Intelligent Transportation System for Traffic prediction

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Bolineni Chooshnita(2010030487)

Telugu Rohtih(2010030201)

Perla Nada Asish(2010030552)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
K L DEEMED TO BE UNIVERSITY
AZIZNAGAR, MOINABAD , HYDERABAD-500 075

MARCH 2023

BONAFIDE CERTIFICATE

This is to certify that the project titled **Intelligent Transportation System for Traffic prediction** is a bonafide record of the work done by

Bolineni Chooshnita(2010030487)

Telugu Rohtih(2010030201)

Perla Nada Asish(2010030552)

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** of the **K L DEEMED TO BE UNIVERSITY, AZIZNAGAR, MOINABAD , HYDERABAD-500 075**, during the year 2022-2023.

Dr. Giridhar Urkude	**Dr. Arpita Gupta**
Project Guide	Head of the Department

Project Viva-voce held on __________________________

Internal Examiner	**External Examiner**

ABSTRACT

Congestion in traffic is a major problem these days. Despite the fact that it appears to pervade all over, urban cities are the ones most influenced by it. And it's ever increasing nature makes it imperative to know the road traffic density in real time for better signal control and effective traffic management. There can be different causes of congestion in traffic like insufficient capacity, unrestrained demand, large Red Light delays etc. While insufficient capacity and unrestrained demand are somewhere interrelated, the delay of respective light is hard coded and not dependent on traffic. Therefore, the need for simulating and optimizing traffic control to better accommodate this increasing demand arises.

In recent years, image processing and surveillance systems have been widely used in traffic management for traveller's information, ramp metering and updates in real time. The traffic density estimation can also be achieved using Image Processing. This project presents the method to use live images feed from the cameras at traffic junctions for real time traffic density calculation using image processing. It also focuses on the algorithm for switching the traffic lights according to vehicle density on road, thereby aiming at reducing the traffic congestion on roads which will help lower the number of accidents. In turn it will provide safe transit to people and reduce fuel consumption and waiting time. It will also provide significant data which will help in future road planning and analysis.

In further stages multiple traffic lights can be synchronized with each other with an aim of even less traffic congestion and free flow of traffic. The vehicles are detected by the system through images instead of using electronic sensors embedded in the pavement. A camera will be placed alongside the traffic light. It will capture image sequences. Image processing is a better technique to control the state change of the traffic light. It shows that it can decrease the traffic congestion and avoids the time being wasted by a green light on an empty road. It is also more reliable in estimating vehicle presence because it uses actual traffic images. It visualizes the practicality, so it functions much better than those systems that rely on the detection of the vehicles' metal content.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Dr. Giridhar Urkude, our project guide, for helping us and guiding us in the course of this project .

Dr. Arpita Gupta, the Head of the Department, Department of DEPARTMENT NAME.

Our internal reviewers, **Dr. Sumit Hazra** , **Dr K Srinivas** , **Mrs.P sree lakshmi** for their insight and advice provided during the review sessions.

We would also like to thank our individual parents and friends for their constant support.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background of the Project

The project's background addresses the significant and growing issue of traffic congestion, particularly in urban areas. It emphasizes the continuous increase in traffic congestion due to several contributing factors, such as insufficient road capacity, unrestrained demand for transportation services, and the inefficiency of fixed traffic light control systems. The document points out that existing traffic management systems often have predetermined traffic light timings, which do not respond to real-time traffic conditions, leading to wasted time and increased fuel consumption.

1.1.1 Subsection Title

The document identifies various causes of traffic congestion, including insufficient road capacity (roads not being able to handle the volume of vehicles), unrestrained demand (high demand for transportation services), and lengthy delays at red lights. These factors contribute to the overall problem.

As metioned in the article by G.S.; Sakhare. [1] and the article by Khekare.

Sub Sub Section Title

It mentions that the delays caused by traffic lights are currently hard-coded and not responsive to real-time traffic conditions. This lack of adaptability in traffic signal control contributes to congestion.

1.2 Problem Statement

Traffic congestion in urban areas is a growing issue due to unrestrained demand, insufficient road capacity, and inefficient traffic light control. To address this, a novel traffic management solution uses live image feeds from cameras at traffic junctions for real-time traffic density calculation. This algorithm reduces congestion, improves safety, and optimizes traffic flow. By leveraging image processing technology, this project provides a practical and efficient alternative to conventional traffic management systems, paving the way for data-driven decision-making in future road planning and analysis.

1.3 Objectives

To develop a system that utilizes image processing and surveillance cameras to calculate real-time traffic density and optimize traffic control .[1] To overcome the limitations of previous systems by using surveillance cameras on four-way junctions, requiring no constant maintenance and being less prone to failure .[2] To reduce fuel wastage by effectively managing traffic congestion and reducing traffic jams . To utilize the OpenCV library for computer vision and the ThingSpeak cloud platform for data storage and analysis .[2] To create a low-cost and efficient traffic management system that can be implemented in urban cities [3].

1.4 Scope of the Project

The scope of the project includes the development and implementation of a traffic management system using image processing and surveillance cameras to calculate real-time traffic density and optimize traffic control. The system aims to overcome the limitations of previous systems by utilizing surveillance cameras on four-way junctions, requiring minimal maintenance and being less prone to failure . The project focuses on improving traffic management in urban cities, reducing congestion, and potentially saving fuel by efficiently managing traffic flow .which is easily accessible from the surveillance cameras present at each traffic junction.

Chapter 2

Literature Review

2.1 Intelligent transportation system

Surveillance systems are being developed to address vehicular congestion. These systems utilize cameras at junctions and data transmission over a mobile Ad-hoc network. Image analysis and foreground/background modeling are crucial elements of surveillance. Experiments show potential for efficiency and real-time execution in traffic systems, enhancing the overall traffic management.

refer to a Figure 2.1

SURVEY	AUTHOR	YEAR	EXPLANATION
SURVEY 1: Yolo9000: Better, faster, stronger	J. Redmon and A. Farhadi	2020	▪ Deep Convolutional Network ▪ Colour Point ▪ Pixel Count
SURVEY 2: A computer vision based vehicle detection and counting system	Nilakorn Seenouvong, Ukrit Watchareerueta	2019	Reference Line Model with Gaussian Mixer Grey Scale Counting
Survey 3: "You only look once: Unified, real-time object detection	J. Redmon, S. Divvala	2018	Majority Pixel Count Colour and Hidden Markov Model

Figure 2.1: Figure Title

2.2 Table

The dataset used in this project consists of traffic data collected from surveillance cameras at various junctions . The dataset includes images captured by the cameras, which are used for image processing and analysis. These images contain information about the vehicles present at each junction, including their positions, sizes, and movements. To refer to the table use the same 2.1

Dataset Name	Description
TrafficData1	Contains traffic data collected from surveillance cameras at various junctions
TrafficData2	Includes historical traffic data for analysis and comparison

Table 2.1: Data Set Table

2.3 Overview of related works

Khekare and Sakhare [4] proposed the development of VANETs (Vehicular Ad Hoc Networks) as a framework for smart cities to transmit information about traffic conditions and aid drivers in making smart decisions to prevent vehicular congestion.

Badura and Lieskovsky [5] presented a model for intelligent traffic systems that incorporate surveillance cameras on junctions and utilize image analysis and data delivery systems for efficient data transmission over a mobile ad-hoc network.

Salama, Saleh, and Eassa [7] designed an integrated intelligent system for managing and controlling traffic lights using photoelectric sensors. The system employs an algorithm based on the relative weight of each road to control traffic lights and can be programmed for emergency scenarios.

2.4 Advantages and Limitations of existing systems

Intelligent transportation systems offer advantages such as real-time monitoring, enabling authorities to gather accurate traffic conditions for informed decisions. However, constraints include high service costs, limited coverage, dependence on infrastructure, and lack of integration. These limitations need to be acknowledged and addressed to fully realize the benefits of data-driven decision-making in managing traffic flow.

Chapter 3

Proposed System

3.1 System Requirements

Hardware:- A laptop with a CPU featuring an i5 processor, at least 8 GB of RAM, and a 500 GB SSD .
Software:- An open-source computer vision library used for image processing and analysis, Anaconda graphical user interface

3.1.1 Image Analysis

The proposed traffic management system aims to utilize a combination of technologies and algorithms to optimize traffic flow and reduce congestion.The proposed system integrates these technologies and techniques to optimize traffic signal timings, dynamically adjust traffic flow, and improve overall traffic management efficiency.

3.2 Design of the System

To design the proposed traffic management system, a structured approach consisting of several key steps can be followed. Initially, it's crucial to identify the core objectives of the system, which revolve around optimizing traffic flow, reducing congestion, and enhancing the overall efficiency of traffic management. With these objectives in mind, the next step involves defining the specific requirements for the system, encompassing hardware, software, connectivity, and environmental considerations.

To effectively gather real-time traffic data, a data collection mechanism should be established, often through the deployment of surveillance cameras at key junctions. The captured images will then undergo image processing techniques, such as edge detection and foreground/background modeling, to extract pertinent information.

3.2.1 Design Intelligent Traffic Light Control

Intelligent traffic light control is a core element of the system, necessitating the development of algorithms that make decisions based on sensor readings and comprehensive traffic data analysis. Furthermore, the incorporation of Digital Signal Processing (DSP) and Nios II, with a dual-CPU architecture and FPGA logic control, is vital for functions like cross-phase adjustment and information exchange.

3.3 Algorithms and Techniques used

The techniques of edge detection, background subtraction, blob analysis, optimum flow, and image preprocessing are often used in the field of computer vision. These methods play a crucial role in many applications such as object recognition, motion tracking, and picture segmentation. Edge detection aims to identify the boundaries between different objects or regions in an image. Background

3.3.1 Optical flow

Algorithms are utilized to estimate the motion of objects over a sequence of images. By capturing the movement patterns of vehicles, these algorithms enable the system to detect vehicle mobility and analyze the overall traffic flow effectively.

Chapter 4

Implementation

4.1 Tools and Technologies used

This academic paper discusses several technologies used in the field of computer science, including image processing libraries, machine learning frameworks, cloud computing platforms, and Internet of Things (IoT) platforms. These technologies play a crucial role in the development and implementation of advanced computational systems. The paper explores the features, functionalities, and applications of each technology, providing an overview of their significance in the field. Additionally, the paper examines the interplay between these technologies and their potential impact on various industries and domains.The topics of interest include communication protocols, geographic information systems, and traffic simulation software.

4.1.1 Communication Protocols

Communication Protocols: Communication protocols such as MQTT (Message Queuing Telemetry Transport) and HTTP are used for data exchange between various system components, including sensors, cameras, and the central traffic management server.

4.2 Modules and their descriptions

Image Processing: This module utilizes image processing techniques to analyze the captured images from surveillance cameras. It performs tasks such as edge detection, object recognition, and vehicle counting to extract relevant information about traffic flow and congestion .

Input Data: This module is responsible for collecting real-time traffic data from various sources, including surveillance cameras, sensors, and connected vehicles. It captures information such as vehicle counts, speeds, and occupancy, which is used as input for further analysis and decision-making .

Main: The main module serves as the central control system of the traffic management system. It coordinates the different modules, receives input data, and executes the necessary algorithms and strategies to optimize traffic flow and manage congestion .

Setup: The setup module is responsible for configuring and initializing the traffic management system. It includes tasks such as setting up communication protocols, connecting to data sources, and configuring system parameters .

Traffic Prediction: This module uses historical and real-time traffic data to predict future traffic conditions. It employs algorithms and machine learning techniques to forecast traffic congestion, travel times, and identify potential bottlenecks. These predictions help in making informed decisions for traffic management strategies .

Optimized Traffic Manager: This module is responsible for optimizing traffic flow and reducing congestion based on the predictions and real-time data. It uses algorithms and decision-making techniques to dynamically adjust traffic signal timings, reroute vehicles, and implement traffic control strategies to minimize delays and improve overall traffic efficiency .

4.2.1 Main

The main module serves as the central control system of the traffic management system. It coordinates the different modules, receives input data, and executes the necessary algorithms and strategies to optimize traffic flow and manage congestion .

4.3 Flow of the System

4.3.1 Image Processing

Image Processing: This module utilizes image processing techniques to analyze the captured images from surveillance cameras. It performs tasks such as edge detection,object

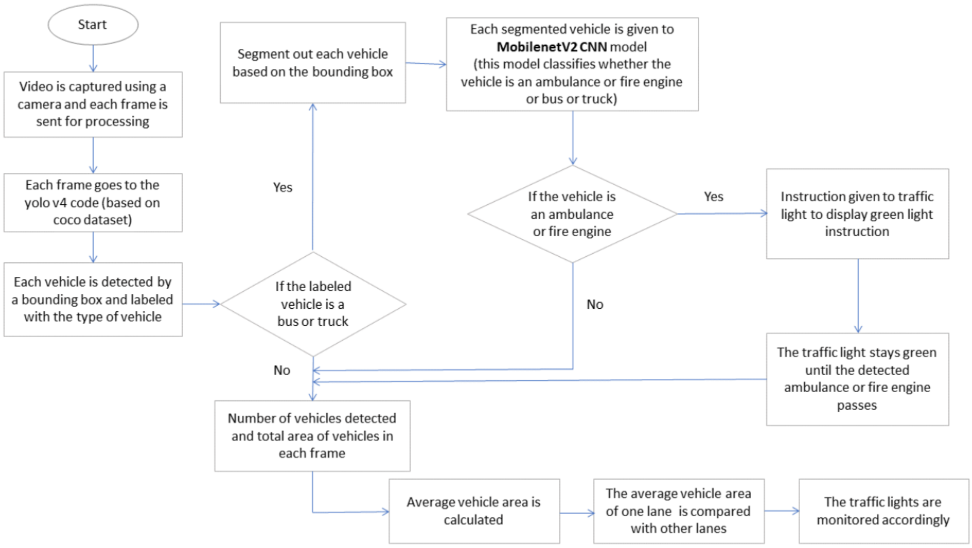


Figure 4.1: System Architecture for detection of Emergency Vehicles and Traffic Management

recognition, and vehicle counting to extract relevant information about traffic flow and congestion .

Chapter 5

Results and Analysis

5.1 Performance Evaluation

Initially, a CNN model with 8 layers instead of the MobileNet V2 CNN model was experimented. The 8-layered CNN structure consists of two convolutional layers, two Max Pooling layers, one dropout layer of 0.5, one fattening layer, and the last two Fully connected layers out of which one has activation functions as relu and another as a sigmoid. The Validation accuracy of Experimented 8-layered CNN model is 69.79MobileNet V2 CNN model has 95.49and The Validation loss of 0.181. The performance of the CNN model is very low with low validation accuracy and high validation loss when compared to the MobileNet V2 CNN model

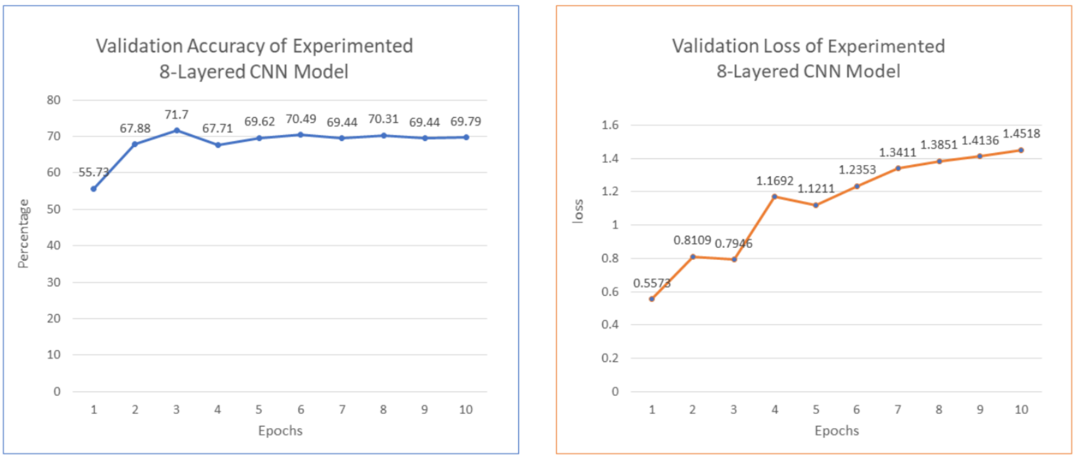
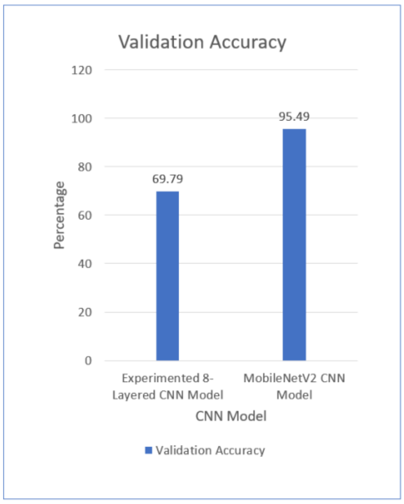


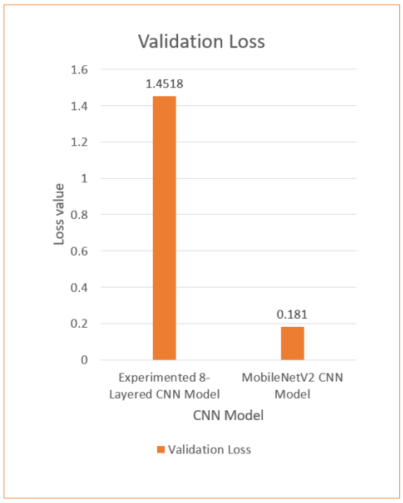
Figure 5.1: Performance Metrics of 8-Layered CNN Model

5.2 Comparison with existing systems

A project description is a high-level overview of why you're doing a project. The document explains a project's objectives and its essential qualities. Think of it as the elevator pitch that focuses on what and why without delving into how.



(a) Comparison of Validation Accuracy



(b) Comparison of Validation Loss

Figure 5.2: Comparison of Experimented 8-layered CNN Model and MobileNet V2 CNN Model

5.3 Limitations and future scope

Low Light Conditions: The system may not perform optimally in low light conditions, as the image processing algorithms may struggle to accurately detect and analyze traffic patterns . Emergency Vehicle Detection: The system does not have the capability to distinguish emergency vehicles such as ambulances or fire brigades, which may require priority in traffic management . Hardware Dependency: The system relies on surveillance cameras and sensors for data collection, which may require additional installation and maintenance costs .

Enhanced Low Light Performance: The system can be improved by incorporating infrared cameras or other suitable image processing techniques to ensure effective traffic control even in low light or night-time conditions . Emergency Vehicle Priority: By implementing advanced image processing algorithms, the system can be enhanced to detect and prioritize emergency vehicles, allowing them to pass through traffic more efficiently . Data Analytics and Integration: Leveraging real-time data from various sources, such as social media, GPS, and mobile apps, can enable more accurate traffic predictions and better decision-making in traffic management . Intelligent Coordination Control: Implementing a coordinated system of traffic signals can further optimize traffic flow by synchronizing signal timings to create long strings of green lights for drivers . Environmental Considerations: The system can be expanded to incorporate environmental factors such as air quality monitoring and emission control, contributing to sustainable and eco-friendly traffic management .

Chapter 6

Conclusion and Recommendations

6.1 Summary of the Project

The project aims to improve traffic management in urban cities by utilizing image processing and surveillance systems for real-time traffic control. Cameras capture images of lanes and calculate traffic density, optimizing traffic signal control. By switching traffic lights based on vehicle density, the project reduces congestion, accidents, fuel consumption, and waiting time. The proposed system uses surveillance cameras on four-way junctions, requiring no constant maintenance and less prone to failure. Image processing techniques like resizing, RGB to gray conversion, enhancement, and edge detection are employed, with the Canny Edge Detection algorithm being used for robust edge detection. The system is implemented as the primary controller, connecting to traffic lights and transmitting data to the cloud for analytics. The project highlights potential benefits, including reduced fuel consumption, lower maintenance costs, and improved vehicle spacing. Future enhancements could involve infrared cameras for low light conditions and image processing techniques for emergency vehicles.

6.2 Contributions and achievements

Real-time traffic management systems can enhance transportation efficiency by optimizing signal timings, reducing congestion, and minimizing delays. This can improve traffic flow, enhance safety by monitoring traffic conditions, and optimize resource utilization. By dynamically adjusting traffic signal timings, these systems can reduce fuel consumption and environmental impact. Real-time traffic information can help drivers make informed decisions about routes and avoid congested areas. Data-driven decision-making can also benefit from these systems, providing valuable insights for transportation planning, infrastructure development, and policy-making. Overall, traffic management systems contribute to improved road safety and efficient resource utilization.

6.3 Recommendations for future work

The incorporation of Intelligent Transportation Systems (ITS), the integration of Advanced Machine Learning Techniques, the incorporation of Multi-Modal Traffic Management, the integration of Big Data Analytics, and the incorporation of Sustainable and Green Traffic Management are all crucial components of traffic management.

Bibliography

[1] S. Deepajothi, D. P. Rajan, P. Karthikeyan, and S. Velliangiri, “Intelligent traffic management for emergency vehicles using convolutional neural network,” vol. 1, pp. 853–857, 2021.

[2] N. N. S. N. Dzulkefli, S. Rohafauzi, A. N. Jaafar, R. Abdullah, R. Shafie, M. S. Selamat, N. S. Azman, and M. Z. Z. Muhammad, “Density based traffic system via ir sensor,” vol. 1529, no. 2, p. 022061, 2020.

[3] S. Roy and M. S. Rahman, “Emergency vehicle detection on heavy traffic road from cctv footage using deep convolutional neural network,” pp. 1–6, 2019.

[4] R. Harshitha, H. Shetty, B. Kavitharani, R. Madhura, and M. Jamuna, “Automatic traffic management system using yolo,” Available at SSRN 3328586, 2018.

[5] U. E. Prakash, K. Vishnupriya, A. Thankappan, and A. A. Balakrishnan, “Density based traffic control system using image processing,” pp. 1–4, 2018.

[6] G. M. Bhoomika, “Ambulance detection using image processing,” International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2022.

[7] K. Agrawal, M. Nigam, S. Bhattacharya, and G. Sumathi, “Ambulance detection using image processing and neural networks,” vol. 2115, no. 1, p. 012036, 2021.

[8] S. V. Bhate, P. V. Kulkarni, S. D. Lagad, M. D. Shinde, and S. Patil, “Iot based intelligent traffic signal system for emergency vehicles,” pp. 788–793, 2018.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020

Appendices

Appendix A

Source code

A project description is a high-level overview of why you're doing a project. The document explains a project's objectives and its essential qualities. Think of it as the elevator pitch that focuses on what and why without delving into how.

```
# -*- coding: utf-8 -*-
"""Traffic_Prediction.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1zCo6Lw7Kup2i9aggeV78Dka_yZczGjTT
"""

import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

dataframe = pd.read_csv(r'dataset/TrafficDataset10000.csv')
dataframe.head()

dataset = np.array(dataframe)

m = dataset.shape[0]

s= int(0.9*m)

train = dataset[:s,:]
test = dataset[s:,:]

train_predictors = train[:,:4]
train_labels = train[:,21]

test_predictors = test[:,:4]
test_labels = test[:,21]

#print(predictors)
#print(labels)
```

```

train_predictors = tf.convert_to_tensor(train_predictors.astype(float))
train_labels = tf.convert_to_tensor(train_labels.astype(float))
test_predictors = tf.convert_to_tensor(test_predictors.astype(float))
test_labels = tf.convert_to_tensor(test_labels.astype(float))

model = tf.keras.models.Sequential([
                                    tf.keras.layers.Dense(units = 128, input_shape = [4], activation='relu'),
                                    tf.keras.layers.Dense(units = 64, activation='relu'),
                                    tf.keras.layers.Dense(units = 1)
                                   ])
model.summary()
model.compile(optimizer='adam',
              loss='mean_squared_error',
              )

history = model.fit(train_predictors, train_labels, epochs = 200, validation_data=(test_predictors, test_labels), batch_size = 150, steps_per_epoch=100, validation_steps = 10)

model.evaluate(test_predictors, test_labels)

x = model.predict(test_predictors)
y = test_labels

t = np.array(range(test_predictors.shape[0]))

print(x[1:10])
print(y[1:10])

plt.plot(t,y)
plt.plot(t,x)
plt.title("Comparison between predicted and actual values")
plt.ylabel('Optimum Time Cycle')
plt.xlabel('Time')
plt.legend(['Actual', 'Predicted'], loc='upper left')
plt.show()

plt.plot(t,y)
plt.show()

plt.plot(t,x)
plt.show()
```

Appendix B

Screen shots

B.1 Output

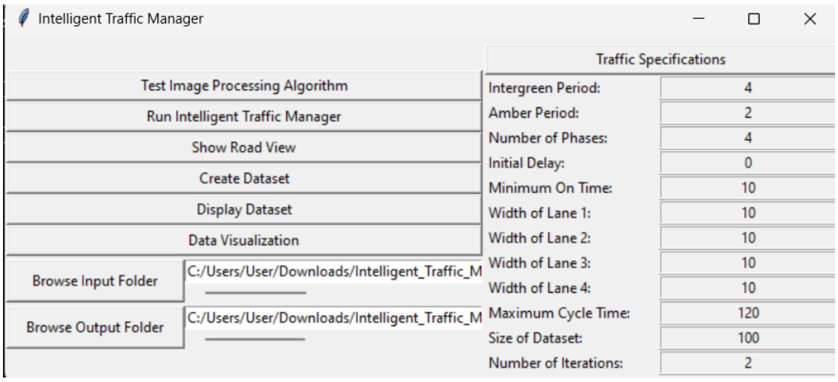


Figure B.1: Graphic Unit

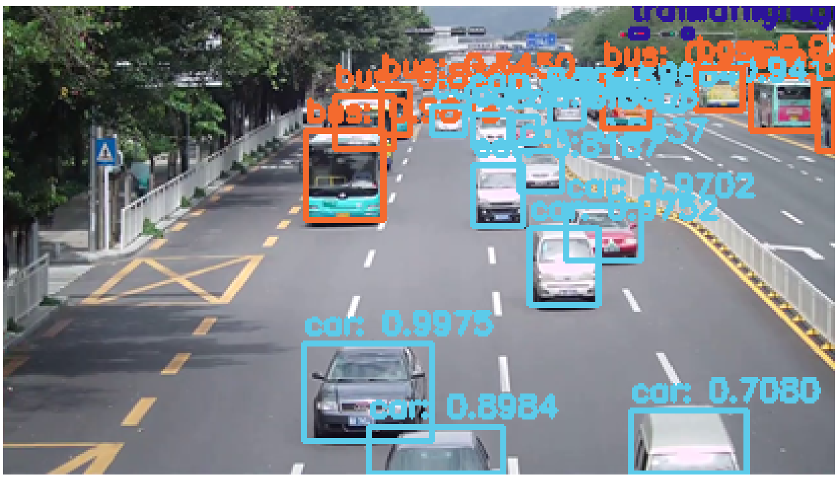


Figure B.2: Recognition of vehicles

Appendix C

Data sets used in the project

Initially, a CNN model with 8 layers instead of the MobileNet V2 CNN model was experimented. The 8-layered CNN structure consists of two convolutional layers, two Max Pooling layers, one dropout layer of 0.5, one fattening layer, and the last two Fully connected layers out of which one has activation functions as relu and another as a sigmoid. The Validation accuracy of Experimented 8-layered CNN model is 69.79 Percentage and the Validation loss of 1.4518 where as the MobileNet V2 CNN model has 95.49 Percentage Validation accuracy and The Validation loss of 0.181. The performance of the CNN model is very low with low validation accuracy and high validation loss when compared to the MobileNet V2 CNN model. The classification error is the parameter that gives the error percentage in classifying among categories in the training dataset which is calculated from predicted values. The classification error percentage is around 5 Percentage for the MobileNet V2 CNN model. This demonstrates that the classification error rate is quite low. Each training epoch lasts about 65 seconds, and the entire training session lasts 650 seconds. As the execution time is quick, this method is more practical. MobileNet V2 is designed to be computationally efficient, making it appropriate and convenient for mobile and embedded devices with limited processing power and it can be customized. The architecture uses depth-wise separable convolutions, which split a standard convolution into separate depth-wise and pointwise convolutions. This reduces the number of parameters and computations required compared to the Experimented 8-layered CNN model while maintaining high accuracy.