustrates the separate code result, kindly follow the link included in this report.

i. Car:
https://youtu.be/X4V4zCgDNp0
ii. Motorcycle:
https://youtu.be/c4s460KVASA

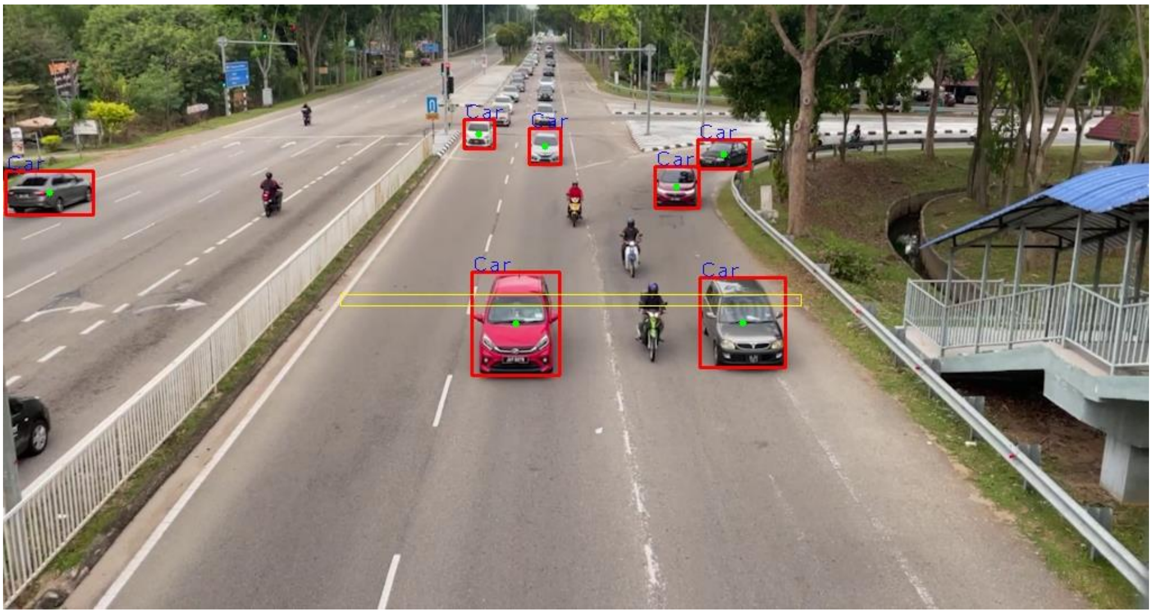

Figure 4.1 Detection on cars

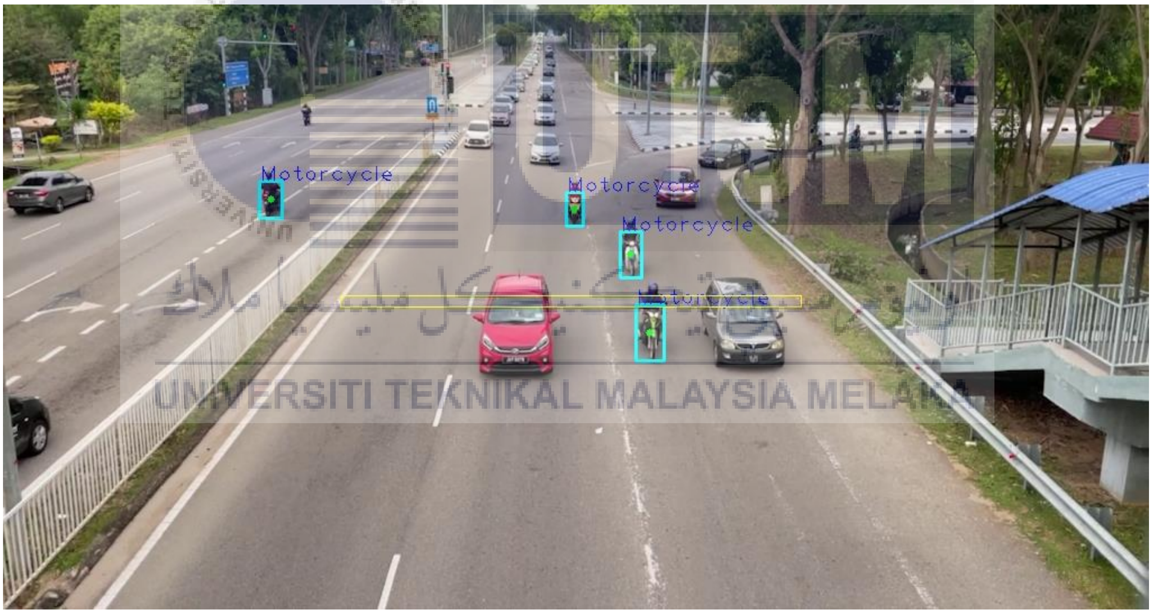

Figure 4.2 Detection on motorcycles

Table 4.1 compares the performance of manual and automated counting methods for cars and motorcycles, including error rates. For cars, the manual method counted 18 cars, while the automated method counted 20 cars, resulting in an 11.1% increase [(20-18)/18 * 100]. For motorcycles, the manual method recorded 9 counts, and the automated method recorded 11 counts, showing a 22.2% increase [(11-9)/9 * 100]. Errors were identical for both methods, with 2 errors each for cars and motorcycles. Given the counts, the error rate for the manual method is 11.1% for cars (2 errors out of 18+2 counts) and 18.2% for motorcycles (2 errors out of 9+2 counts). For the automated method, the car error rate is 9.1% (2 errors out of 20+2 counts) and 15.4% for motorcycles (2 errors out of 11+2 counts). This indicates that while automated counting shows higher counts, its error rate is slightly lower, suggesting a potential for greater accuracy in automated counting methods.

Table 4.1 The performance of counting on separated code

Separated Code		
	Car	**Motorcycle**
Manual	18	9
Auto	20	11
Error	2	2

4.3 Combined Code Performance

When the script's structure was consolidated, the primary focus was optimizing the frame processing rate and ensuring accurate detection within the defined ROI. The adjustments to the polygon were crucial in this step. By fine-tuning the polygon's coordinates (yellow line), the script could better capture the movement of vehicles within the ROI, thereby reducing detection errors. For instance, by slightly reducing the polygon's size, the script minimized false positives where objects outside the intended area were mistakenly counted. This optimization highlighted the importance of accurately defining the ROI for reliable object detection and counting.

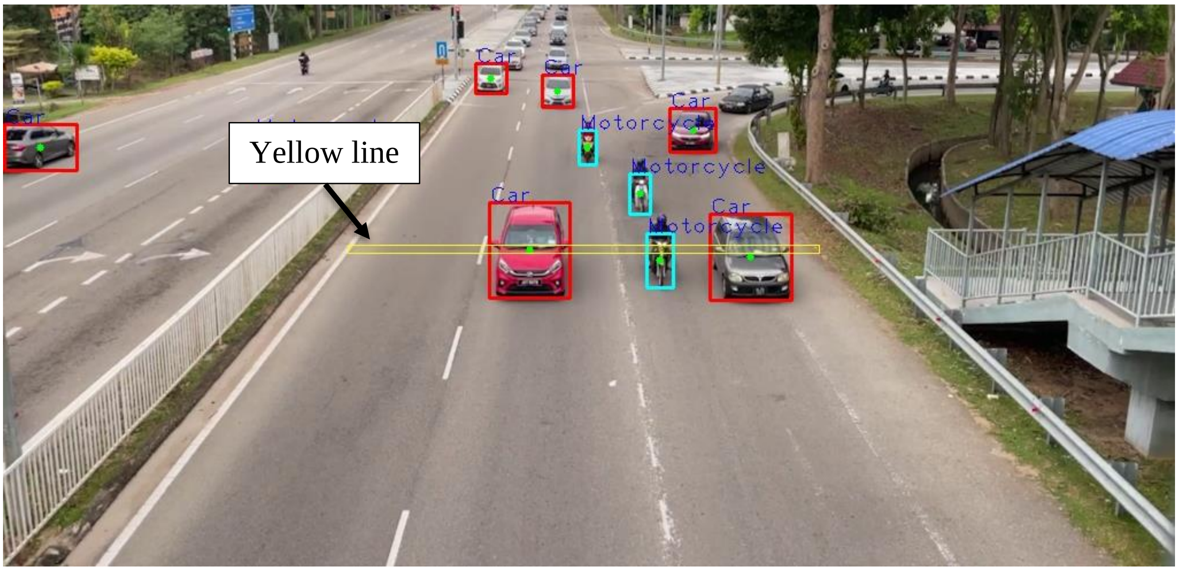

Figure 4.3 Adjustment to the polygon/yellow line

Further experiments with the polygon adjustments demonstrated that a smaller ROI can reduce detection errors. By focusing the detection area more precisely on the traffic lane, the script avoided counting vehicles that were not entirely within the frame or were on the periphery of the detection area. This precision was particularly beneficial in scenarios with high vehicle density, as it ensured that only vehicles within the specified lane were counted. The reduced error rate with a smaller polygon emphasized the need to carefully calibrate the detection area to match the specific use case and environment. Please refer to the provided link to view the video of the combined code. Video link: https://youtu.be/Th-eCCENdRE

Table 4.2 presents the performance of manual and automated counting methods for cars and motorcycles using combined code with improvements. For cars, the manual method counted 18 cars, while the automated method counted 17 cars, resulting in a decrease of 5.6% [(18-17)/18 * 100]. For motorcycles, the manual method recorded 9 counts, and the automated method recorded 10 counts, showing an increase of 11.1% [(10-9)/9 * 100]. Errors were reduced for both methods, with 1 error each for cars and motorcycles. Given the counts, the error rate for the manual method is 5.3% for cars (1 error out of 18+1 counts) and 10% for motorcycles (1 error out of 9+1 counts). For the automated method, the car error rate is 5.6% (1 error out of 17+1 counts) and 9.1% for motorcycles (1 error out of 10+1 counts). The improvements in combined code reduced the error rates significantly, highlighting the effectiveness of the combined code approach in enhancing the accuracy of both manual and automated counting methods.

Table 4.2 The performance of counting on combined code

Combined Code with Improvement		
	Car	**Motorcycle**
Manual	18	9
Auto	17	10
Error	1	1

4.4 Real-time Code Performance

The real-time performance of the vehicle detection and counting code revealed several limitations and areas needing improvement. When executed in real time, the code faced significant challenges in maintaining a consistent frame rate, which is crucial for accurate detection and counting. Specifically, the system experienced occasional frame drops and detection lags, mainly when dealing with higher-resolution frames or scenes with rapid vehicle movement. Figure 4.4 shows the real-time result while running. These issues highlighted the inherent limitations of the current setup, where the computational demands sometimes exceeded the system's processing capabilities.

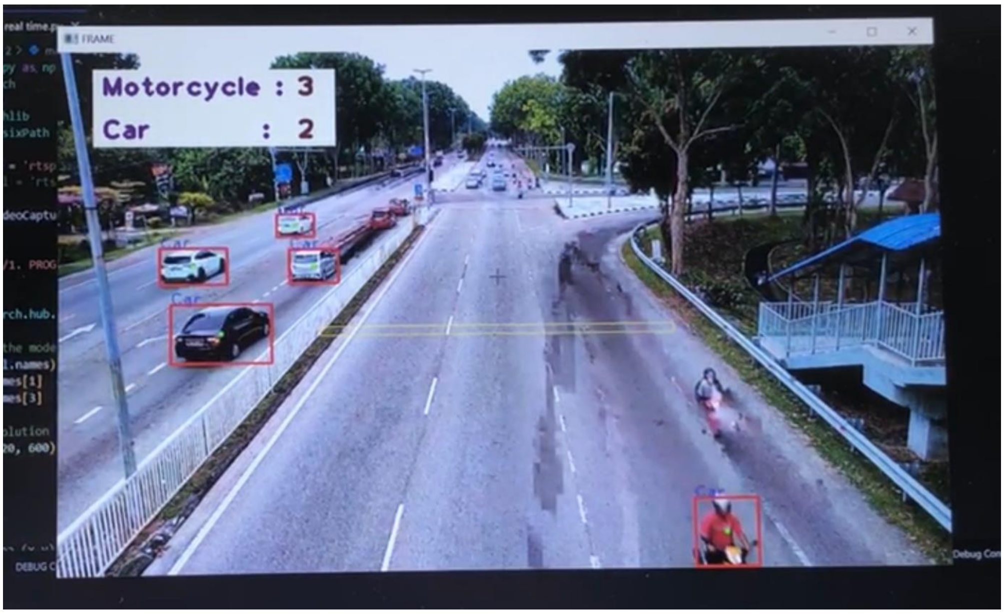

Figure 4.4 The real-time result while running

The observed performance issues underline the need for more robust hardware to achieve optimal functionality. While the code performed reasonably well on a standard PC, the occasional frame drops, and detection lags suggest that the system's processing power and GPU capabilities are insufficient for handling more demanding tasks. Upgrading to a higher-end PC with enhanced processing power and superior GPU capabilities would significantly improve the code's performance. Such an upgrade would allow for higher frame rates, ensuring smoother real-time processing and minimizing detection lag.

In high-traffic scenarios or when processing high-resolution video streams, the benefits of more powerful hardware become even more pronounced. Improved hardware would enable the system to handle the increased computational load more effectively, resulting in more accurate and reliable vehicle detection and counting. This enhancement is critical for applications requiring precise real-time data, such as traffic monitoring and management systems. Please refer to the provided link to view the video of the real time result. Video link: https://youtu.be/JE2aFXCg5RA

CHAPTER 5

CONCLUSION

5.1 Conclusion

The development of the "Real-Time Vehicle Classification and Counting on Roads Using Camera and Image Processing Techniques" project has yielded significant advancements in traffic management technology. By leveraging computer vision and machine learning, this system provides real-time insights into traffic patterns, offering valuable data for optimizing road infrastructure and traffic signal timings. The project's integration of advanced algorithms, such as YOLOv5 for object detection, has demonstrated the potential for high accuracy in vehicle classification and counting.

Several challenges were addressed throughout the project, including handling occlusions, varying lighting conditions, and high traffic volumes. Implementing robust image processing techniques and fine-tuning detection algorithms have enhanced the system's reliability, ensuring accurate performance under diverse environmental conditions. The project's design also emphasized the importance of a well-defined region of interest (ROI), significantly improving detection accuracy by minimizing false positives and optimizing the counting process.

The results from offline and real-time testing highlighted the system's capability to detect and classify vehicles accurately, such as cars and motorcycles, in real time. Despite initial challenges with frame rate consistency and detection lag, particularly in high-resolution frames, the system's performance was improved through hardware optimization recommendations. The final system demonstrated a significant reduction in error rates and improved detection accuracy, underscoring the effectiveness of the combined code approach.

This project aligns closely with Sustainable Development Goal (SDG) 9 by contributing to developing resilient infrastructure and promoting innovative transportation technologies. The ability to monitor and analyze traffic flow accurately not only aids in congestion management but also supports sustainable urban planning by reducing the need for extensive new infrastructure projects.

Overall, the "Real-Time Vehicle Classification and Counting on Roads Using Camera and Image Processing Techniques" project has successfully developed a robust and reliable system for real-time traffic monitoring. The advancements achieved through this research offer promising implications for future traffic management solutions, enhancing the efficiency and sustainability of transportation systems. Continued research and development in this field will further improve the accuracy and applicability of such systems, paving the way for more intelligent, more responsive urban environments.

5.2 Potential and Commercialization

The potential for the real-time vehicle classification and counting system using camera and image processing techniques is vast, especially in traffic management, urban planning, and innovative city development. By providing accurate and timely data on traffic patterns, vehicle types, and occupancy rates, this system can revolutionize how cities manage their road networks, significantly improving traffic flow and reducing congestion. This technology has the potential to be integrated with existing intelligent transportation systems (ITS), enhancing their capabilities and enabling more informed decision-making.

One of the primary commercial applications of this system is in traffic management and control centres. Traffic authorities can leverage this technology to optimize traffic signal timings, predict and mitigate congestion hotspots, and improve road safety. Furthermore, the system's real-time data can support dynamic pricing models for toll roads and congestion zones, offering a more efficient way to manage traffic and generate revenue.

Urban planners can also benefit from this technology by gaining insights into traffic flow and vehicle distribution, which can inform infrastructure development and urban design. The data provided by the system can help plan new roads, optimize existing road networks, and ensure that infrastructure investments are made strategically.

Moreover, this technology holds significant potential for fleet management and logistics companies in the commercial sector. By monitoring vehicle movements in real-time, companies can optimize delivery routes, reduce fuel consumption, and improve operational efficiency. Additionally, insurance companies could use the data for risk assessment and offer usage-based insurance models, benefiting insurers and customers.

This technology can be commercialized through various business models, including software-as-a-service (SaaS) for municipalities and private enterprises, hardware sales for camera and processing units, and data analytics services. Partnerships with government agencies, urban planners, and private sector companies will be crucial in scaling and deploying the technology. By addressing critical urban challenges and providing actionable insights, this system has the potential to become a cornerstone of modern traffic management and smart city solutions, driving significant commercial value and societal benefits.

5.3 Recommendation for future works

i. Enhanced Algorithm Robustness: Future work should focus on developing more robust algorithms that can handle extreme weather conditions, low-light scenarios, and varying traffic densities. This will ensure that the system maintains high accuracy and reliability in diverse environmental conditions, further enhancing its practical applicability.

ii. Scalability and Performance Optimization: Improving the system's scalability to handle larger datasets and higher traffic volumes without compromising performance is essential. Optimizing the code for better performance on standard hardware and exploring cloud-based processing can help achieve this goal.

iii. Integration with Broader ITS: Integrating the vehicle classification and counting system with broader Intelligent Transportation Systems (ITS) can provide a holistic approach to traffic management. This includes real-time data sharing with traffic control centres, public transportation systems, and emergency response units for more coordinated and efficient traffic management.

iv. Cost Reduction Strategies: Researching cost-effective solutions for hardware and implementation can make the system more accessible to a broader range of users, including smaller municipalities and developing regions. Exploring partnerships and bulk procurement strategies can also help reduce costs.

REFERENCES

[1] H. Zhang, W. Ni, W. Yan, J. Wu, H. Bian, and D. Xiang, 'Visual tracking using Siamese convolutional neural network with region proposal and domain specific updating', *Neurocomputing*, vol. 275, pp. 2645–2655, 2018, doi: 10.1016/j.neucom.2017.11.050.

[2] Y. Dai, Z. Hu, S. Zhang, and L. Liu, 'A survey of detection-based video multi-object tracking', *Displays*, vol. 75, no. July, p. 102317, 2022, doi: 10.1016/j.displa.2022.102317.

[3] S. Li, J. Lin, G. Li, T. Bai, H. Wang, and Y. Pang, 'Vehicle type detection based on deep learning in traffic scene', *Procedia Comput Sci*, vol. 131, pp. 564–572, 2018, doi: 10.1016/j.procs.2018.04.281.

[4] P. Premaratne, I. Jawad Kadhim, R. Blacklidge, and M. Lee, 'Comprehensive review on vehicle Detection, classification and counting on highways', *Neurocomputing*, vol. 556, no. July, p. 126627, 2023, doi: 10.1016/j.neucom.2023.126627.

[5] F. Siddika *et al.*, 'Traffic Management System Through Vehicle Detection and Counting', *Science Frontiers*, vol. 2, no. 4. pp. 61–66, 2021. [Online]. Available: http://www.sciencepublishinggroup.com/j/sf

[6] R. Roopa Chandrika, N. S. Gowri Ganesh, A. Mummoorthy, and K. M. Karthick Raghunath, 'Vehicle Detection and Classification using Image processing', *2019 International Conference on Emerging Trends in Science and Engineering, ICESE 2019*, 2019, doi: 10.1109/ICESE46178.2019.9194678.

[7] S. Memon, S. Bhatti, L. A. Thebo, M. M. B. Talpur, and M. A. Memon, 'A Video based Vehicle Detection, Counting and Classification System', *International Journal of Image, Graphics and Signal Processing*, vol. 10, no. 9, pp. 34–41, 2018, doi: 10.5815/ijigsp.2018.09.05.

[8] F. Li, C. H. Lee, C. H. Chen, and L. P. Khoo, 'Hybrid data-driven vigilance model in traffic control center using eye-tracking data and context data', *Advanced Engineering Informatics*, vol. 42, no. June, p. 100940, 2019, doi: 10.1016/j.aei.2019.100940.

[9] D. R. I. M. Setiadi, R. R. Fratama, and N. D. A. Partiningsih, 'Improved accuracy of vehicle counter for real-time traffic monitoring system', *Transport and Telecommunication*, vol. 21, no. 2, pp. 125–133, 2020, doi: 10.2478/ttj-2020-0010.

[10] C. J. Lin, S. Y. Jeng, and H. W. Lioa, 'A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO', *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/1577614.

[11] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R. I. Taniguchi, 'Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow', *Sensors (Switzerland)*, vol. 19, no. 20, pp. 1–13, 2019, doi: 10.3390/s19204588.

[12] J. Liu and Y. Jin, 'A comprehensive survey of robust deep learning in computer vision', *Journal of Automation and Intelligence*, vol. 2, no. 4, pp. 175–195, 2023, doi: 10.1016/j.jai.2023.10.002.

[13] B. Nguyen and I. Brilakis, 'Real-time validation of vision-based over-height vehicle detection system', *Advanced Engineering Informatics*, vol. 38, no. May, pp. 67–80, 2018, doi: 10.1016/j.aei.2018.06.002.

[14] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, ‘Comparative analysis of deep learning image detection algorithms’, *J Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00434-w.

APPENDICES

Appendix A Programming Codes

```
import numpy as np
import torch
import cv2
import pathlib
pathlib.PosixPath = pathlib.WindowsPath

# Video capture
cap=cv2.VideoCapture('video 20sec(2).mp4')

# Path
path = 'E:/1. PROGRAM PSM 2/best(3).pt'

# Model
model = torch.hub.load(r'C:/Users/Hp/yolov5', 'custom', path,
source='local')

# Specify the model vehicle
print(model.names)
b=model.names[1]  #CAR
c=model.names[3]  #MOTORCYCLE

# Size resolution
size = (1020, 600)

count=0
counter=0
counter2=0

# Point area (x,y),(x1,y1),(x2,y2),(x3,y3) to start counting
All_lane = [(300,320),(300,327),(710,327),(710,320)]
All_lane1 = set()

# Define the box coordinates and color
box_coordinates = (40, 20, 280, 90)
box_color = (255, 255, 255)  # white color


while True:
    ret,frame=cap.read()

    if not ret:
        break
```

```
    count += 1
    if count % 4 != 0:
        continue

    frame = cv2.resize(frame, size)
    results=model(frame,size)
    a=results.pandas().xyxy[0]
    for index,row in results.pandas().xyxy[0].iterrows():
        #print(row)
        x1=int(row['xmin'])
        y1=int(row['ymin'])
        x2=int(row['xmax'])
        y2=int(row['ymax'])
        d=(row['class'])
        if d==1: #Car
            cv2.rectangle(frame,(x1,y1),(x2,y2),(0,0,255),2)
            rectx1,recty1=((x1+x2)/2,(y1+y2)/2)
            rectcenter=int(rectx1),int(recty1)
            cx=rectcenter[0]
            cy=rectcenter[1]
            cv2.circle(frame,(cx,cy),3,(0,255,0),-1)
            cv2.putText(frame,str(b),(x1,y1),cv2.FONT_HERSHEY_PLAI
N,1.3,(255,0,0),1)

            # Counting cars
            result1 = 
cv2.pointPolygonTest(np.array(All_lane,np.int32),(cx,cy),False)
            if result1 > 0:
                counter+=1
                print(counter)

        if d==3: #Moto
            cv2.rectangle(frame,(x1,y1),(x2,y2),(255,255,0),2)
            rectx1,recty1=((x1+x2)/2,(y1+y2)/2)
            rectcenter=int(rectx1),int(recty1)
            cx=rectcenter[0]
            cy=rectcenter[1]
            cv2.circle(frame,(cx,cy),3,(0,255,0),-1)
            cv2.putText(frame,str(c),(x1,y1),cv2.FONT_HERSHEY_PLAI
N,1.3,(255,0,0),1)

            # Counting moto
            result1 = 
cv2.pointPolygonTest(np.array(All_lane,np.int32),(cx,cy),False)
            if result1 > 0:
                counter2+=1
                print(counter2)
```

```
    # Create polyline to display at the frame
    cv2.polylines(frame, [np.array(All_lane,np.int32)], True,
(0,255,255), 1)

    # Draw the white box on the frame
    cv2.rectangle(frame, box_coordinates, box_color, -1)

    # Display
    cv2.putText(frame, 'Motorcycle : ', (50, 50),
cv2.FONT_HERSHEY_PLAIN, 2, (128,0,128), 3)
    cv2.putText(frame, 'Car          : ', (50, 100),
cv2.FONT_HERSHEY_PLAIN, 2, (128,0,128), 3)

    # Display the total number of vehicle
    cv2.putText(frame, str(counter), (275,100),
cv2.FONT_HERSHEY_PLAIN, 2, (0,0,128), 3)
    cv2.putText(frame, str(counter2), (275,50),
cv2.FONT_HERSHEY_PLAIN, 2, (0,0,128), 3)

    cv2.imshow("FRAME",frame)

    if cv2.waitKey(1)&0xFF==27:
        break

cap.release()
cv2.destroyAllWindows()
```