REAL TIME TRAFFIC DENSITY DETECTION AND DYNAMIC TRAFFIC CONTROL

A DESIGN PROJECT REPORT

submitted by

ABINAYA SHREE J

CHARULATHA K

HARSHITHA K

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

NOVEMBER, 2024

REAL TIME TRAFFIC DENSITY DETECTION AND DYNAMIC TRAFFIC CONTROL

A DESIGN PROJECT REPORT

submitted by

ABINAYA SHREE J (811722104004)

CHARULATHA K (811722104023)

HARSHITHA K (811722104052)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

NOVEMBER, 2024

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this project report titled **"REAL TIME TRAFFIC DENSITY DETECTION AND DYNAMIC TRAFFIC CONTROL"** is Bonafide work of **ABINAYA SHREE J (811722104004), CHARULATHA K (811722104023), HARSHITHA K (811722104052)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr A Delphin Carolina Rani M.E.,Ph.D.,	Mrs.R. Sathya,M.E.,(Ph.D.,)
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	Assistant Professor
Department of CSE	Department of CSE
K Ramakrishnan College of Technology	K Ramakrishnan College of Technology
(Autonomous)	(Autonomous)
Samayapuram – 621 112	Samayapuram – 621 112

Submitted for the viva-voice examination held on………………

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"ADVANCED VEHICLE DETECTION AND TRAFFIC ROAD ANALYSIS USING REAL -TIME DEEP LEARNING TECHNIQUES"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of Bachelor Of Engineering. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor Of Engineering.

Signature

ABINAYA SHREE J

CHARULATHA K

HARSHITHA K

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution "**K RAMAKRISHNAN COLLEGE OF TECHNOLOGY"** for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank **Dr. N. VASUDEVAN, M.Tech, Ph.D.,**Principal, who gave opportunity to frame the project the full satisfaction.

We whole heartily thank **Dr. DELPHIN CAROLINA RANI, M.E, Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

We express my deep and sincere gratitude to my beloved project guide **Mrs. R. SATHYA, M.E,(Ph.D)** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out this project.

We render my sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

We wish to express my special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

The integration of real-time ambulance detection and dynamic traffic management offers a significant opportunity to improve emergency response times and enhance road safety. This project aims to develop a system that detects ambulances using YOLOv5, a state-of-the-art deep learning model for object detection, and utilizes multiple side cameras to monitor and analyze traffic conditions across various lanes. The system is designed to ensure swift clearance for ambulances, particularly in congested traffic scenarios, through intelligent lane prioritization and dynamic traffic signal adjustments. A key component of the project is the creation of a custom dataset for YOLOv5 to accurately detect ambulances in real-time across different environments and traffic conditions. The graphical user interface (GUI) displays real-time alerts, including flashing icons or color-coded lane indicator, to notify road users of the ambulance's location and lane priority. Additionally, the system is integrated with local traffic light control infrastructure to adjust the traffic signals dynamically. Using real- time traffic data, the system provides alternative paths for the ambulance to ensure it can reach its destination swiftly, minimizing delays.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	**v**
	LIST OF FIGURES	**ix**
	LIST OF ABBREVIATIONS	**x**
1	**INTRODUCTION**	**1**
	1.1 Overview	1
	1.2 Problem Statement	2
	1.3 Objectives	2
	1.4 Implication	3
2	**LITERATURE SURVEY**	**5**
3	**SYSTEM ANALYSIS**	**9**
	3.1 Existing System	**9**
	3.2 Proposed System	10
	3.3 Block Diagram for Proposed System	10
	3.4 Approach Used	11
	3.5 Process Cycle	11
	3.6 Flow Chart	12
	3.7 Class Diagram	13
4	**MODULES**	
	4.1 Data Collection and Acquisition	**14**

	4.2 Object Detection and Analysis	14
	4.3 Traffic Density Analysis	14
	4.4 Real- Time Management and Visualisation	15
	4.5 Data-Driven Insights and Optimization	15
	4.6 User Interface	15
5	**SYSTEM SPECIFICATION**	**16**
	5.1 Hardware Requirements	16
	5.2 Software Requirements	16
6	**METHODOLOGY**	**17**
	6.1 Introduction	17
	6.2 Data Collection and Preprocessing	17
	6.3 Model Training	18
	6.4 Real-Time Camera Feed Processing	18
	6.5 Traffic Condition Evaluation and Lane Prioritization	18
	6.6 Dynamic Traffic Signal Control	19
	6.7 Rerouting the Ambulance	19
	6.8 Notification System for Traffic Authorities	20
	6.9 Graphical User Interface(GUI)	20
	6.10 Continuous Monitoring and Adaptation	21
7	**CONCLUSION AND FUTURE ENHANCEMENT**	**23**
	7.1 Conclusion	23
	7.2 Future Enhancement	24

APPENDIX A (Source Code) 25

APPENDIX B (Output Screenshots) 36

REFERENCES 39

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	Proposed System Architecture	10
3.2	Process Overview	11
3.3	Moments and Result	12
3.4	Function involved and Actions	13
6.1	Simulation Overview	22

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
YOLO	You Only Look Once
CNN	Convolutional Neural Network
CV	Computer Vision
PINN	Physics Informed Neural Network
GUI	Graphical User Interface
IoT	Internet of Things
IDE	Integrated Development System
EV	Electric Vehicle
AR	Augmented Reality
VR	Virtual Reality
ML	Machine Learning
GPS	Global Positioning System
COCO	Common Objects in Context
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The proposed system integrates Artificial Intelligence (AI), real-time traffic monitoring, and dynamic traffic management to mitigate ambulance delays in urban traffic. Leveraging deep learning, computer vision, and IoT, it prioritizes ambulance movement and enhances Emergency Medical Services (EMS) efficiency. This innovative approach addresses the critical issue of traffic congestion impacting ambulance response times. The system utilizes advanced object detection algorithms (YOLOv5) and a multi-camera surveillance setup for 360-degree visibility. When an ambulance is detected, the system evaluates traffic conditions, identifies potential congestion, and dynamically adjusts traffic signals to provide a clear path. Additionally, it incorporates rerouting capabilities, analyzing real-time traffic data to recommend alternative routes.

The system communicates with local authorities and emergency services through automated notifications, providing real-time updates on ambulance location, traffic conditions, and signal status. The user-friendly interface offers a graphical representation of traffic conditions, ambulance locations, and signal status, enhancing public awareness and encouraging cooperation with emergency protocols. This integrated solution reduces ambulance response times, enhances EMS efficiency, and improves overall traffic flow management.

1.2 PROBLEM STATEMENT

In urban areas, traffic congestion poses a significant challenge to emergency medical services (EMS), particularly ambulances, which are critically dependent on timely arrival at their destinations. When responding to emergencies, every second counts, and delays in reaching the patient or the hospital can lead to severe consequences, including worsening of medical conditions or even loss of life. Despite advancements in traffic management, existing systems fail to address the specific needs of emergency vehicles, exacerbating delays during critical situations. The proposed system integrates AI, computer vision, and IoT technologies to address these issues. It employs multi-camera surveillance for 360-degree coverage, accurately detecting ambulances and evaluating traffic conditions. Once detected, the system dynamically adjusts traffic lights to prioritize the ambulance's lane, clearing congestion and ensuringsmooth passage. It also suggests alternative routes in case of severe traffic and communicates real-time updates to authorities. This scalable system is designed to work autonomously, significantly reducing delays for ambulances.

1.3 OBJECTIVES

The primary objective is to develop an intelligent traffic management system that prioritizes ambulance movement, reducing response times and saving lives. This system integrates AI, computer vision, and IoT to optimize traffic signal control and ensure swift emergency response.

- **System Components-**The proposed system utilizes a fine-tuned YOLOv5 model for accurate real-time ambulance detection, leveraging a custom dataset to minimize false positives and ensure reliable identification. This model is trained on various ambulance types, lighting conditions, and environments.

- **Comprehensive Monitoring-**Four strategically placed cameras provide 360-degree monitoring of traffic conditions, capturing ambulance movement from all directions, including intersections, congested areas, and nighttime scenarios.This comprehensive coverage enables efficient detection and response.
- **Intelligent Traffic Management-**The system adjusts traffic signals dynamically to prioritize lanes with detected ambulances, reducing delays, congestion, and potential accidents. This optimization considers real-time traffic data, road conditions, and priority protocols.
- **Automated Alerts-**Real-time notifications are sent to local traffic authorities, emergency services, nearby hospitals, ambulance dispatch centers, and relevant stakeholders, facilitating coordinated clearance efforts and minimizing response times.
- **Optimized Routing-**An advanced algorithm analyzes real-time traffic data to suggest optimal alternative routes, ensuring ambulances reach destinations efficiently. Factors considered include traffic congestion, road closures, construction, time of day, and weather conditions.
- **Enhanced Response-**The system improves public safety, healthcare outcomes, emergency response efficiency, reduces healthcare costs, and saves lives by minimizing ambulance delays. Additionally, it enhances situational awareness, streamlines emergency response protocols, and promotes data-driven decision-making.

1.4 IMPLICATION

The proposed ambulance detection and traffic management system offers numerous benefits. Improved emergency response times save lives and minimize damage. Enhanced health outcomes result from faster medical access. Increased public safety is ensured through automated clearance and optimized traffic flow. Ultimately, the system integrates with smart city infrastructure, providing scalable and efficient solutions.

Improved Emergency Response

The proposed system significantly reduces emergency response times byprioritizing ambulance movement through real-time detection and dynamic traffic signal management. Studies suggest response times can be reduced by up to 50%, enabling faster arrival at emergency scenes and saving lives. Efficient emergency response protocols are crucial in life-threatening situations.

Enhanced Health Outcomes

Faster access to medical care profoundly impacts health outcomes, particularly in time-sensitive situations. Quick response times minimize damage from heart attacks, improve recovery chances for stroke victims, and enhance survival rates for severe trauma patients. Reduced mortality rates and decreased risk of complications from delayed medical attention are additional benefits.

Increased Public Safety

Automated clearance of paths for ambulances and emergency vehicles enhances road safety for emergency responders, pedestrians, and motorists. By optimizing traffic flow, the system reduces congestion-related hazards and accident risks. This integrated approach ensures safer passage for emergency vehicles.

Smart City Integration

The proposed system contributes to the vision of smart cities by leveraging AI, IoT, and machine learning. Real-time data-driven decision-making optimizes traffic management, enhancing urban planning and emergency response strategies. Integration with existing smart city infrastructure enables seamless coordination and improved quality of life for residents.

Scalable and Efficient Solutions

This scalable solution is applicable to various urban settings, accommodating multiple emergency vehicles. By streamlining emergency response efforts, the system reduces healthcare costs and economic burdens. Enhanced efficiency and effectiveness enable emergency response teams to focus on saving lives rather than navigating traffic chaos.

CHAPTER 2

LITERATURE SURVEY

2.1 Title: AI-Integrated Traffic Information System: A Synergistic Approach of Physics-Informed Neural Network and GPT-4 for Traffic Estimation and Real-Time Assistance

Authors: Tewodros Syum Gebre, Leila Hashemi-beni , Eden Tsehaye Wasehun, and Freda Elikem Dorbu

Year: 2024

Publication: IEEE ACCESS

Description: This paper proposes an innovative AI-integrated traffic system, combining Physics-Informed Neural Networks (PINNs) and GPT-4. Existing models rely on historical data or simplistic algorithms, yielding inaccurate predictions. PINNs incorporate physical laws governing traffic flow, enhancing accuracy. GPT-4 enables natural language processing for real-time assistance. Key features include traffic alerts, optimal route suggestions and traffic management recommendations. Experimental evaluations demonstrate robustness across urban environments. Results show significant improvements in accuracy, response time and user engagement. The user- friendly interface provides easy access to real-time data. Improved decision-making andtraffic flow management are achieved. Potential applications include smart city frameworks and transportation technology. Future research directions include integrating IoT devices and social media feeds. This contributes to smarter, sustainable urban transportation solutions.

2.2 Title: Artificial Hummingbird Optimization Algorithm with Hierarchical Deep Learning for Traffic Management in Intelligent Transportation Systems

Authors: Abdulrahman Alruban, Hanan Abdullah Mengash, Majdy M. Eltahir ,Nabil Sharaf Almalki, Ahmed Mahmud, And Mohammed Assiri

Year: 2024

Publication: IEEE ACCESS

Description: This paper combines Artificial Hummingbird Optimization Algorithm (AHOA) and Hierarchical Deep Learning (HDL) for enhanced traffic management. Conventional methods struggle with real-time adaptability and resource allocation. AHOA optimizes traffic routing and resource distribution inspired by hummingbirds' foraging behavior. HDL models complex traffic scenarios through multiple abstraction layers. Simulations in diverse urban settings demonstrate improved route optimization and efficiency. Results show lower travel times and reduced congestion. The model is scalable, adaptable, and suitable for smart city initiatives. Potential applications include real-time monitoring, emergency response, and multimodal transportation. The system addresses various traffic management challenges. Future research recommendations include integrating IoT sensor data and hybrid optimization techniques. This enhances traffic management strategies in evolving urban landscapes. The approach optimizes traffic flow and resource allocation. Improved decision-making results from accurate predictions.

2.3 Title: Context-Aware Multiagent Broad Reinforcement Learning for Mixed Pedestrian-Vehicle Adaptive Traffic Light Control

Authors: RuijieZhu ,Shuning Wu, , Lulu Li,PingLv, , Mingliang Xu

Year: 2022

Publication: IEEE internet of things journal

Description: This paper introduces Context-Aware Multiagent Broad Reinforcement Learning (CAMBRL) for adaptive traffic light control. Traditional systems fail to address dynamic pedestrian-vehicle interactions, increasing delays and safety risks. CAMBRL integrates multiple agents, representing vehicles and pedestrians, analyzing traffic conditions comprehensively. Context-aware mechanisms adapt decision-making based on real-time factors like traffic density, pedestrian flow, and weather. Simulations demonstrate reduced waiting times and enhanced traffic flow efficiency. The system learns and adapts over time, improving performance. Scalable and flexible, CAMBRL addresses diverse traffic scenarios within smart city infrastructure. Future research directions include integrating real-time surveillance feeds and IoT devices. CAMBRL optimizes traffic flow, reducing congestion. Waiting times decrease for pedestrians and vehicles. Traffic safety and efficiency improve. Overall, CAMBRL enhances urban mobility. Effective traffic management is achieved.

2.4 Title: Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation Systems

Authors: Neetesh Kumar, Sarthak Mittal, Vaibhav Garg, Neeraj Kumar

Year: 2022

Publication: IEEE transactions on intelligent transportation systems

Description: This paper proposes a Deep Reinforcement Learning (DRL) framework for traffic light scheduling within Software-Defined Networking (SDN) environments. Traditional systems rely on fixed timing, causing suboptimal traffic flow. The DRL-based framework continuously learns from real-time traffic data, adjusting timings accordingly. SDN integration enables centralized control and dynamic adaptation. Simulations demonstrate improved traffic throughput and reduced travel times. Vehicle delay and queue length show substantial enhancements. The framework is scalable and suitable for smart city initiatives. Future research directions include multi-agent systems and predictive analytics. The approach optimizes traffic flow and reduces congestion. Real-time adaptation enhances responsiveness. Traffic management solutions become more efficient. Overall effectiveness is significantly improved. Intelligent transportation technologies are integrated.

2.5 Title: Development of a Smart Traffic Light Control System With Real-Time Monitoring

Authors: Luiz Fernando Pinto de Oliveira ,Leandro Tiago Manera ,Paulo Denis Garcez da Luz

Year: 2021

Publication: IEEE internet of things journal

Description: This paper presents an advanced Smart Traffic Light Control System, integrating sensors, cameras and machine learning algorithms to optimize traffic light timings based on real-time traffic data. Traditional systems' fixed schedules often cause congestion. Field tests demonstrate reduced vehicle waiting times, lower congestionand improved pedestrian safety. The user-friendly interface provides real-time insights for informed decision-making. Potential integrations include existing smart city infrastructure and IoT devices. Future research directions include scalability and application in diverse urban contexts, contributing to smarter transportation networks.

2.6 Title: Edge ML Technique for Smart Traffic Management in Intelligent Transportation Systems

Authors: A nakhi Hazarika, Nikumani Choudhury, Moustafa M. Nasralla, Sohaib Bin Altaf Khattak, And Ikram Ur Rehman

Year: 2024

Publication: IEEE ACCESS

Description: This paper introduces Edge Machine Learning (Edge ML) for smart traffic management, overcoming cloud-based solution limitations. Traditional methods suffer from latency and connectivity issues, causing delayed decision-making and congestion. The Edge ML framework processes real-time traffic data locally, reducing latency and enabling immediate adjustments. It consists of data acquisition units, edge computing nodes and a centralized control interface, analyzing traffic information and sending insights for further analysis. Edge ML enhances traffic signal control, routing and management, improving responsiveness, reducing congestion and travel times, and suitability for existing Intelligent Transportation Systems (ITS).

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Traffic monitoring systems have evolved to manage urban mobility and enhance road safety. Traditional methods include loop detection systems, measuring traffic volume, speed, and occupancy accurately, but requiring costly installation and maintenance. CCTV-based monitoring systems provide real-time visuals, but need human oversight and raise privacy concerns. Modern approaches include radar and Lidar systems, offering accurate data in various weather conditions, but with high setup costs. GPS-based systems provide real-time insights over extensive areas without fixed installations, but may lack accuracy. Mobile phone data analytics offer low-cost, real-time data without infrastructure, but raise privacy and accuracy concerns. Machine learning and computer vision detect vehicles accurately, requiring computational resources and high-quality video inputs. Integrated traffic management systems combine multiple technologies for comprehensive oversight, enabling real-time adjustments and data-driven decision-making. Effective traffic management has become a priority. Various systems capture, analyze, and respond to traffic conditions. Loop detectors measure traffic volume, speed, and occupancy. CCTV cameras provide real-time visuals. Radar and Lidar systems detect vehicles accurately. GPS-based systems track traffic flow. Machine learning enhances detection accuracy. Integrated systems optimize traffic flow. Real-time data informs decision-making, enhancing mobility.

3.2 PROPOSED SYSTEM

A proposed system integrates cutting-edge technologies to enhance ambulance movement efficiency in urban traffic. It utilizes YOLOv5 for real-time ambulance detection, multiple side cameras for traffic monitoring, and dynamic traffic signalcontrol for lane prioritization. The system adjusts traffic signals, provides rerouting suggestions, and sends notifications to traffic authorities and emergency services. A graphical user interface offers real-time monitoring and control. The system reduces delays and congestion, ensuring ambulances reach destinations quickly. It optimizes emergency response efforts, providing timely medical care. The system's scalability andflexibility suit various urban settings. Real-time decision-making and data-driven approaches enhance traffic management. Ambulance response times are minimized, saving lives. The system streamlines communication between stakeholders**.**

3.3 BLOCK DIAGRAM FOR PROPOSED SYSTEM

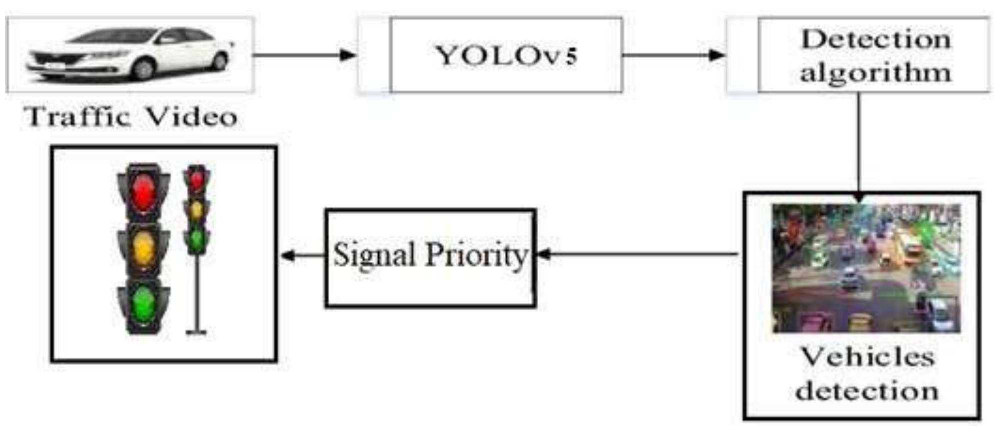

Fig 3.1 Proposed System Architecture

3.4 APPROACH USED

The project employs various approaches, including YOLOv5 object detection, deep learning, and multi-camera setup for traffic monitoring. Dynamic traffic signal control and lane prioritization optimize ambulance movement. Rerouting capabilities suggest alternative routes. Automated notifications alert traffic authorities and emergency services. A graphical user interface provides real-time monitoring. Real- time data analytics and historical data analysis enhance traffic management. Integration with traffic infrastructure and emergency services streamlines communication. Machine learning algorithms improve detection accuracy. Data-driven decision-making enables efficient traffic management. These approaches combine to reduce ambulance response times and enhance urban mobility.

3.5 PROCESS CYCLE

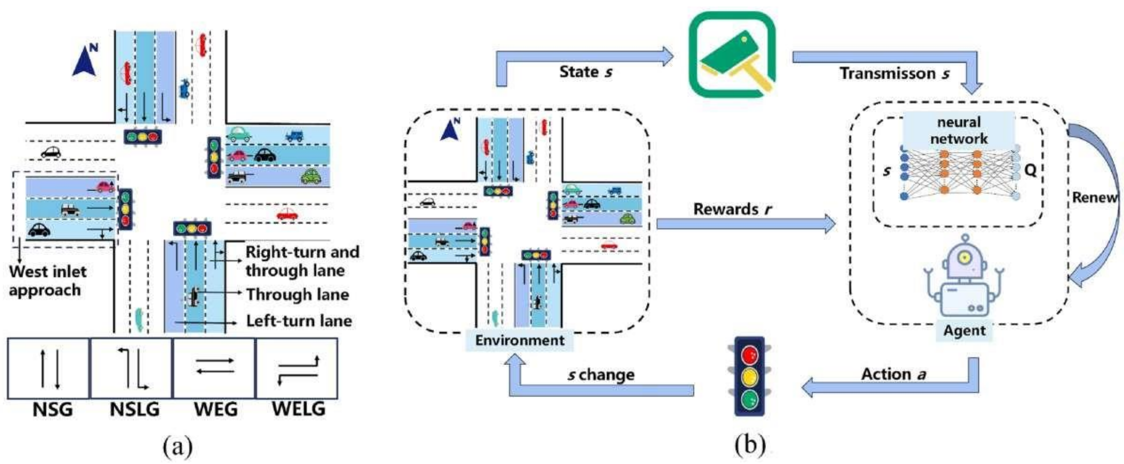

Fig 3.2 Process Overview

3.6 FLOWCHART

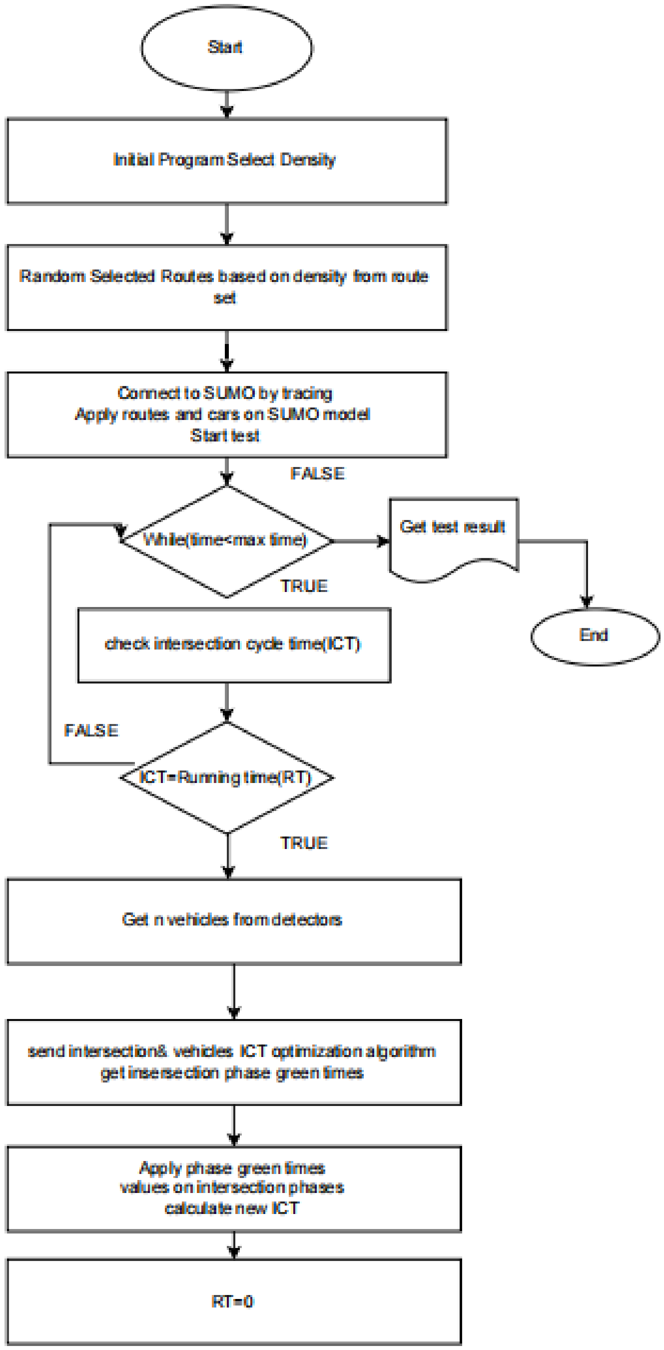

Fig 3.3 Moments and Results

3.7 CLASS DIAGRAM

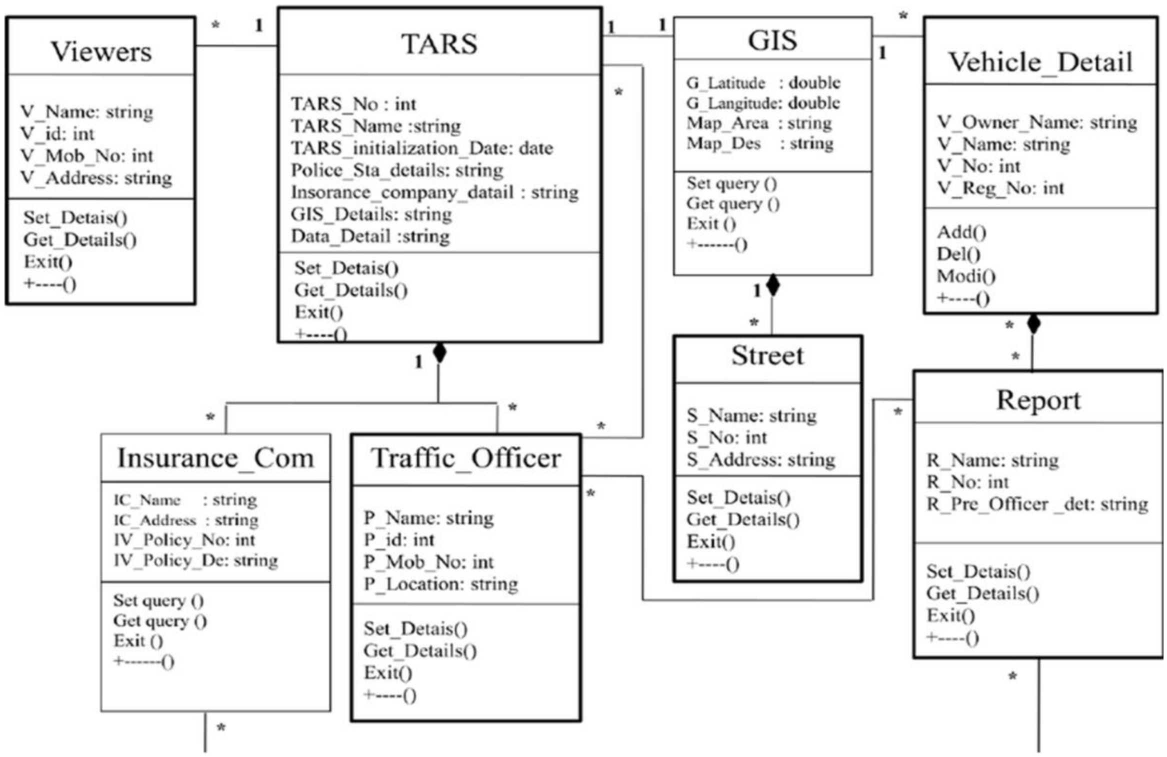

Fig 3.4 Functions Involved and Actions

CHAPTER 4

MODULE DESCRIPTION

4.1 Module 1: Data Collection and Acquisition

The system's foundation lies in high-definition cameras capturing live video feeds of roadways. Strategically positioned cameras provide comprehensive coverage of intersections, highways and high-traffic areas. Data acquisition processes these feeds, compressing and extracting frames for analysis. This module ensures efficient data transmission and processing, leveraging advanced compression algorithms and frame extraction capabilities.

4.2 Module 2: Object Detection and Analysis

The YOLOv5-powered object detection module analyzes frames to detect and classify vehicles, including cars, trucks, buses and motorcycles. Traffic density analysis calculates congestion levels, enabling informed decisions regarding traffic management. Emergency vehicle detection prioritizes lanes, minimizing response times. Deep reinforcement learning optimizes traffic signal control, adapting to changing conditions. Object detection accuracy is enhanced through continuous model training and updates.

4.3 Module 3: Traffic Density Analysis

The proposed traffic monitoring system computes traffic density for each lane, analyzing congestion levels and informing traffic management decisions. Traffic density is calculated by counting vehicles in specific time intervals and determining flow rates. The system aggregates data over time, visualizing density patterns and identifying peak congestion periods. Dynamic traffic density heat maps represent congestion levels across lanes and intersections. The traffic density analysis module provides real-time updates, reflecting current traffic conditions. Advanced algorithms predict traffic trends based on historical data, enabling proactive congestion management. This enhances the system's capability to monitor and manage urban traffic flows efficiently.

4.4 Module 4: Real-Time Management and Visualization

Real-time data visualization presents key metrics in intuitive formats, including dashboards, heat maps and alerts. Traffic signal control integrates with existing infrastructure, adjusting timings based on real-time data. The user interface provides easy access to data, visualizations and alerts, facilitating prompt decision-making. Customizable dashboards cater to specific operational requirements. Real-time data sharing enables coordinated responses to traffic incidents.

4.5 Module 5: Data-Driven Insights and Optimization

Historical data storage enables trend analysis, informing strategic planning and infrastructure development. Reporting and insights generation provide actionable recommendations for optimizing traffic flow, enhancing safety and improving overall urban traffic management effectiveness. Predictive analytics forecast traffic patterns, identifying potential congestion points. Data-driven decision-making enables proactive management, reducing congestion and improving emergency response times.

4.6 Module 6: User Interface (UI)

The user interface (UI) of the proposed traffic monitoring system provides real-time data, visualizations, and alerts for traffic management personnel. The UI is user-friendly, intuitive, and customizable, displaying key metrics in easily interpretable formats. Operators can monitor traffic conditions, access detailed reports, and navigate between modules with ease. The UI facilitates multi-user access, enabling coordinated responses to incidents.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 Hardware Requirements

- **Processor:** Intel processor2.6.0GHZ
- **RAM:** 8GB
- **Hard disk:** 160 GB Compact Disk: 650Mb
- **Keyboard:** Standard keyboard
- **Monitor :**15inchcolormonitor

5.2 SOFTWARE REQUIREMENTS

- **Operating system:** Windows 10
- **Front End:** PYTHON
- **Tool:** PYCHARM Application
- Google colab

CHAPTER 6

METHODOLOGY

6.1 Introduction

The methodology for the proposed ambulance prioritization system involves several stages, from the data collection and model training to the integration of real- time traffic monitoring and decision-making processes. Each stage has been meticulously designed to ensure that the system operates efficiently and accurately, witha focus on optimizing ambulance movement through urban traffic. The system leveragesadvanced technologies, including deep learning models, computer vision, and dynamic traffic management algorithms, to deliver timely and effective intervention in emergency situations. Below is a step-by-step breakdown of the methodology:

6.2 Data Collection and Preprocessing

The first step in the methodology involves gathering a diverse set of data to train the object detection model. This includes collecting a wide variety of images and videos of ambulances in different settings, such as urban streets, highways, and intersections. The data must cover various weather conditions, times of day, and traffic situations to ensure that the YOLOv5 model can generalize well to real-world scenarios. This dataset should also include a range of ambulance types and images captured from different angles to ensure robustness. Once the data is collected, preprocessing is performed, which includes labeling the ambulance images with bounding boxes to identify their location within the frame. Data augmentation techniques, such as random cropping, flipping, and color variations, are applied to expand the dataset and improve the model'sability to detect ambulances in different environmental conditions.

6.3 Model Training

The YOLOv5 model is chosen due to its speed and accuracy in real-time object detection tasks. The model is pre-trained on a large general-purpose dataset (such as COCO) and then fine-tuned on the custom ambulance dataset. During training, the model learns to identify ambulances based on various visual cues, such as flashing lights, sirens, and the red cross emblem. The model is evaluated using metrics such as mean average precision (mAP) and intersection over union (IoU) to ensure that it is capable of detecting ambulances accurately, even in challenging conditions such as occlusion, lighting changes, or overlapping vehicles. Hyperparameters like learning rate, batch size, and epochs are adjusted to optimize the model's performance. Once training is complete, the model is tested on a separate validation set to assess its ability to detect ambulances under different traffic conditions.

6.4 Real-Time Camera Feed Processing

The trained YOLOv5 model is integrated into the system for real-time ambulance detection. The system uses multiple cameras strategically placed around intersections or busy road segments to capture live video feeds. These side cameras provide various angles to track the ambulance's movement, helping to detect it even if partially blocked by other vehicles. Each camera feed is processed by the YOLOv5 model to identify ambulances within the frame. The system uses a computer vision pipeline that continuously analyzes the video data to detect the presence and position of ambulances. Once an ambulance is detected, its coordinates are extracted, and the system evaluates whether the ambulance is facing any obstructions or delays in its path.

6.5 Traffic Condition Evaluation and Lane Prioritization

Once the ambulance is detected, the system proceeds to analyze the traffic conditions surrounding the ambulance. Using real-time video feeds from the cameras, the system evaluates the density of vehicles in each lane, determining whether any lanes are blocked or congested. The goal is to identify the most efficient lane for the ambulance to travel through and prioritize it. The system uses an algorithm that calculates lane congestion based on vehicle proximity, speed, and movement patterns.If the detected lane is congested, the system flags it as a priority lane and starts the next phase of intervention: adjusting traffic signals. The traffic condition evaluation also includes detecting obstacles and any other traffic-related anomalies that may affect the ambulance's movement. These evaluations help the system decide whether lane prioritization alone is enough or whether rerouting the ambulance is necessary.

6.6 Dynamic Traffic Signal Control

The next step in the methodology is dynamic control of the traffic light system. Once the ambulance has been identified and its lane has been flagged as a priority, the system communicates with the traffic light infrastructure to change signal patterns. The system calculates the most appropriate action by analyzing the ambulance’s position and its direction of movement. If the ambulance is approaching an intersection, the traffic lights in its path are automatically adjusted to give it a clear lane. The system may change the lights to green for the ambulance’s direction, while turning red for all other directions to prevent any obstruction. Additionally, the system ensures that the green light is held for an adequate amount of time for the ambulance to pass throughthe intersection safely, depending on its speed and distance from the signal. The signal changes are synchronized with other intersections to create a continuous green path, where possible, to further reduce travel time.

6.7 Rerouting the Ambulance

In cases where traffic congestion is so severe that prioritizing the current lane isn’t enough to clear the way for the ambulance, the system implements rerouting strategies. This involves calculating alternative paths for the ambulance based on real-time traffic data. The system uses the information gathered from side cameras and traffic sensors to analyse congestion levels across multiple routes. If a clear route is available, the system dynamically adjusts the ambulance’s path to ensure it can avoid traffic bottlenecks and reach its destination more efficiently. The rerouting information is communicated to the ambulance driver through the system's interface, providing instructions on the new route. The system continuously tracks the ambulance’s position and makes further rerouting suggestions if necessary, ensuring that the ambulance is always on the fastest possible path.

6.8 Notification System for Traffic Authorities

An essential feature of the proposed system is the real-time notification system that alerts traffic authorities and emergency services about the status of the ambulance and traffic conditions. Once the ambulance is detected and actions such as lane prioritization and rerouting are initiated, the system sends automatic notifications to traffic control centers and local emergency services. These notifications include information about the ambulance’s location, the status of traffic signals, and any potential issues that may arise, such as continued congestion or delays. This ensures that authorities are informed in real-time and can take further actions, such as deploying additional resources, coordinating with other traffic control systems, or assisting with road clearance. The notification system enhances communication between traffic management and emergency services, promoting faster response times.

6.9 Graphical User Interface (GUI)

The system includes a user-friendly graphical interface that provides traffic authorities and emergency responders with real-time updates on the ambulance's status and the surrounding traffic conditions. The GUI displays video feeds from the cameras, showing the ambulance’s location and the current state of the road. Key information such as the detected ambulance’s lane, nearby congestion, and the status of traffic signals is displayed clearly. Flashing alerts and color-coded icons highlight critical information, allowing traffic controllers to quickly assess the situation and make informed decisions. The interface also includes features to adjust traffic signals manually or override automatic decisions if necessary. Furthermore, the GUI stores historical data that can be used for post-incident analysis, providing insights into traffic flow patterns and the system's performance during emergency situations.

6.10 Continuous Monitoring and Adaptation

To ensure the system remains efficient over time, continuous monitoring is conducted to assess its performance. The system adapts to changing traffic patterns by using machine learning algorithms that learn from past events. The data gathered during real-time operation is analysed to improve the system's predictions and decision- making processes, ensuring that it becomes more accurate and reliable with use. The methodology emphasizes adaptive learning and optimization, allowing the system to handle different traffic conditions, time of day, and emergency situations, making it a scalable solution for urban environments. The methodology for the ambulance prioritization system is based on advanced computer vision, machine learning, and real-time traffic management algorithms. By combining these technologies, the system ensures that ambulances can navigate congested urban areas with minimal delays, ultimately saving lives by improving emergency response times.

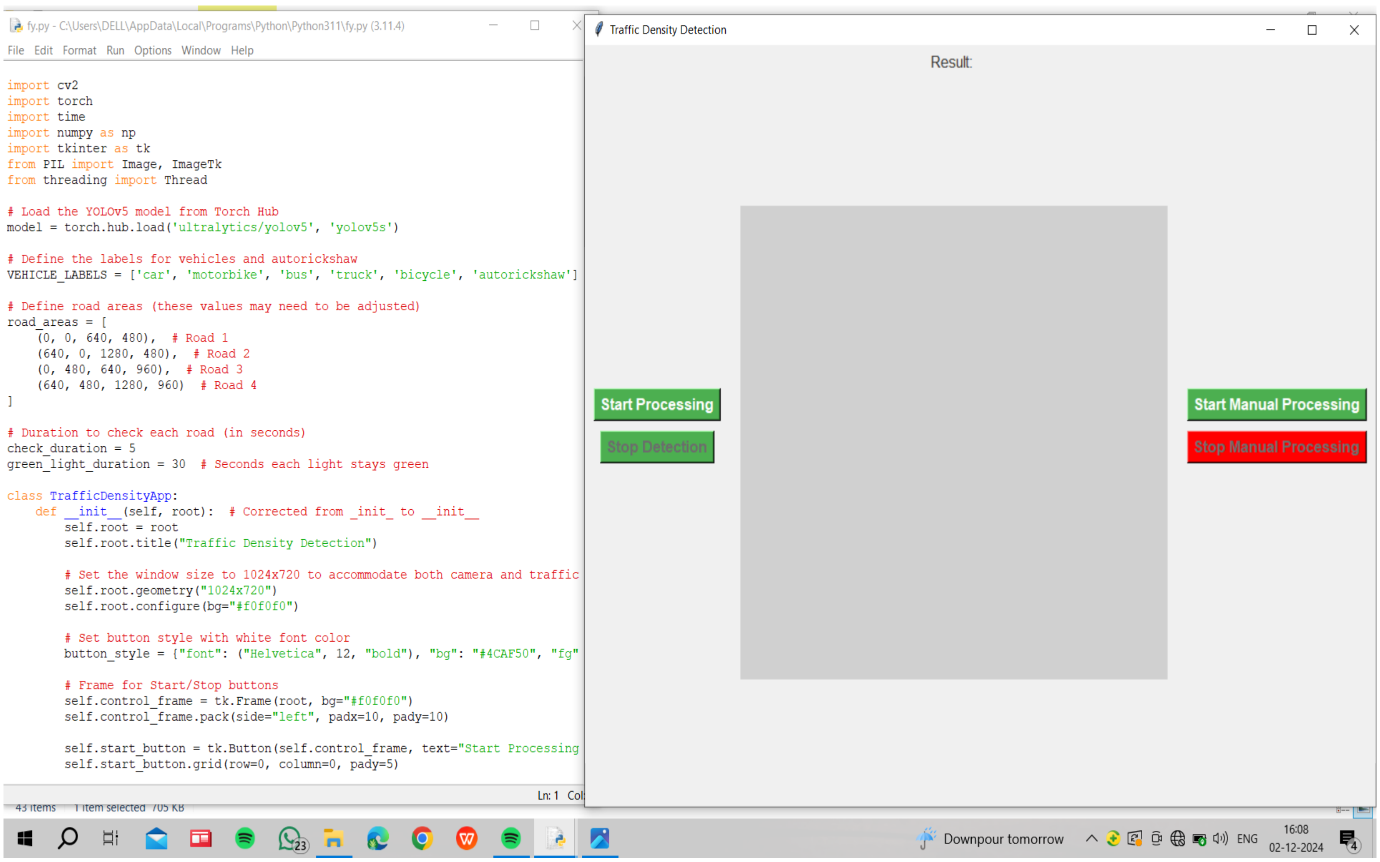

FIG 6.1 Simulation Overview

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The AI-powered vehicle detection system transforms emergency response in urban environments. Leveraging YOLOv5 and multi-camera surveillance, it detects ambulances in real-time, adjusting traffic signals and prioritizing lanes. The system reduces delays, enhances traffic flow, and public safety. Automated alerts and integration with traffic infrastructure ensure coordinated emergency response. This innovation has far-reaching potential, saving lives and improving emergency services. It sets a foundation for future traffic management innovations. The system can be extended to other emergency vehicles. Urban populations will benefit from reduced response times. This solution addresses traffic congestion challenges. It contributes to developing safer, smarter cities. This project also highlights the potential for broader applications in smart cities, where AI, computer vision, and IoT technologies can work together to create more responsive, efficient urban systems. While this system focuses on ambulance prioritization, the same framework can be extended to other types of emergency vehicles, further improving the safety and efficiency of urban traffic management. The impact of this system is far-reaching, offering the potential to save lives, reduce traffic-related fatalities, and improve the overall quality of emergency services in urban areas. By addressing one of the most pressing issues in emergency response, the system sets a foundation for future innovations in traffic management and emergency services, contributing to the development of safer, smarter cities. As urban populations continue to grow and traffic congestion becomes increasingly prevalent, solutions like this will be essential in ensuring that emergency vehicles can navigate city streets efficiently, reducing response times and ultimately saving lives.

7.2 FUTURE ENHANCEMENT

The traffic monitoring system has potential enhancements to improve functionality, accuracy, and urban traffic management. Integrating AI and Machine Learning algorithms can enhance predictive analytics, forecasting traffic patterns and congestion points. IoT device integration can collect broader data, improving accuracy and situational awareness. Enhanced user interfaces and data visualization tools, like AR/VR, can facilitate quicker decision-making. Mobile apps can provide real-time traffic information, promoting informed commuting decisions. Sustainability efforts can include eco-friendly technologies, electric vehicle monitoring, and charging station locations. These enhancements can lead to smoother traffic flow, reduced congestion, and improved public engagement. Future developments can integrate smart city initiatives, improving overall efficiency. Real-time data sharing and analysis can optimize traffic signal control and emergency response. These advancements can create a smarter, more sustainable transportation ecosystem.

APPENDIX A

Source code

```
import cv2

import torch

import time

import numpy as np

import tkinter as tk

from PIL import Image, ImageTk

from threading import Thread


# Load the YOLOv5 model from Torch Hub

model = torch.hub.load('ultralytics/yolov5', 'yolov5s')


# Define the labels for vehicles and autorickshaw

VEHICLE_LABELS = ['car', 'motorbike', 'bus', 'truck', 'bicycle', 'autorickshaw']


# Define road areas (these values may need to be adjusted)

road_areas = [

    (0, 0, 640, 480), # Road 1

    (640, 0, 1280, 480), # Road 2

    (0, 480, 640, 960), # Road 3

    (640, 480, 1280, 960) # Road 4
```

```
]

# Duration to check each road (in seconds)
check_duration = 5
green_light_duration = 3 # Seconds each light stays green

class TrafficDensityApp:
    def __init_(self, root):
self.root = root
self.root.title("Traffic Density Detection")

        # Set the window size to 1024x720 to accommodate both camera and traffic
lights
self.root.geometry("1024x720")
self.root.configure(bg="#f0f0f0")

        # Set button style
button_style = {"font": ("Helvetica", 12, "bold"), "bg": "#4CAF50", "fg":
"white", "relief": "raised"}
self.start_button = tk.Button(root, text="Start Processing",
command=self.start_detection, **button_style)
self.start_button.pack(pady=5)
```

```
self.manual_start_button = tk.Button(root, text="Start Manual Processing",
command=self.start_manual_processing, **button_style)
self.manual_start_button.pack(pady=5)

self.manual_stop_button = tk.Button(root, text="Stop Manual Processing",
command=self.stop_manual_processing, state=tk.DISABLED,
font=("Helvetica", 12, "bold"), bg="red", fg="white", relief="raised")
self.manual_stop_button.pack(pady=5)

self.stop_button = tk.Button(root, text="Stop Detection",
command=self.stop_detection, state=tk.DISABLED, font=("Helvetica", 12,
"bold"), bg="red", fg="white", relief="raised")
 self.stop_button.pack(pady=5)

        # Display label for results
self.result_label = tk.Label(root, text="Result: ", font=("Helvetica", 12),
bg="#f0f0f0", fg="#333")
 self.result_label.pack(pady=5)

        # Canvas to show the camera feed with adjusted height and width
self.canvas = tk.Canvas(root, width=480, height=360, bg="#d0d0d0")
self.canvas.pack(side="left", padx=10, pady=10)
```

```
    # Canvas for traffic lights with more compact layout
self.traffic_light_canvas = tk.Canvas(root, width=250, height=400, bg="#ffffff",
highlightthickness=0)
self.traffic_light_canvas.pack(side="right", padx=10, pady=10)


    # Drawing traffic light indicators with road labels and adjusted positioning
self.traffic_lights = []
    for i in range(4):
y_position = 50 + i * 90
self.traffic_light_canvas.create_text(125, y_position - 15, text=f"Road {i + 1}",
font=("Helvetica", 10, "bold"), fill="#333")
red_light = self.traffic_light_canvas.create_oval(100, y_position, 150,
y_position + 40, fill="red")
green_light = self.traffic_light_canvas.create_oval(160, y_position, 210,
y_position + 40, fill="gray")
self.traffic_lights.append((red_light, green_light))


self.running = False
self.cap = None
self.manual_running = False


  def update_traffic_lights(self, active_index):
    for i, (red_light, green_light) in enumerate(self.traffic_lights):
```

```
            if i == active_index:
self.traffic_light_canvas.itemconfig(red_light, fill="gray")
self.traffic_light_canvas.itemconfig(green_light, fill="green")
            else:
self.traffic_light_canvas.itemconfig(red_light,  fill="red")
self.traffic_light_canvas.itemconfig(green_light, fill="gray")
self.root.update_idletasks()


    def reset_traffic_lights(self):
        for red_light, green_light in self.traffic_lights:
self.traffic_light_canvas.itemconfig(red_light, fill="red")
self.traffic_light_canvas.itemconfig(green_light, fill="gray")
self.root.update_idletasks()


    def start_detection(self):
self.start_button.config(state=tk.DISABLED)
self.manual_start_button.config(state=tk.DISABLED)
self.stop_button.config(state=tk.NORMAL)
self.running = True


camera_thread = Thread(target=self.detect_traffic_density)
camera_thread.start()
```

```
    def stop_detection(self):
self.running = False
        if self.cap:
self.cap.release()
self.start_button.config(state=tk.NORMAL)
self.manual_start_button.config(state=tk.NORMAL)
self.stop_button.config(state=tk.DISABLED)
self.reset_traffic_lights()

    def  start_manual_processing(self):
        if not self.manual_running:
self.manual_running = True
self.manual_start_button.config(state=tk.DISABLED)
self.manual_stop_button.config(state=tk.NORMAL)
manual_thread  =  Thread(target=self.manual_processing)
manual_thread.start()

    def stop_manual_processing(self):
self.manual_running = False
self.manual_start_button.config(state=tk.NORMAL)
self.manual_stop_button.config(state=tk.DISABLED)

    def manual_processing(self):
```

```
    while self.manual_running:
      for i in range(4):
        if not self.manual_running:
          break
self.update_traffic_lights(i)
time.sleep(green_light_duration)
self.reset_traffic_lights()

  def detect_traffic_density(self):
self.cap = cv2.VideoCapture(0)
road_vehicle_counts = [0] * len(road_areas)

    while self.running:
      for i in range(len(road_areas)):
        if not self.running:
          break

start_time = time.time()
frame_counts = []
saved_frame = None

        while time.time() - start_time<check_duration and self.running:
          ret, frame = self.cap.read()
```

```
        if not ret:
print("Failed to capture video frame")
          break


rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        results = model(rgb_frame)
        detections = results.pandas().xyxy[0].to_dict(orient="records")


vehicle_count = 0
        for detection in detections:
          if detection['name'] in VEHICLE_LABELS:
vehicle_count += 1
frame_counts.append(vehicle_count)


        for detection in detections:
          if detection['name'] in VEHICLE_LABELS:
xmin, ymin, xmax, ymax = int(detection['xmin']), int(detection['ymin']),
int(detection['xmax']), int(detection['ymax'])
            cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0),
2)
            label = f"{detection['name']} {int(detection['confidence'] *
100)}%"
```

```
            cv2.putText(frame, label, (xmin, ymin - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)

          x1, y1, x2, y2 = road_areas[i]
          cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)

          cv2.putText(frame, f"Road: {i + 1}", (20, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
average_count = int(np.mean(frame_counts))
          cv2.putText(frame, f"Avg Count: {average_count}", (20, 70),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)

img = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
imgtk = ImageTk.PhotoImage(image=img)
self.canvas.create_image(0, 0, anchor=tk.NW, image=imgtk)
self.root.update_idletasks()
self.root.update()

saved_frame = frame.copy()

          if not self.running:
            break
```

```
road_vehicle_counts[i] = average_count

        if saved_frame is not None:
          filename = f"road_{i + 1}_detected.jpg"
          cv2.imwrite(filename, saved_frame)

self.display_results(road_vehicle_counts)
self.update_traffic_lights_based_on_counts(road_vehicle_counts)
time.sleep(1)

  def update_traffic_lights_based_on_counts(self, road_vehicle_counts):
sorted_indices = sorted(range(len(road_vehicle_counts)), key=lambda i:
road_vehicle_counts[i], reverse=True)

    for i in range(len(sorted_indices)):
self.update_traffic_lights(sorted_indices[i])
time.sleep(green_light_duration)

  def display_results(self, road_vehicle_counts):
result_text = ", ".join([f"Road {i+1}: {count} vehicles" for i, count in
enumerate(road_vehicle_counts)])
self.result_label.config(text=f"Result: {result_text}")
```

```python
# Initialize Tkinter application
root = tk.Tk()
app = TrafficDensityApp(root)
root.mainloop()
```

APPENDIX B

OUTPUT SCREENSHOTS

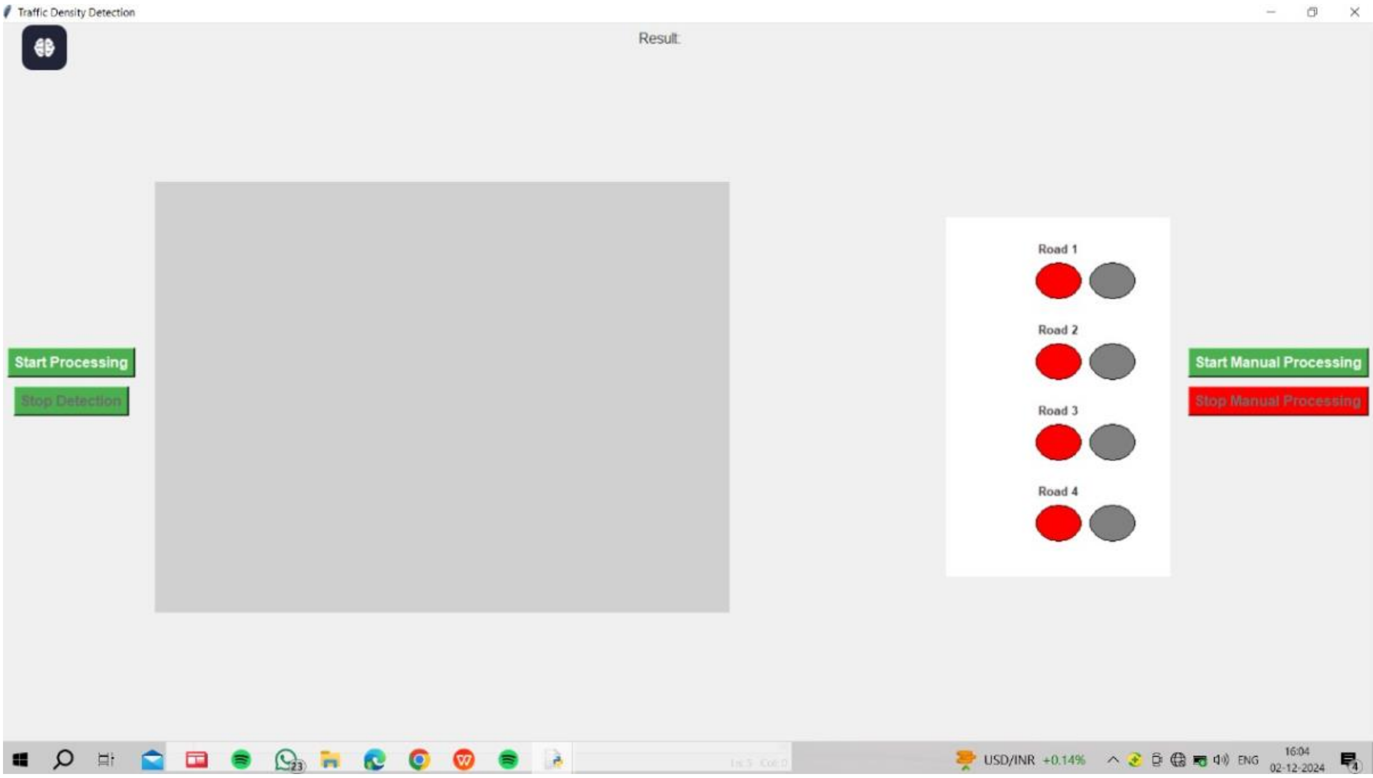

FIG B.1 Traffic GUI Interface

MANUAL TRAFFIC SIGNAL WORKING

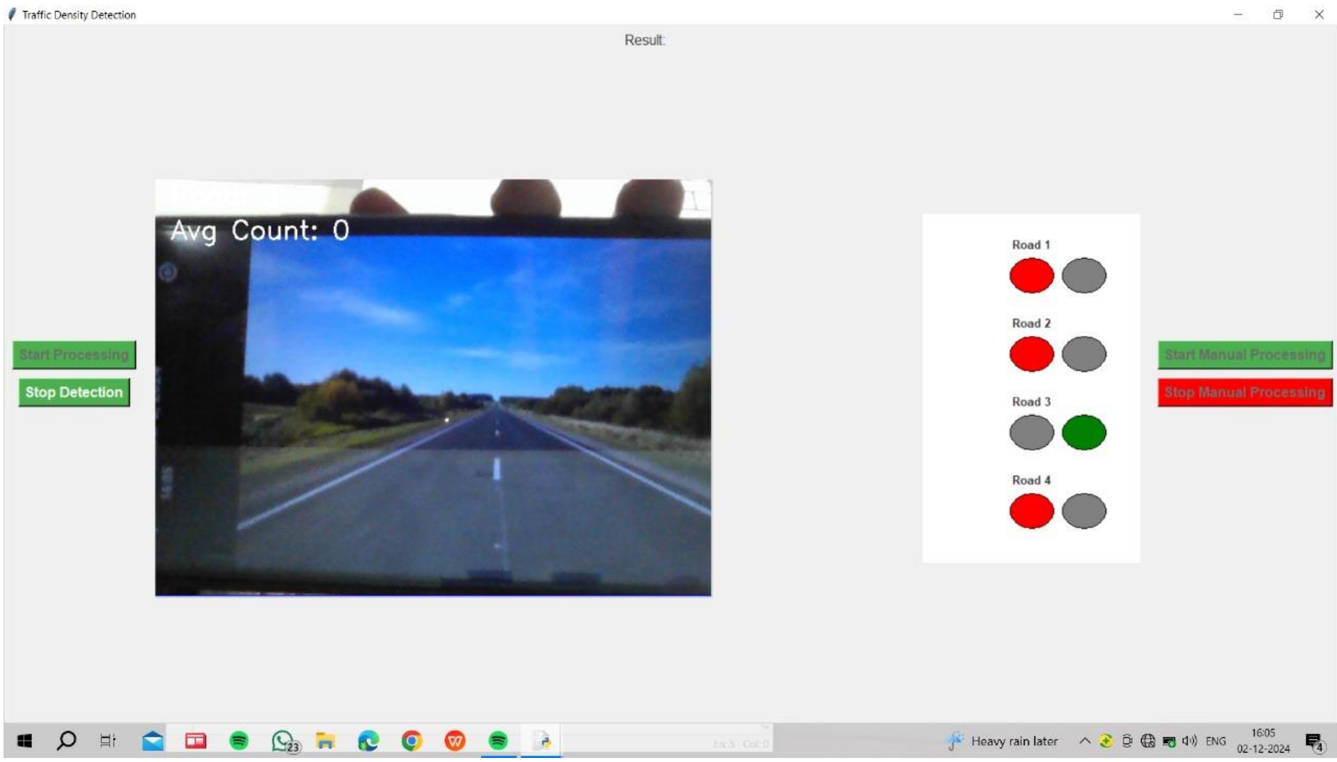

FIG B.2 No Vehicles In Lane

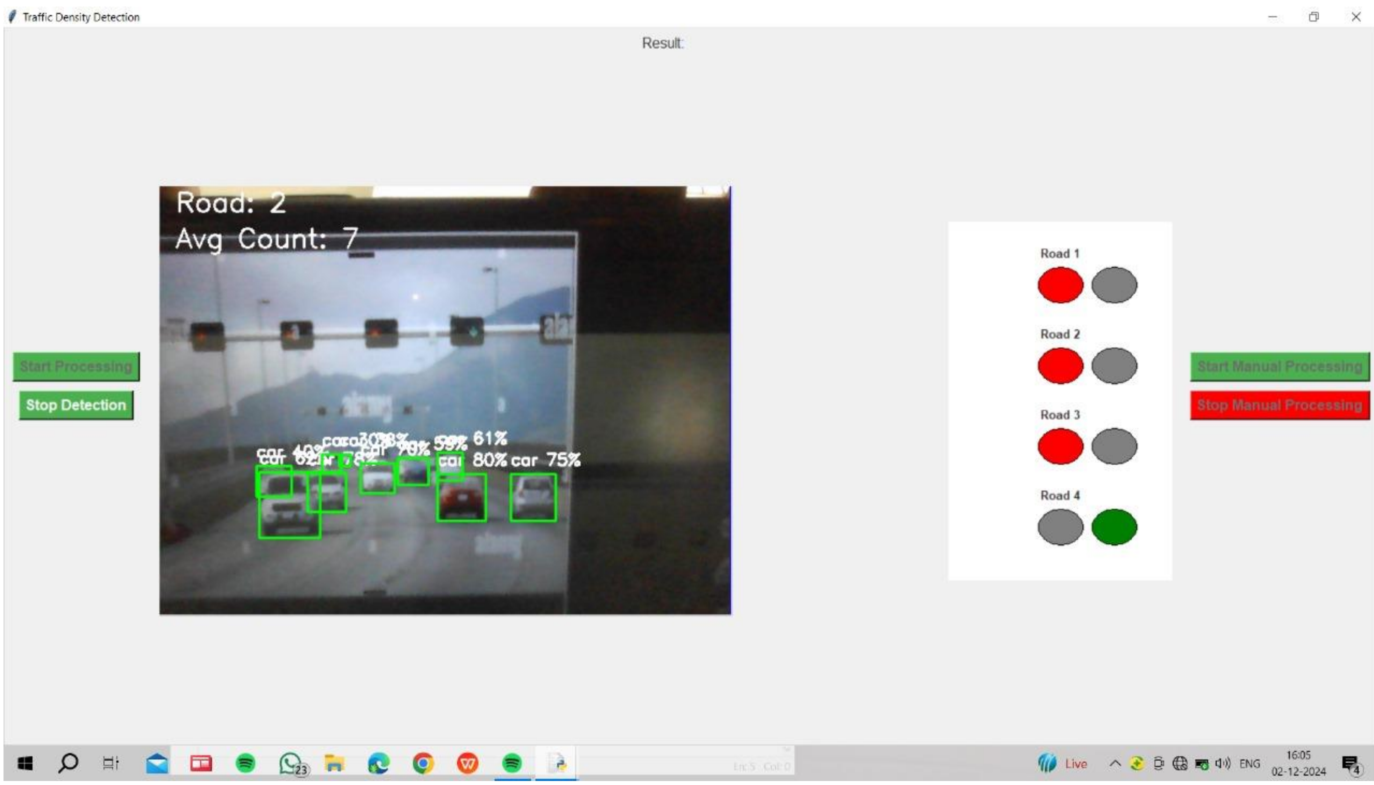

FIG B.3 Vehicle Detection

REFERENCES

1. Tewodros Syum Gebre, Leila Hashemi-beni , Eden Tsehaye Wasehun,and Freda Elikem Dorbu , "AI-Integrated Traffic Information System:A Synergistic Approach of Physics Informed Neural Network and GPT-4 for Traffic Estimation and Real-Time Assistance"15 May 2024.
2. Abdulrahman Alruban, Hanan Abdullah Mengash, Majdy M. Eltahir ,Nabil Sharaf Almalki, Ahmed Mahmud, And Mohammed Assiri "Artificial Hummingbird Optimization Algorithm With Hierarchical Deep Learning for Traffic Management in Intelligent Transportation Systems",6 February 2024.
3. RuijieZhu ,Shuning Wu, , Lulu Li,PingLv, , Mingliang Xu "Context-Aware Multiagent Broad Reinforcement Learning for Mixed Pedestrian-Vehicle Adaptive Traffic Light Control" ,15 OCTOBER 2022.
4. Neetesh Kumar, Sarthak Mittal, Vaibhav Garg, Neeraj Kumar,"Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System", MARCH 2022.
5. Anakhi Hazarika, Nikumani Choudhury, Moustafa M. Nasralla, Sohaib Bin Altaf Khattak, And Ikram Ur Rehman,“ Edge ML Technique for Smart Traffic Management in Intelligent Transportation Systems",22 February 2024.
6. Ossama Younis , Nader Moayeri,” Employing Cyber-Physical Systems: Dynamic Traffic Light Control at Road Intersections.” 2017.

7. QinaatHussain , Wael Alhajyaseen , Kris Brijs, Ali Pirdavani , and Tom Brijs. “Improved Traffic Flow Efficiency During Yellow Interval at Signalized Intersections Using a Smart Countdown System”, 2022

8. Alessandro Casavola, Giuseppe Franzè ,Gianfranco Gagliardi, and Francesco Tedesco, Improving Lighting Efficiency for Traffic Road Networks: A Reputation Mechanism-Based Approac,2022.

9. M. Mansouri and M. Barati. Cloud-Based Traffic Monitoring System: A Comprehensive Overview. International Journal of Information Technology and Computer Science, 11(9):29–36, 2019.