SHAPING MOBILITY ON ROADS USING DYNAMIC PRIORITY SCHEDULING

Capstone Project Report

END SEMESTER EVALUATION

Submitted by:

(101903113) JAPLEEN KAUR

(101903125) SHAURYA PRATAP SINGH BHADAURIA

(101903198) PRAMIT DEEP KAUR GOGNA

(101953006) APURVI GARG

BE Fourth Year, COE

CPG No: 121

Under the Mentorship of

Dr. Tarunpreet Bhatia

Assistant Professor

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

December 2022

ABSTRACT

Traffic congestion is a serious issue, not just affecting the citizens, but also reducing the interests of the business activities. Failure of signals, poor law enforcement and bad traffic management leads to traffic congestion. Traffic jams not only cause extra delay and stress for the drivers, but also increase fuel consumption and air pollution. According to the TomTom Traffic Index [1], 3 of the top 10 cities facing the most traffic congestion in India are Mumbai, Bengaluru, and New Delhi. People are compelled to spend hours stuck in traffic jams, wasting away their precious time commuting.

This project aims at overcoming the problem of traffic congestion by developing a Smart Traffic Management System (STMS) which would save a lot of time, will help in reduction of mental stress and consumption of non-renewable resources. Our proposed system aims to utilize live images from the cameras at traffic junctions for traffic density calculation using image processing and AI. It also focuses on the algorithm for scheduling the timing of traffic lights based on the vehicle density in a particular lane to reduce congestion, thereby providing faster transit to people. The system also prioritizes emergency vehicles in a lane and schedules more time for them to pass as soon as possible. The project consists of 3 parts namely Vehicle Detection and Density Calculation Module, Prioritization and Signal Switching Scheduling Algorithm and Simulation module that helps to simulate the results and change the timers.

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled Shaping Mobility on Roads using Dynamic Priority Scheduling is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr. Tarunpreet Bhatia during 6th and 7th semester (2022).

Date: 19 December, 2022

Roll No.	**Name**	**Signature**
101903198	Pramit Deep Kaur Gogna	
101903125	Shaurya Pratap Singh Bhadauria	
101953006	Apurvi Garg	
101903113	Japleen Kaur	

Counter Signed By:

Mentor:
Dr. Tarunpreet Bhatia
Assisstant Professor
Computer Science & Engineering Department
TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our thanks to our mentor Dr. Tarunpreet Bhatia. She has been of great help in our venture, and an indispensable resource of technical knowledge. She is truly an amazing mentor to have.

We are also thankful to Mr. Maninder Singh, Head of Computer Science and Engineering Department, the entire faculty and staff of the Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards the successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement. They always wanted the best for us and we admire their determination and sacrifice.

Date: 19 December, 2022

Roll No.	**Name**	**Signature**
101903198	Pramit Deep Kaur Gogna	
101903125	Shaurya Pratap Singh Bhadauria	
101953006	Apurvi Garg	
101903113	Japleen Kaur	

Counter Signed By:

Mentor:
Dr. Tarunpreet Bhatia
Assisstant Professor
Computer Science & Engineering Department
TIET, Patiala

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATION

AI	Artificial Intelligence
DFD	Data Flow Diagram
GPS	Global Positioning System
IEEE	Institute for Electrical and Electronics Engineers
LEDs	Light Emitting Diodes
SMRUDPS	Shaping Mobility on Roads Using Dynamic Priority Scheduling
UML	Unified Modelling Language
YOLO	You Only Look Once

INTRODUCTION

1.1 Project Overview

Congestion in traffic is a difficult issue, influencing the residents, yet in addition lessening the interests of the business exercises. Disappointment of signs, unfortunate policing awful traffic the executives prompt gridlock. Gridlocks not just goal additional deferral and stress for the drivers, yet in addition increment fuel utilization and air contamination. According to the TomTom Traffic Index [1], 3 of the top 10 cities facing the most traffic congestion is in India viz. Mumbai, Bengaluru, and New Delhi. People are compelled to spend hours stuck in traffic jams, wasting away their precious time commuting.

Hence to overcome the problem of congestion we aimed at developing a Shaping Mobility on Roads Using Dynamic Priority Scheduling (SMRUDPS) which would save a lot of time, will helpin reduction of mental stress and consumption of non-renewable resources. Our proposed framework expects to use live pictures from the cameras at traffic intersections for traffic density computation utilizing picture handling and simulated intelligence. It additionally centers around the calculation for exchanging the traffic signals in view of the vehicle density to decrease obstruction, in this manner giving quicker travel to individuals. The system is also capable of prioritizing emergency vehicles using YOLO algorithm.

The project consists of 2 parts:

1. **Vehicle Detection Module and Density Calculation:** This module is responsible for detecting the number of vehicles in each lane. A video feed containing traffic congestion of four lanes will be provided as the input to this module, then density calculation based on the area occupied by the vehicles will be done and based on the density calculation results, further modules will process the timings of the signals.

2. **Simulation Module**: A simulation is developed using Pygame library to simulate traffic signals and vehicles moving across a traffic intersection. The video feed of vehicles while intersecting the lanes are taken as the input feed, the density of each lane is calculated on the basis of area occupied by the various classes of vehicles, the lane with the highest traffic congestion will be preempted first and hence reducing the congestion for the busiest lane.

1.2 Need Analysis

With the world progressing faster than ever in terms of technology, market, & economy, we face new modern world problems. One of the most relevant and popularmodern world problems is the progressively increasing traffic congestion on the roadsof our country.

Considering the fact India's road network (including national highways etc.) has grown by just about a third in the last decade whereas vehicle registrations have increased by almost three times. Thus, vehicle density is increasing at a much faster pace than road length - obviously, congestion will be higher. Analyzing, the latestdata from the government's VAHAN dashboard [12] (as shown in Fig 1), in January 2019, more than a million-and-half (1,607,315) new vehicles were bought and registered with various regional transport offices across the country. That comes to an average of more than 51,000 new vehicles bought across India, every single day of January. This constant increase in congestion is starting to fail the traditional traffic

management system as the traditional traffic management system has its drawbacks such as there is no real time condition-based management and during signals breakdown, there are serious and wide-spread traffic difficulties during peak hours. This raises the need for a more intelligent, flexible and efficient smart traffic management system.

FIGURE 1: VAHAN Dashboard Data of 2022 [12]

At times this traffic congestion leads to severe or even life-threatening situations. Consider the scenario of an ambulance stuck in congestion where even a bit of delay can be really fatal. According to a report published by Times of India about 146,133 people were killed in road accidents in India in the year 2016. The National Crime Records Bureau (NCRB) reports that there were 4,03,116 reported cases of road accidents in 2021, up from 3,54,796 in 2020. There has been a 16.8% increase in road accident fatalities, rising from 1,33,201 in 2020 to 1,55,622 in 2021. Unfortunately, about 30% of deaths are caused due to delayed ambulances. Another Indian government data shows that more than 50% of heart attack cases reach hospital late, which can constitute unavailability of ambulances too but majority of it is due to patients stuckin traffic. Hence, the delay of ambulances is a really severe problem which needs to be addressed with a proper solution.

Existing conventional methods face many drawbacks. The manual controlling system requires an outsized amount of manpower. We need a better system to regulate the traffic. Static traffic controlling purposes a traffic light with a clock, which is consistent and doesn't adjust to the continuous traffic fluctuations. While utilizing electronic sensors i.e., proximity sensors or loop detectors, the exactness and inclusion are in struggle on the grounds that the social occasion of excellent data requires costly advancements, and a restricted spending plan will lessen the quantity of offices. Sensors have a restricted area of inclusion and to cover the entire region for the most part requires lots of sensors.

1.3 Research Gaps

Following are some points for research gaps in the existing system:

1. The conventional traffic systems depend on the timing and hence are nonlinear and complex in nature. This system is traffic dependent and uses the traffic density.

2. An advancement to the traditional model of traffic control is addition of some proximity sensors on the road. This sensor gives the data about the traffic on the road and according to the sensor the traffic signals are controlled. But sensors will add up to the cost and hence installation of cameras will be an economic way out.

3. Insufficient capacity, unrestrained demand and large red-light delays are types of traffic congestion faced in the mega cities. Hence, the need of simulating and advancing traffic light to more readily oblige this rising interest emerges Using the video monitoring and surveillance systems for traffic density estimation and vehicle classification can also be used achieved using video monitoring systems.

1.4 Problem Definition and Scope

There are a total of 5 cities from India in the top 20 cities with highest traffic congestion in the world. With Mumbai being 5th most congested city in terms of traffic followed by Bengaluru at 10th, Delhi at 11th and Pune at 21st world rank in traffic congestion according to latest data [1]. This shows the need for an evolved and smart traffic management system.

A traffic light system that monitors real time traffic congestion density and based on the monitored data, schedules the timing of the traffic light to optimally minimize the

traffic congestion on the road. Also, we tend to design the system to provide special priority to public service vehicles such as ambulances in order to reduce the delay in these critical life-death situations.

We believe the introduction of this system can make the traffic management really efficient, reduce delays for ambulances and even contribute to preventing prolonged fuel consumption at the traffic lights.

1.5 Assumptions and Constraints

S. No	Assumptions
1	It is assumed that the area of each lane is uniform in order to calculate the density.This will reduce the complexity of the algorithm to calculate road density.
2	The dimensions of the vehicles to be monitored are at least 4-5% of the scene size. It means that the vehicle length should be around 60 pixels on 1280px X 720px scene for the simulation module.
3	It is assumed that the vehicles on the road are not blocked by obstacles thatinterfere with the detection of them.
4	It is assumed that the lens quality of the camera is good and does not have anydistortions due to flare or reflection, scratches, or cause any barrel distortion, etc.

S. No	Constraints
1	The vehicle detection in video is performed only in proper lightconditions and not in dark.
2	The camera which captures the lanes of the road must be of good resolution.
3	All types of vehicles belonging to same class are considered to occupy same area.
4	Traffic lights should be in proper working conditions.

1.6 Standards

ISO/IEC 9126

This standard deals with the following aspects to determine the quality of a software application:

- **Quality model**: The quality model groups programming quality in an organized arrangement of qualities and sub-qualities.
- **External metrics**: External metrics are applicable to running software.
- **Internal metrics**: Internal metrics are those which do not rely on software execution (static measure).
- **Quality in use metrics**: Quality-being used measurements are just accessible when the eventual outcome is utilized in genuine circumstances. Preferably, the interior quality decides the outside quality and outer quality decides quality being used.

This standard presents some set of quality attributes for any software such as:

- **Functionality**: A bunch of characteristics that bear on the presence of a bunch of capabilities and their predetermined properties. The capabilities are those that fulfill expressed or inferred needs.
- **Reliability**: A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.
- **Usability**: A bunch of properties that bear on the work required for use, and on the singular evaluation of such use, by an expressed or inferred set of clients.
- **Efficiency**: A bunch of qualities that bear on the connection between the degree of execution of the product and the quantity of assets utilized, under expressed conditions.
- **Maintainability**: A set of attributes that bear on the effort needed to make specified modifications.

- **Portability**: A set of attributes that bear on the ability of software to betransferred from one environment to another.

830-1998 - IEEE Recommended Practice for Software Requirements Specifications

Replaced by ISO/IEC/IEEE 29148:2011. The content and qualities of a good software requirements specification (SRS) are described and several sample SRS outlines are presented. This recommended practice is aimed at specifying requirements of software to be developed but also can be applied to assist in the selection of in-house and commercial software products. Guidelines for compliance with IEEE/ EIA 12207.1- 1997 are also provided.

1.7 Approved Objectives

The objectives of our project are as follows:

1. To identify and classify various vehicles using object detection algorithms.
2. To develop a scheduling algorithm which controls the traffic light timings based on traffic density and prioritization of emergency vehicles on each lane.
3. To create a simulated environment of a smart traffic control system.

1.8 Methodology

Our intelligent traffic management system monitors the traffic lanes to identify the area with high traffic congestion and in accordance with it, the system schedules the traffic flow accordingly and hence reduces congestion. Our system consists of various sub-modules such as vehicle detection program, signal scheduling algorithm on the basis of congestion traced along with priority function for public service emergency vehicles and finally simulation on Pygame to display complete methodology. The system would work in the following steps:

1. **Vehicle Detecting Algorithm:** The YOLOv3 calculation is fit for exact item location (traffic members) with close to constant execution. This calculation begins with extricating a solitary picture from the video transfer, in the subsequent stage the separated picture is resized to 416x416 that addresses contribution to the just go for its organization. Darkflow is used to detect vehicles in this case. Here, vehicles being detected are: car,bus, bike, rickshaw and truck.

2. **Priority vehicles detection and signal switching algorithm:** The traffic system has to give more priority to emergency situations. In the proposed system, the vehicles are sensed and processed through imaging techniques. A web camera is placed with the traffic light to capture image frames.

Pictures extricated from the video are then dissected to recognize and count vehicles. Then relying upon the sign cycle, time is assigned to every path.

3. **Pygame Simulation:** We tried to simulate the whole process via Pygame for real-life traffic issues (Pygame is a cross-platform set of Python modules designed for writing video games). It contains a 4-way intersection with 4 traffic signals. Each signal has a timer on top of it, which shows the time remaining for the signal to switchfrom green to yellow, yellow to red, or red to green. Pygame is highly portable and runs on nearly every platform and operating system.

Fig. 2 is the block diagram of the of the proposed system:

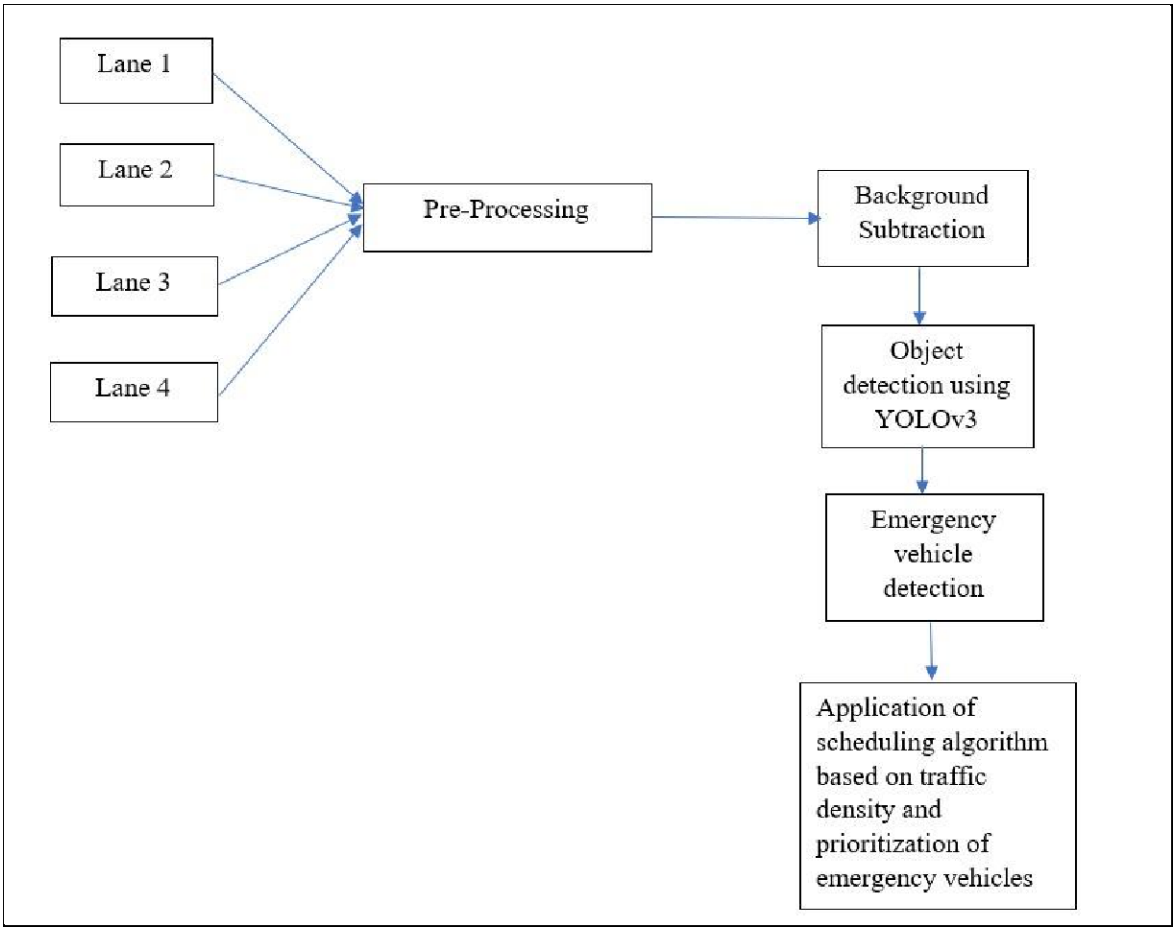


FIGURE 2: Block Diagram of the proposed system

1.9 Project Outcomes and Deliverables

- The smart traffic control system would manage the traffic based on real time situation of the lanes thus enabling it to manage traffic uncertainties.
- It would reduce the waiting time of emergency vehicles like ambulances, fire brigades etc.
- This would comfort people and help them fight distress caused by longwaiting times.

1.10 Novelty of Work

The project is mostly a bag of borrowings that tends to fulfil research gaps given out by other projects. The uniqueness of this project lies in the fact that it incorporates emergency vehicles (ambulances, fire brigade trucks, law reinforcement vehicles) as well along with the detection and density estimation of vehicles using computer vision (image processing). The scheduling algorithm not only schedules timing based on the density of lanes but also considers the priority of that particular lane due to the presence of emergency vehicle in it. This project involves video processing through camera rather than sensor-based vehicle detection and density estimation.

REQUIREMENT ANALYSIS

2.1 Literature Survey

2.1.1 Theory Associated with Problem Area

One of the most relevant and popular modern world problems is the progressively increasing traffic congestion on the roads of our country. Vehicle density is increasing at a much faster pace than road length. The constant increase in congestion is starting to fail the traditional traffic management system as the traditional traffic management system has its drawbacks such as there is no real time condition-based management and during signals breakdown, there are serious and wide-spread traffic difficulties during peak hours. This raises the need for a more intelligent, flexible and efficient smart traffic management system. Further, our efficient system also contributes to help solve issues like help reduce air pollution & preventing prolonged diesel/petrol consumption at the traffic lights. Traffic congestion is a severe difficulty, now not just affecting the residents, however additionally reducing the pastimes of the enterprise sports. Failure of alerts, bad law enforcement and terrible visitors' management leads to traffic congestion. Traffic jams no longer handiest motive more delay and pressure for the drivers, however also boom gasoline intake and air pollution.

A traffic light system that monitors real time traffic congestion density and based on the monitored data, schedules the timing of the traffic light to optimally minimize the traffic congestion on the road. There is a need for a system that could provide special priority to public service vehicles such as ambulances in order to reduce the delay in these critical life-death situations. Hence to overcome the problem of congestion we aimed at developing a Smart Traffic Management System (STMS) which would save a lot of time, will help in reduction of mental stress and consumption of non- renewable resources. Thus, making traffic management really efficient, reduce delays for ambulances and even contribute to preventing prolonged fuel consumption at the traffic lights.

2.1.2 Existing Systems and Solutions

Though a lot of Indian cities have already implemented smart traffic management system, but not with all the features which are included in our project, let's look at some of the strategies used by various Indian cities to control traffic congestion:

- In 2019, Bhubaneswar, more than a year after chief minister Naveen Patnaik inaugurated the adaptive traffic signal system [13] (ATSC) under the ambitious Smart City programme here, the commissioner police have decided to implement the technology along the busy KIIT-Jayadev Vihar route.
- Zero-Sum Ltd. a Japanese firm with expertise in ITS (Intelligent Traffic System) along with Ahmedabad Municipal Corporation launches its pilot project on real-time traffic information in the city of Ahmedabad [14], the first ITS Solution withintegrated ad system in India. Zero-Sum Ltd plans to replicate its traffic system for allits future implementations in India.
- As part of Smart City project, Indore city administration is going to implement integrated intelligent traffic management system comprising advanced traffic signal management and traffic enforcement systems [15].
- Integrated Command Control Centre (ICCC) [16] for traffic management inching closer to being fully operationalised, city residents will soon witness Artificial Intelligence (AI) in action. The Adaptive Traffic Control System (ATCS), a key feature of the ICCC, that has been set up at 40 junctions with the heaviest traffic volume. The key technology used is AI. Chandigarh Smart City Limited is implementing the project through Bharat Electronics Limited. “The system will become fully operational in February 2022. Once the ICCC, which is under construction at Sector 17, is ready, the facility will become fully operational,” said Anil Garg, additional chief executive officer, CSCL.
- Advanced traffic management system [17] is composed of field equipment and sensors that collect the information on road conditions, traffic conditions, violations, incidents, and weather / environmental conditions and send it to the Traffic Management Centre (TMC) for processing. ITS provides necessary information on road, traffic, incidents, weather, and environment to road users, and alerts regarding violations and incidents to the enforcement agencies/rescue team 24 hours a day, 365 days a year, to ultimately ensure safe and smooth traffic along the Delhi-Mumbai Expressway. It is expected to halve the commute time between Delhi and Mumbai, from nearly 24 hours to 12 hours, and shorten the distance by 130 km.

2.1.3 Research Findings for Existing Literature

S.No.	Roll No.	Name	Paper Title	Tools/Technology	Findings	Citation
1	101903198	Pramit Deep Kaur	Smart Control of Traffic Light System using Image Processing	MATLAB, arduino, camera module	It is used to calculate traffic count and traffic density and accordingly changes the time of signals. It uses image processing using MATLAB. Overall, it gives accuracy of 70-80% in traffic estimation.	Khushi [3]
2	101903198	Pramit Deep Kaur	AI Powered Smart Traffic Control System for Emergency Vehicles	GPS, sensors, cloud	GPS tracking was used to detect emergency vehicles and give way to them by knowing their location retrieved from the cloud.	Kumar at al. [4]
3	101903198	Pramit Deep Kaur	Traffic Light Control and Violation Detection Using Image Processing	SVM, OpenCV	OpenCV was used to process the photos, which was then transformed to grayscale images before SVM was used. Not only does this technology assesstraffic density, but it also looks for red light violators	Soman at al. [7]

4	101953006	Apurvi Garg	Smart traffic lights switching and traffic density calculation using video processing	Camera, video and image processing	It takes live video feed from the cameras at traffic junctions for real time traffic density calculation using video and image processing. The system doesn't workupto the expectations due to lowerlight conditions, soswitching oversystem to hardcoded during night time.	Kanungo at al. [11]
5	101953006	Apurvi Garg	Smart Control of Traffic Light Using Artificial Intelligence	Image processing and Artificial intelligence.	The system aims to utilize live images from the cameras at traffic junctions for traffic density calculation using image processing and AI. It also focuses on the algorithm for switching the traffic lights based on the vehicle density to reduce congestion.	Gandhi at al. [12]

6	101953006	Apurvi Garg	Improving Traffic Light Control by Means of Fuzzy Logic	VISSIM, MATLAB, fuzzy controller	It uses a fuzzylogic-controlled traffic light to tailor the traffic scenarios. Formajor and secondary driveway sit needs two fuzzy controllers with three inputs and one output each.The controllers take information about thecurrent traffic conditions and	Vogel at al. [5]

					make appropriate decisions for the next phase in the signal switching algorithm	
7	101903125	Shaurya Pratap Singh Bhadauria	A distributed algorithm for adaptive traffic lights control	Simulation, sensors.	They address the problem of controlling traffic lights at an intersection with a spatially distributed sensor network. In this architecture, we define and evaluate through simulations an adaptive traffic light control algorithm.	Faye at al. [13]
8	101903125	Shaurya Pratap Singh Bhadauria	Smart controlling for traffic light time	Algorithm development, Image processing.	This paper develops an automatic algorithm to control traffic light time based on artificial intelligent techniques and image for cars on traffic lights, this algorithm is validated by compare its results with manual results.	Zaid at al. [14]
9	101903125	Shaurya Pratap Singh Bhadauria	Lightweight PVIDNet: A Priority Vehicles Detection Network Model Based on Deep Learning for Intelligent Traffic Lights	Deep Learning and OpenCV	This work proposes (1) a novel vehicle detection model named Priority Vehicle Image Detection Network (PVIDNet), based on YOLOV3, (2) a lightweight design strategy for the PVIDNet model using an activation function to decrease the execution time of the proposed model	Barbosa at al. [8]

10	101903113	Japleen Kaur	Research on Urban Road Traffic Congestion Charging Based on Sustainable Development	GPS, GSM (global system for mobile communication)	Urban road traffic congestion charge refers that motor vehicles which enter into some region or certain road section in certain time interval will be charged special fare.	Ye Sun [18]
11	101903113	Japleen Kaur	Traffic Congestion Prediction Using Machine Learning Techniques	ML, Signal Processing, Random Forest Regressor, Gradient Boosting Regressor	Model for traffic congestion that can predict congestion based on day, time and several weather data (e.g., temperature, humidity). algorithm	Rafed at al. [19]
12	101903113	Japleen Kaur	Analysis of Traffic Congestion Impacts of Urban Road Network under Indian Condition	Video graphic survey, license plate matching techniques.	Impact of traffic congestion was studied in various sectors like health, economy, environment and mitigation measures were also discussed.	Samal at al. [20]

					up to the expectations due to lowerlight conditions, so switching over system to hard coded during night time.	

2.1.4 Problem Identified

The traffic management systems that exist are based of conventional methods that face many drawbacks. Such as the manual controlling system requires an outsized amount of manpower. Static traffic controlling uses a traffic signal with a timer, which is constant and doesn't adapt to the real-time traffic variabilities. If we consider using electronic sensors i.e., proximity sensors or loop detectors, the accuracy and coverage are in conflict because the gathering of high-quality information requires expensive technologies, and a limited budget will reduce the number of facilities. Sensors have a limited area of coverage and to cover the whole area usually requires tons of sensors which itself doesn't seem feasible.

2.1.5 Survey of Tools and Technologies Used

- **GPS System:** The Global Positioning System (GPS) is a space-based radio-route framework comprising of a group of stars of satellites broadcasting route signals and an organization of ground stations and satellite control stations utilized for observing and control.
- **MATLAB:** MATLAB is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. This platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The core of MATLAB is the MATLAB language, a lattice-based language permitting the most normal articulation of computational math.
- **OpenCV:** OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source Apache 2 License.
- **Machine Learning & Deep Learning models:** In AI, support-vector machines (SVM) are managed learning models with related learning calculations that examine information for grouping and relapse examination.

2.2 Software Requirement Specification

The goal of this requirement specification document is to describe the necessary constraints and requirements of a project aiming to overcome the problem of traffic congestion by developing a Shaping Mobility on Roads Using Dynamic Priority Scheduling (SMRUDPS) based on the traffic density on the road.

2.2.1 Introduction

This section documents the specific requirements (functional and non-functional), performance requirements, design constraints and external interface requirements.

2.2.1.1 Purpose

This Software Requirements Specification (SRS) documents key specifications, describes a prototype in terms of functional and non-functional requirements for the Smart Traffic Management System (STMS). The information documented, helps the intended audience to design and develop the product. We will be launching an official prototype version and will make amendments in order to make the system more efficient.

2.2.1.2 Intended Audience and Reading Suggestions

The primary audience that would be impacted by the project would everyone adopting means of road transport to travel. The Smart Traffic Management System would help the traffic management authorities as well as the emergency services such as ambulance and fire brigade services. Furthermore, with the introduction of efficiency in the traffic management, it would save more fuel and hence would contribute to the environment as well as people on a daily basis.

2.2.1.3 Project Scope

The scope of the project is to introduce efficiency in the traffic management domain and further prevent delay of various emergency vehicles in critical life-death situations. The Shaping Mobility on Roads Using Dynamic Priority Scheduling achieves its aim of efficient management by performing vehicle detection and then using the results to calculatethe traffic occupancy of each lane of traffic light and using these traffic occupancy values to set the priority and corresponding timings for each lane. This system is primarily for traffic lights situated at an intersection of multiple lanes/roads.

2.2.2 Overall Description

2.2.2.1 Product Perspective

Traffic congestion is a major problem faced by every individual nowadays. The extent of congestion is more for extremely populated Indian cities like Mumbai, Bangalore, Delhi etc. Traffic jams not only cause extra delay and stress for the drivers, but also increase fuel consumption and air pollution. Hence, we propose a system for shaping mobility on roads to reduce traffic congestion in a much more efficient manner. The main aim of the project is to develop a traffic management system, which will reduce the timings of each lane, while providing a prioritized pre-emption for emergency vehicles.

The above-mentioned strategies will be implemented using camera, Arduino7. The video feed will be provided in the form of input to the camera, the vehicle detection module will then calculate the density for each lane, the information will then be set as an input for the scheduler which will perform prioritized pre-emption, and finally the change in timings will be controlled by the timing controller.

For the software implementation of the whole system, Pygame library is used to simulate traffic signals and vehicles moving across a traffic intersection. The video feed of vehicles while intersecting the lanes are taken as the input feed, the density of each lane is calculated on the basis of area occupied by the various classes of vehicles,the lane with the highest traffic congestion or with any emergency vehicle will be pre-empted first on priority.

2.2.2.2 Product Features

The software implementation of the project will consist of mainly 3 phases: The Vehicle Detection Module, Signal Switching Algorithm and finally the Pygame simulation. Detailed description of each module is as follows:

- Vehicle Detection Module: This module is responsible for detecting the number of vehicles in each lane. A video feed containing traffic congestion of four lanes will be provided as the input to this module, then density calculation based on the area occupied by the vehicles will be done and based on the density calculation results, further modules will process the timings of the signals.

- Signal Switching Algorithm: This algorithm updates the red and green times of all signals. This module will take density inputs from the Vehicle Detection Module, the lane with the highest density will be pre-empted first, with its timer beingreduced from its default timings. These timers are set based on the count of vehiclesof each class received from the vehicle detection module and will act accordingly if a priority vehicle is detected in the video feed.

- Simulation Module: A virtual simulation is developed using Pygame library to simulate traffic signals and vehicles moving across a traffic intersection. The video feed of vehicles while intersecting the lanes are taken as the input feed, the density of each lane is calculated on the basis of area occupied by the various classes of vehicles,the lane with the highest traffic congestion will be pre-empted first and hence reducing the congestion for the busiest lane.

2.2.3 External Interface Requirements

The section explains about the interface requirements under the project. Some light is thrown on the user, hardware and software interfaces.

2.2.3.1 User Interfaces

The two points of interaction interfaces with the world elements are the camera providing the input video feed and the traffic light system with the corresponding lightings and timings.

2.2.3.2 Hardware Interfaces

A PC system would be the interface for all the vehicle detection and computing for scheduling. Apart from that we'd use an arduino based interface performing the simulation of the traffic light to display the timings and lightings.

2.2.3.3 Software Interfaces

The entire code for virtual simulation, vehicle detection as well as scheduling algorithms for computing are written in Python language. We will be accepting the inputvideo would be in the .mp4 format of use. For the hardware simulation we would be using the Arduino language.

2.2.3 Other Non-functional Requirements

Non-functional requirements are important as all the requirements are considered below in this section. Basic Non- functional requirements are:

1. The vehicle detection of the system should be fast as it needs to keep count of vehicles of various classes in each lane in order to calculate the traffic occupancy on the go dynamically.
2. The priority scheduling algorithm must have a small-time complexity as it would delay the operations if it takes too much of the time.
3. System should be reliable in terms of power failure etc.
4. System should be robust to any ambiguities.

2.2.4.1 Performance Requirements

As for this prototype version, during the testing, the performance of the system in terms of efficiency and throughput given by the system after the integration of different components.

2.2.4.2 Safety Requirements

Since we would be using a durable and protected PC system it would be efficient to perform tasks so make our system efficient. High quality cameras are used with good coding to make the system effective. New features will be added in the future if needed.

2.2.4.3 Security Requirements

Since most of the connections between the various components of the system would be wired and hardware based, the whole system would be able to run autonomously asa separate identity reducing the chance of any security threat or issues.

2.3 Cost Analysis

Since our proposed solution is primarily a software-based program as of now, we don't require any external equipment except our own computer system to run the program.

2.4 Risk Analysis

S. No	RISKS	MITIGATION STEPS
1.	The computing unit may face power shortage.	To use a backup power options.
2.	The system gets hacked by a malicious party which cause difficulties in operation.	Using proper secure network standards which would reduce the chances of such malicious activities.
3.	Error in density calculation	Have to be accurate with the area and calculations
4.	The system faces an error due to presence of error in code.	The code should be correctly written free from syntax and other form of errors.

5.	Inability to complete product due to these unprecedented times.	Keep contact more often, try to finish beforedeadline

METHODOLOGY ADOPTED

3.1 Investigative Techniques

Investigative Technique Involved: DESCRIPTIVE

Our project consists of a vehicle detection module, scheduling and prioritization module which detects the normal and emergency vehicle and sets timers of traffic lights accordingly. It requires a new scheduling algorithm which could detect emergency vehicles in all of the lanes and schedules the duration of red and green light signals accordingly. Formulating a new formula for calculating timings based upon both the traffic density and presence of emergency vehicles, it demands a new algorithm and concept to fulfil the objective of our project. Considering the stated requirements, our investigative technique falls under the category of descriptive.

Our intelligent traffic management system monitors the traffic lanes to identify the area with high traffic congestion and in accordance with it, the system schedules the traffic flow accordingly and hence reduces congestion.

3.2 Proposed Solution

The proposed solution to the existing problem consists of various sub-modules like: the vehicle detection module, density calculation module, scheduling algorithm module, and simulation module. These modules work in the following way:

1. Vehicle Detection and Density Calculation Module: This module is responsible for detecting each type of vehicle and estimating the number of vehicles of that type in each lane. Video feed of all lanes is passed as input, and then processing is done on those videos. Emergency vehicles are detected in each lane, and then the presenceof them is passed to the signal switching algorithm module. The priority is decided on the basis of whether the emergency vehicle is present in the lane or not. Vehicles are detected using yolo-v3 library. Coco dataset is used for detecting vehicles on the road.

2. Signal Switching Algorithm: This algorithm updates the red, green, and yellow times of all signals. This module will take density inputs from the Vehicle Detection Module, the lane with the highest density will be pre-empted first, with its timing being increased from its default timings. These timers are set based on the count of vehicles of each class received from the vehicle detection module and will act

accordingly if a priority vehicle is detected in the video feed. If an emergency vehicle is present in any lane, then its priority is set higher and the timing of green signal is increased from the default value.

3. Simulation Module: A simulation is developed using Pygame library to simulate traffic signals and vehicles moving across a traffic intersection. The video feed of vehicles while intersecting the lanes are taken as the input feed, the density of each lane is calculated based on the area occupied by the various classes of vehicles, the lane with the highest traffic congestion will be preempted first hence reducing the congestion for the busiest lane.

The pseudo code for the Pygame simulation is as follows:

```
function setTime():
 set maxamb=-1   #initial maximum ambulance count=-1
 set noOfCars, noOfBuses, noOfTrucks, noOfRickshaws, noOfBikes,
noOfAmbulances = 0,0,0,0,0,0
 set cycle[currentGreen]="true"
 visitedlanes=visitedlanes+1
 for k=0 to 3 do
        if cycle[k]=="false" then
            for i=1 to 2 do
                for                    j=0                    to
len(vehicles[directionNumbers[k]][i]) do
                    set                 vehicle                =
vehicles[directionNumbers[k]][i][j]
                    if(vehicle.crossed==0) then
                         set vclass = vehicle.vehicleClass
                         if(vclass=='ambulance') then
                             noOfAmbulances += 1
            set ambTrack = noOfAmbulances               #counter
variable to keep track of ambulanace
            if(ambTrack>maxamb) then
                maxamb = ambTrack                        #update
the maximum ambulance count
                nextGreen = k
 for j=0 to len(vehicles[directionNumbers[nextGreen]][0]):
                set                   vehicle                   =
vehicles[directionNumbers[nextGreen]][0][j]
                if(vehicle.crossed==0) then
                    set vclass = vehicle.vehicleClass
                    # print(vclass)
                    noOfBikes += 1
 for i=1 to 3 then
        for                         j=0                         to
len(vehicles[directionNumbers[nextGreen]][i]) then
            set                     vehicle                     =
```

```
vehicles[directionNumbers[nextGreen]][i][j]
 if(vehicle.crossed==0) then
   vclass = vehicle.vehicleClass
   # print(vclass)
   if(vclass=='car') then
   noOfCars += 1
   elif(vclass=='bus') then
   noOfBuses += 1
   elif(vclass=='truck') then
   noOfTrucks += 1
   elif(vclass=='rickshaw') then
   noOfRickshaws += 1
   elif(vclass=='ambulance') then
      noOfAmbulances += 1
    signals[nextGreen].red                                          =
signals[currentGreen].yellow+signals[currentGreen].green
    greenTime      =      math.ceil(((noOfCars*carTime)      +
(noOfRickshaws*rickshawTime) +
    (noOfBuses*busTime)        +        (noOfTrucks*truckTime)+
(noOfBikes*bikeTime)                                            +
(noOfAmbulances*ambulanceTime))/(noOfLanes+1))

    print('Green Time: ',greenTime)
    if(greenTime<defaultMinimum) then
        set greenTime = defaultMinimum
    elif(greenTime>defaultMaximum) then
        set greenTime = defaultMaximum

    signals[nextGreen].green = greenTime

    if(visitedlanes == 4): then
        for j=0 to 3 do
            cycle[j]="false"
        set visitedlanes=0
```

3.3 Work Breakdown Structure

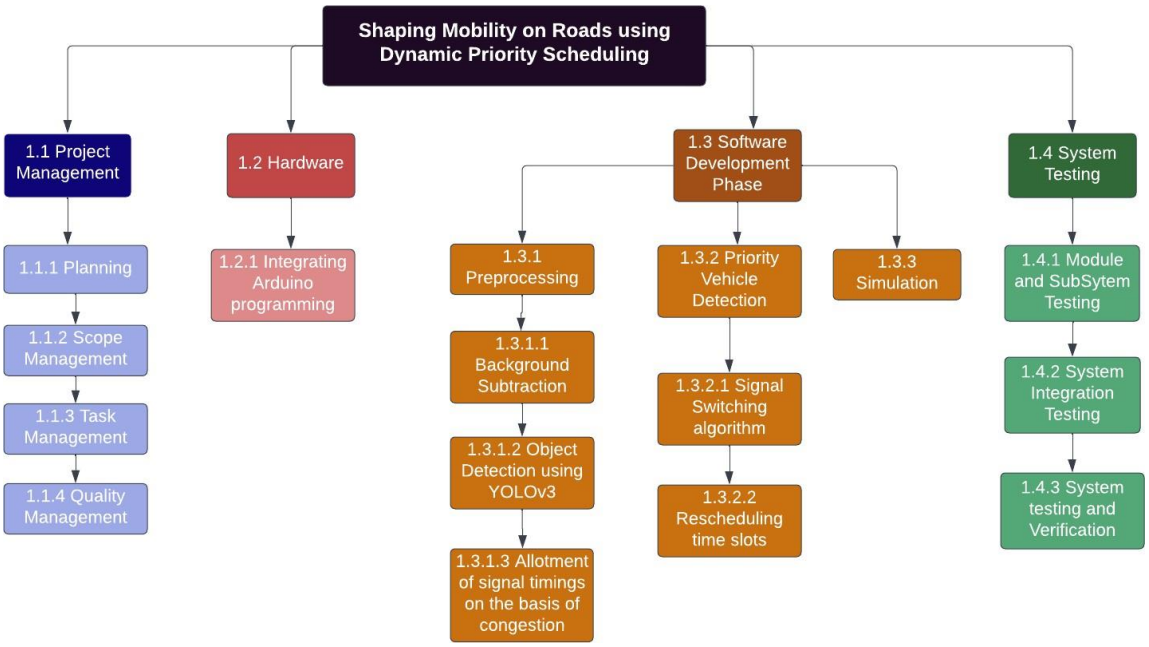


FIGURE 3: Work Breakdown Structure

The work breakdown structure shown in figure 3 demonstrates the discrete work elements in which the whole project is divided. Starting from management, which includes planning, task management along with quality management, followed by hardware. The hardware components will include camera, Arduino, LEDs and a 7-segment display. The project involves 3 main phases (the software development phase being the most important): pre-processing (applying contouring and gray-scaling), vehicle detection which will check for the density of each lane or any emergency vehicle in any of the lanes for prioritizing the pre-emption of emergency vehicles and dynamic allocation to signal switching timers.

3.4 Tools and Technology

- **Arduino Uno :**The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller.
- **Camera Module:** It is used to capture real time traffic scenario of different lanes
- **Open CV:** OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source Apache 2 License.

- **Pygame:** Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.
- **Visual Studio Code:** Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
- **Artificial Intelligence:** It is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. We are using it to schedule timings ofdifferent lanes in our project.
- **Deep Learning:** Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabelled.

4.1 System Architecture

4.1.1 Component Diagram

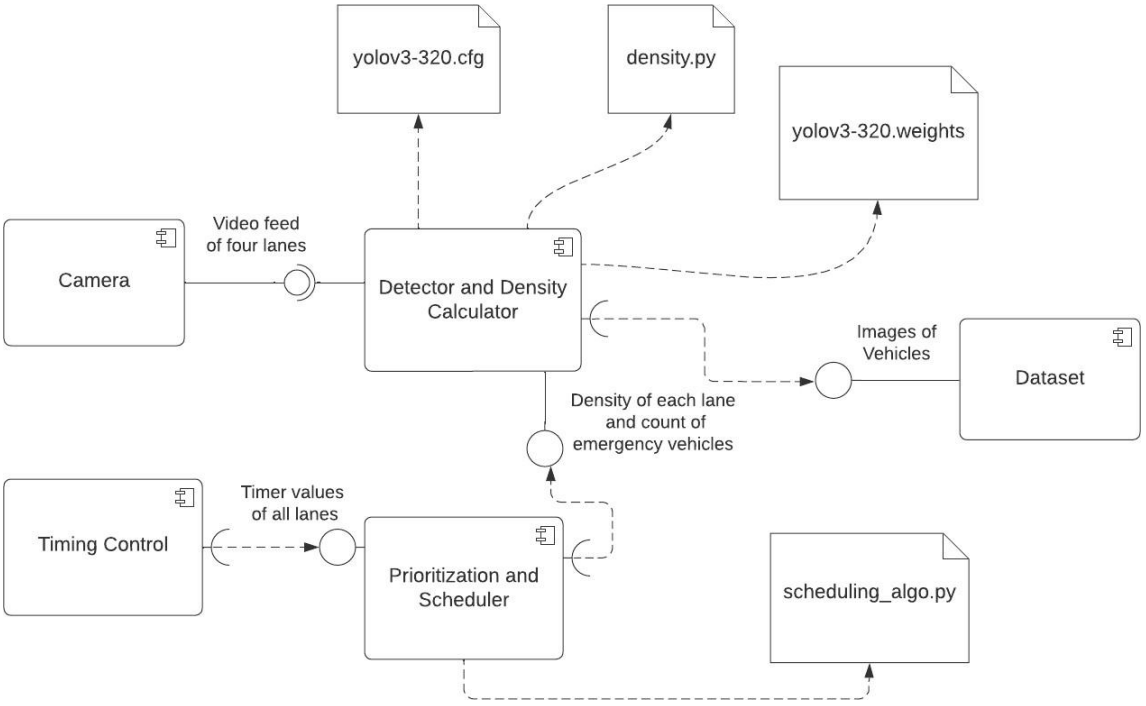


FIGURE 4: Component Diagram for SMRUDPS

The diagram in figure 4 contains five components. First component is camera which provides video feed of all the four lanes. Second component is Detector and Density Calculator which receives the input of video feed of all four lanes. This component then detects the vehicles on the road by using images from the dataset for reference. Then it calculates the density of each road and sends the count of emergency vehicles in all the lanes to the third component. This component uses YOLO v3-320.cfg and YOLO v3-320.weights files for detecting vehicles on road and it uses density.py fileto calculate density of vehicles on road.

Third component is Prioritization and Scheduler, which receives the information about density of each lane and the presenceof emergency vehicles in them and then it assigns the priority to each lane and schedule the time limit of green and red signals for that lane. This component uses scheduling_algo.py file for scheduling time to lanes according to their priority. The timer values provided by this component are then received by Timing Controlcomponent that sets the timers accordingly.

4.2 Design Level Diagrams

4.2.1 Class Diagram

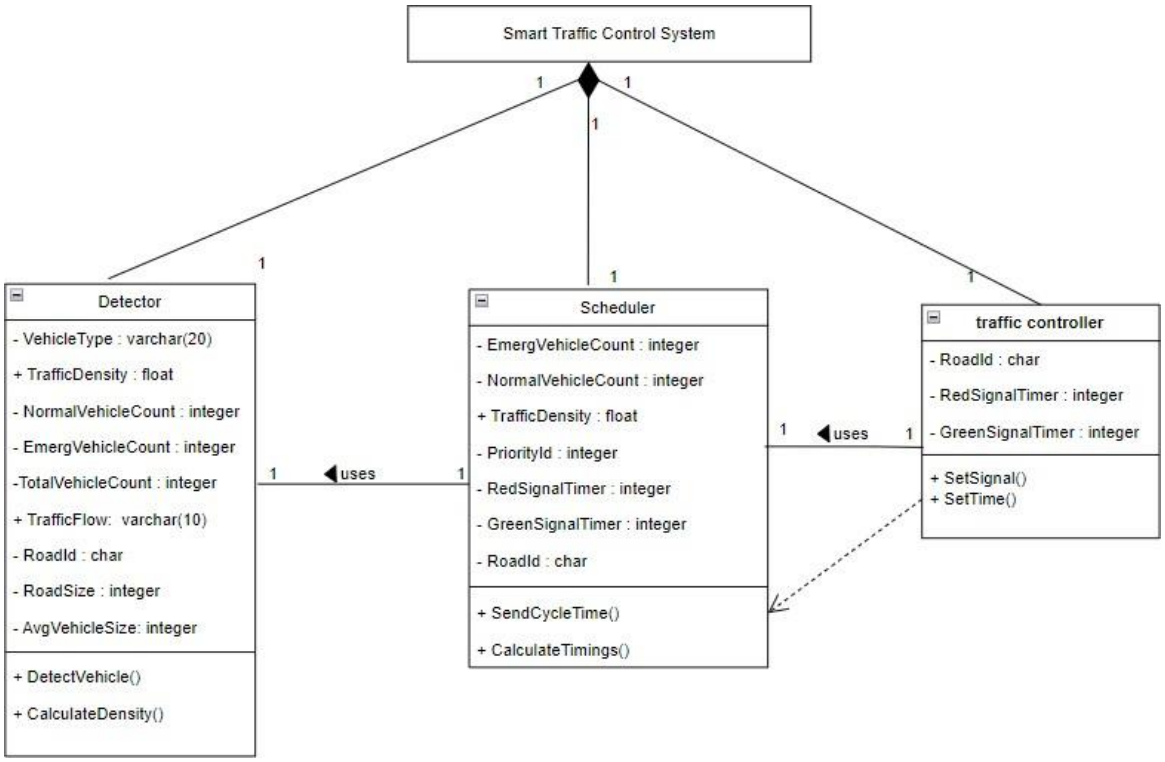


FIGURE 5: Class Diagram for SMRUDPS

The Class Diagram depicted in figure 5 describes the attributes and operations of a class and also the constraints imposed on the system. It is a static diagram which represents the static view of an application.

The smart traffic control system is the main class which comprises class detector, scheduler and traffic controller. This class is crucial for the existence of the allclasses, which is depicted by the composition symbol as shown in fig 5. The detector class performs object detection and density calculation using public methods. The Scheduler class is associated with detector class through 1 - 1 relationship wherein scheduler uses the detector class for its functionality. The traffic controller class method namely SetTime() uses the method CalculateTimings() of detector class so asto set the timings by taking the calculated timings for each lane by detector.

4.2.2 Data Flow Diagrams

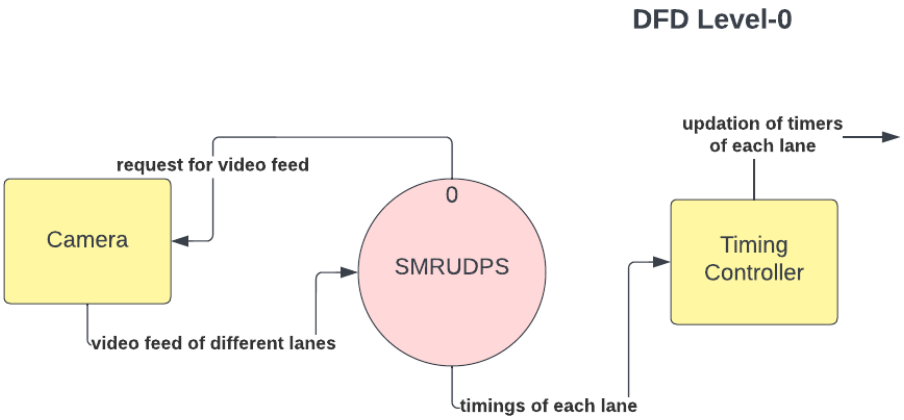


FIGURE 6: DFD Level 0 Diagram for SMRUDPS

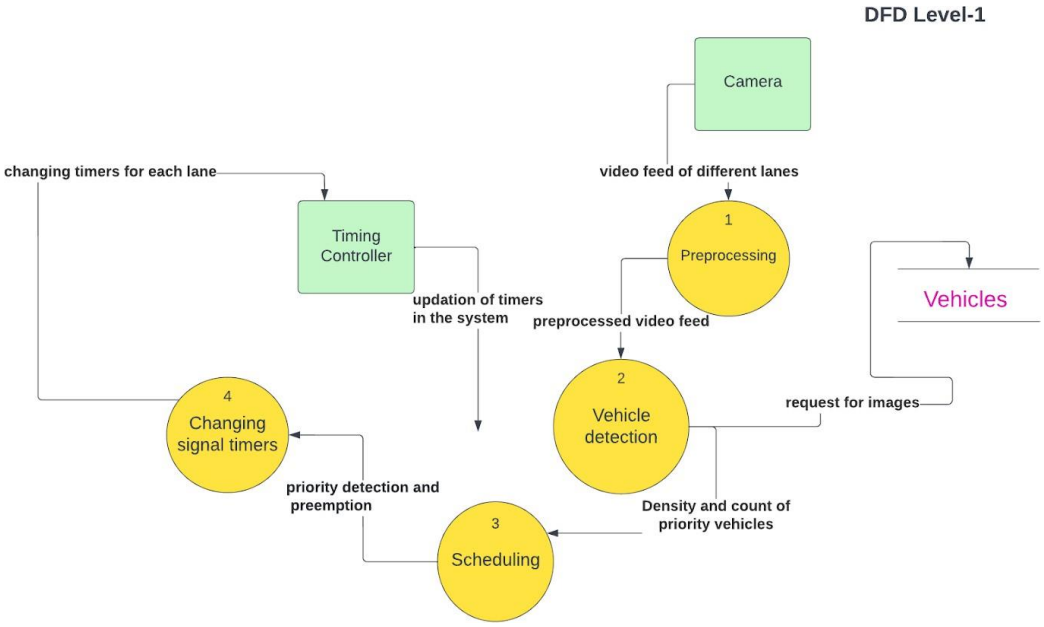


FIGURE 7: DFD Level 1 Diagram for SMRUDPS

DFDs stand for data flow diagrams, these diagrams represent the flow of data through different processes and entities of the project/system. In the figure 5 and 6, two levels of DFDs are presented where in the first DFD (level-0), the whole system is taken as a process and data flow between different entities is shown. The second level is a detailed version of the first DFD wherein each entity is connected to its main process and the databases (Vehicle images in our project) that are required in order to performtheir respective processes. Some rules are to be followed while forming and DFD:

- Level-0 DFD can have only 1 process and other DFD levels can have more than 1 process.
- We cannot present databases at level-0 DFD, and hence level-0 is the simplest/basic level of any DFD.
- The number of incoming arrows should be equal to the number of outgoing arrows for any process.

The level-0 DFD consists of two entities i.e. camera and the timing controller, SMRUDPS being the only process. The level-1 DFD consists of the same two entities with an increase in the number of processes, which are interacting with each other. The process named pre-processing will send the video feed to the vehicle detection module which will give us the count of priority vehicles and density of each lane after using the vehicle images from the database, these results will be served as input to the scheduling process for prioritization and pre-emption, and finally change of timings will be processed through the timing controller.

4.2.3 Activity Diagram

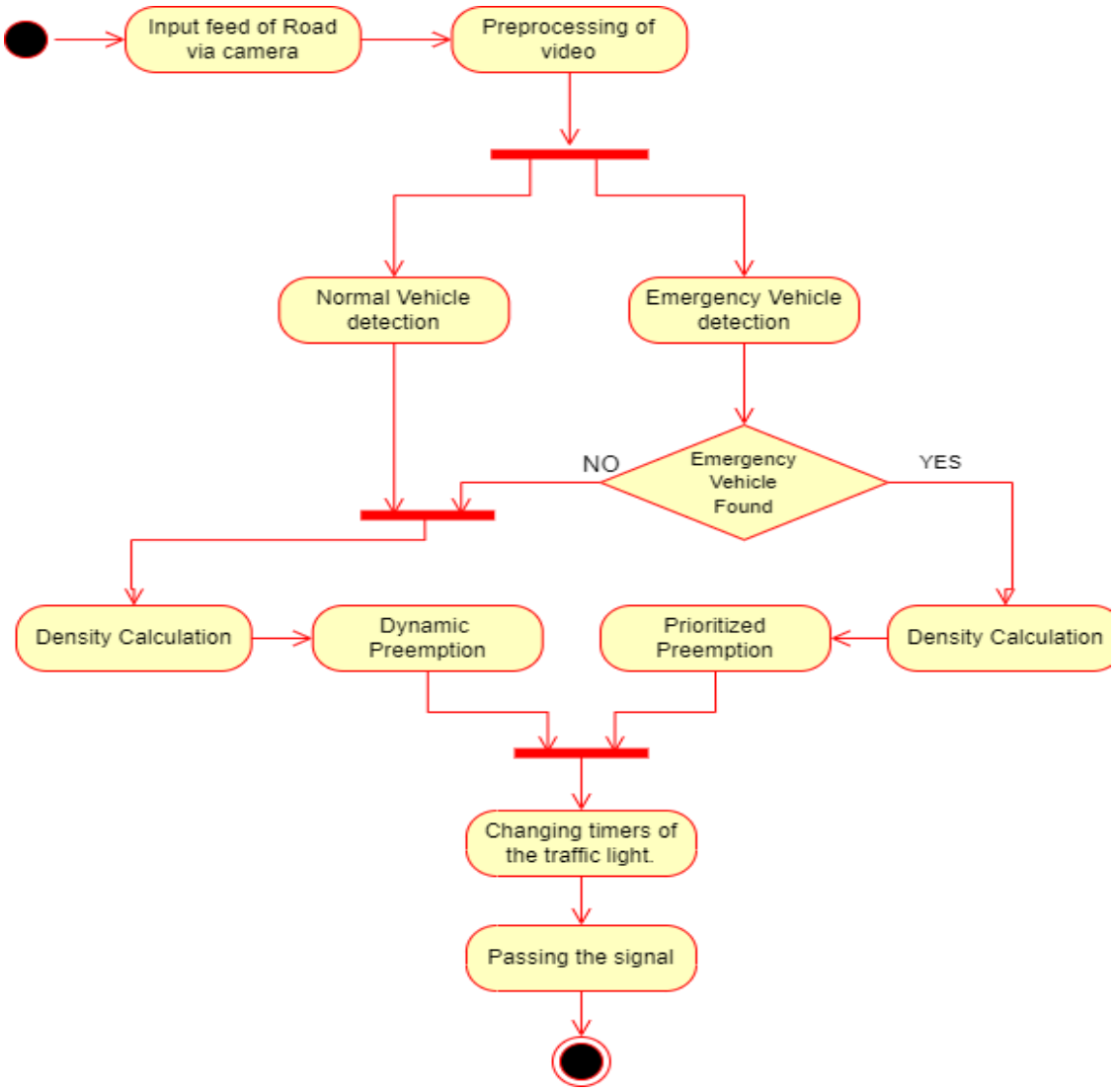


FIGURE 8: Activity Diagram for SMRUDPS

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

The activity diagram shown in figure 8. It shows the complete set of operations conducted by our smart traffic control system in order to achieve optimization in terms of efficiency.

The process starts with receiving the input video feed from the camera and further pre-processing it for further use. The next step is vehicle detection of both normal as well as emergency vehicles. We calculate the traffic density of every lane with the

help of vehicle count. If the emergency vehicle is found then we move onto prioritized pre-emption else we go for normal dynamic pre-emption. In accordance with the pre-emption occurred the new timings of the traffic light are dynamically calculated. These timing signals are then conveyed to the traffic lights.

4.2.4 Swimlane Diagram

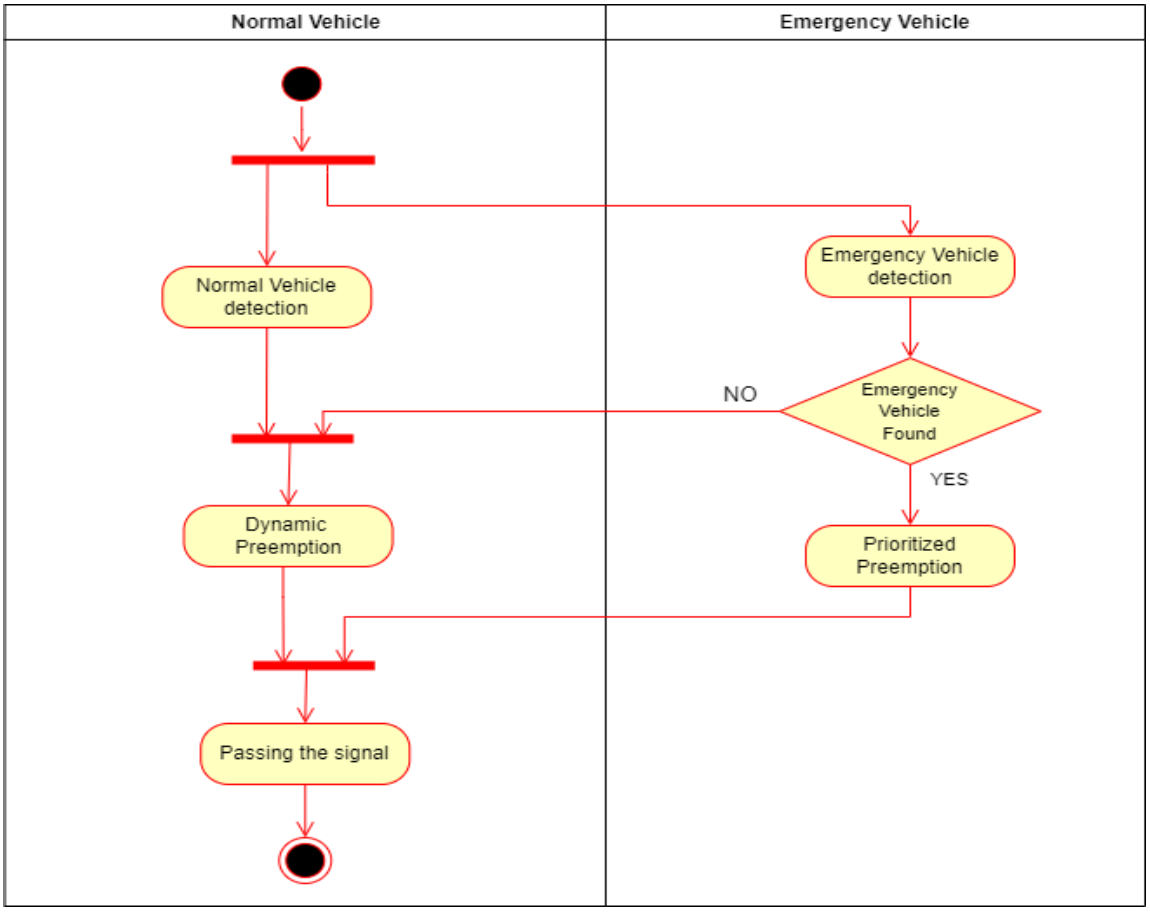


FIGURE 9: Swimlane Diagram for SMRUDPS

Swimlane diagram is another important diagram in UML to describe the dynamic aspects of the system in relation to the entities present in the system. Swimlane diagram is basically a modification of an activity diagram which shows the relationship between each operation and the corresponding entities/objects in thesystem.

In the Figure 9, we can clearly see the relationship between the operations shown in the activity diagram and the object entities of the system that are normal vehicles and emergency vehicles. We can present the observation that prioritized pre-emption is a

special case that occurs in the presence of emergency vehicle detection, else dynamic pre-emption occurs.

4.2.5 Sequence Diagram

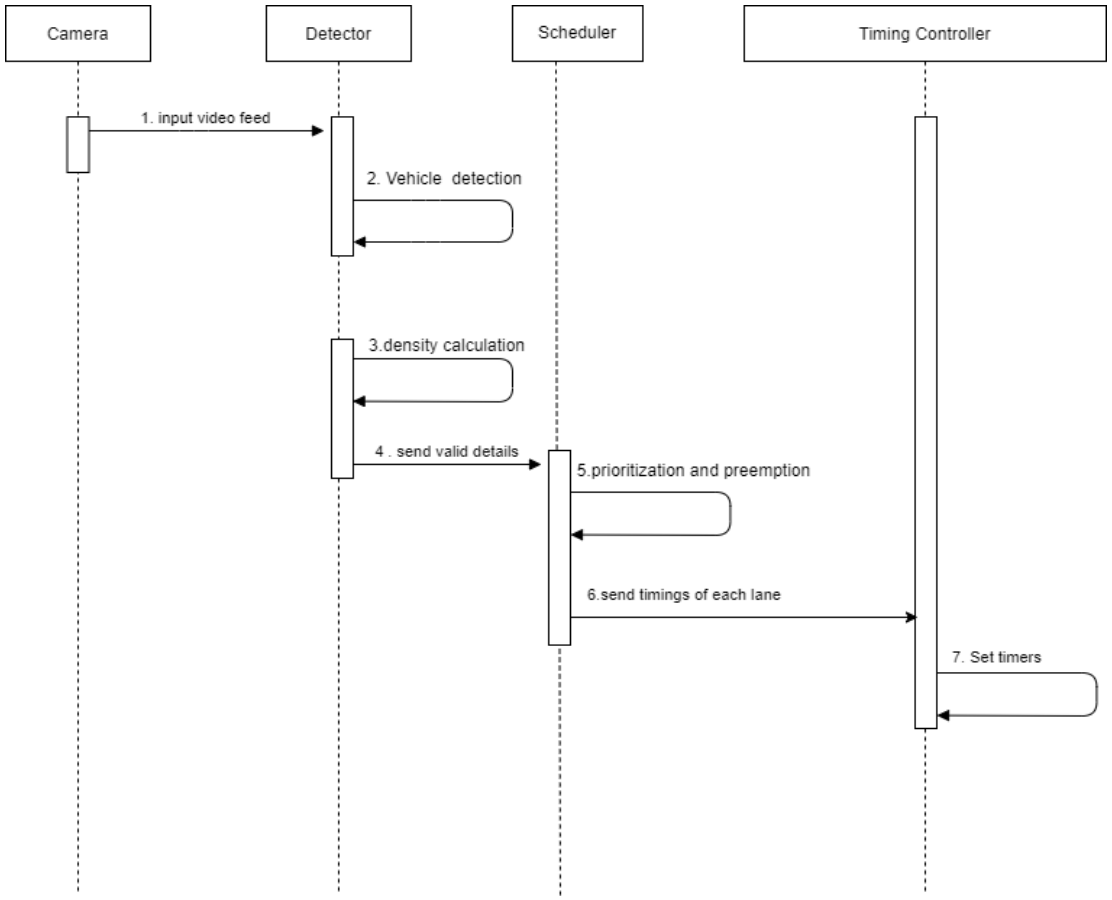


FIGURE 10: Sequence Diagram for SMRUDPS

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. The above diagram showsthe sequence of events performed by various components of the system in a sequentialmanner.

4.1.1 Use Case Diagram and Template

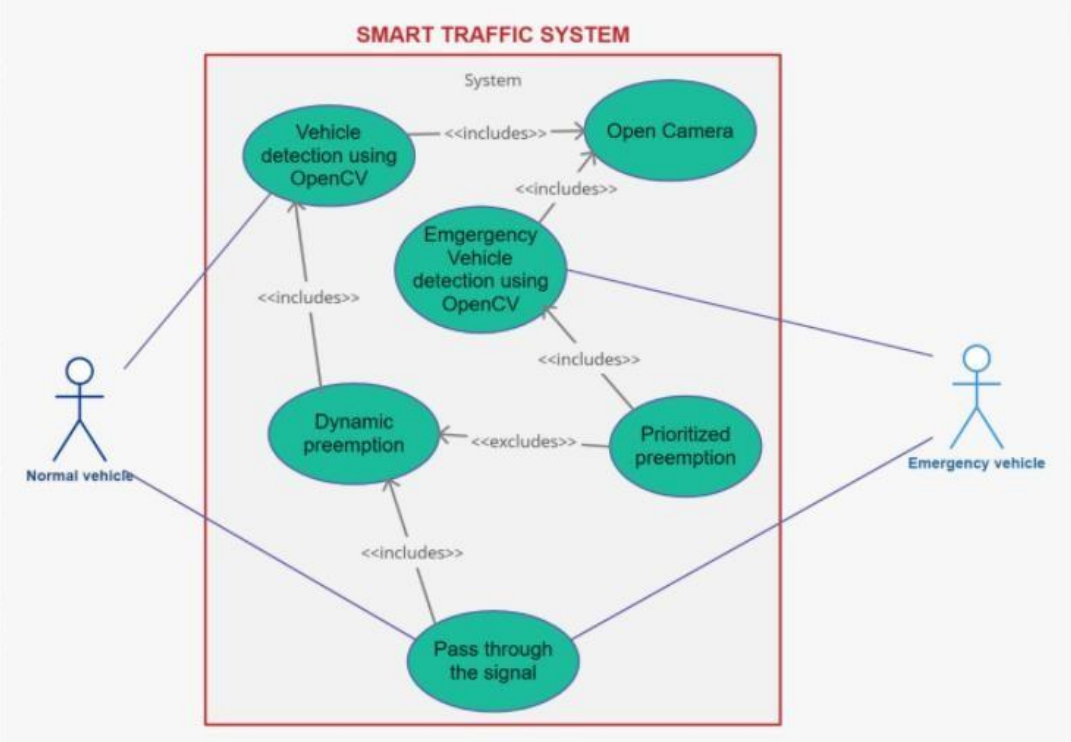


FIGURE 11: Use Case Diagram for SMRUDPS

Use case diagram, figure 11 demonstrates the high-level functions and the scope of a system. Normal vehicle and priority vehicles are the actors in the diagram which access the Vehicle detection use case and emergency vehicle use case respectively. The vehicle detection case includes an open camera. The vehicle detection case then has to be included for dynamic pre-emption use case. The prioritized pre-emption use case will include emergency vehicle detection case. The dynamic pre-emption and theprioritized pre-emption cases will exclude each other as there is no need of dynamic allocation once pre-emption is prioritized. And finally, the pass-through signal use case comes in contact with the actors.

Use Case Template:

Name	Vehicle Detection Using OpenCV
Description	This use case detects the vehicles on the lanes and classify them and accordingly, the density of each lane is calculated.
Actors	Normal Vehicle, Emergency Vehicle
Preconditions	There must be a real time feed through the camera of the lanes forthis use case to work.
Main Course	1. The system demands the real time video feed of the traffic lanesas the input. 2. The system applies Vehicle Detection algorithm on the videofeed and classify vehicles. 3. The system calculates the density of each lane depending on thenumber and type of vehicles in that particular lane. 4. The system then sends the density measurements to thescheduling algorithm.
Alternative Course	AC1 The system is unable to detect the vehicles on the road 1. Error message is displayed on the screen. 2. Return to main course step 2 AC2 The system is unable to detect the input video feed 1. Error message is displayed on the screen. 2. Return to main course step 1

Postcondition	The density measurements are passed to the scheduling algorithmas input.
Exceptions	EX1 The video feed is invalid or not supported 1. An error message “File type not supported” is displayed onthe screen. 2. Return to main course step 1

Name	Emergency Vehicle Detection Using OpenCV
Description	This use case detects the emergency vehicles on the lanes and setsthe priority of that particular lane high.
Actors	Normal Vehicle, Emergency Vehicle
Preconditions	There must be a real time feed of the lanes through the camera forthis use case to work.
Main Course	1. The system demands the real time video feed of the traffic lanesas the input. 2. The system applies Vehicle Detection algorithm on the videofeed and detects emergency vehicles. 3. The system sets the priority of that particular lane whereemergency vehicle is detected high.

	4. The priority is then sent to the dynamic scheduling algorithm.
Alternative Course	AC1 The system is unable to detect the emergency vehicles on theroad 1. Error message is displayed on the screen. 2. Return to main course step 2 AC2 The system is unable to detect the input video feed 1. Error message is displayed on the screen. 2. Return to main course step 1
Postcondition	The priorities of the lanes are passed to the scheduling algorithmas input.
Exceptions	EX1 The video feed is invalid or not supported 1. An error message “File type not supported” is displayed onthe screen. 2. Return to main course step 1

Name	Dynamic Preemption
Description	This use case dynamically schedules the time duration of trafficlights depending on the density of the lanes.
Actors	Normal Vehicle, Emergency Vehicle
Preconditions	The density of all the lanes should be passed as an input to thisscheduling algorithm.
Main Course	1. The system demands the densities of all the lanes as an input. 2. The system applies scheduling algorithm to the lanes and sets the priority of each lane. The lane with highest density has the highest priority. 3. The system sets the time duration of green light according to the priority of each lane. If the priority is higher then the time duration of green light will be more.

Alternative Course	AC1 The system is unable to receive the density measurements 1. Error message is displayed on the screen. 2. Return to main course step 1
Postcondition	The time duration of the traffic lights respective to each lane is setaccording to their priority.
Exceptions	None

Name	Prioritized Pre-emption
Description	This use case dynamically schedules the time duration of trafficlights depending on the emergency vehicle present in the lane.
Actors	Normal Vehicle, Emergency Vehicle
Preconditions	The presence of emergency vehicle in any lane must be passed asan input to this scheduling algorithm.
Main Course	1. The system demands the assurance of existence of emergency vehicle in any lane as an input. 4. The system applies scheduling algorithm to the lanes and sets the priority of each lane. The lane with the emergency vehicle has the highest priority. 5. The system sets the time duration of green light according to the priority of each lane. If the priority is higher than the time duration of green light will be more.

Alternative Course	AC1 The system is unable to receive the signal showingemergency vehicle is present in some lane 1. Error message is displayed on the screen. 2. Return to main course step 1
Postcondition	The time duration of the traffic lights respective to each lane is setaccording to their priority.
Exceptions	None

Name	Pass through the signal
Description	This use case lets the vehicles to pass through the signal when theturn of that particular lane comes
Actors	Normal Vehicle, Emergency Vehicle
Preconditions	The time duration according to the priority order of the lanes ispassed as an input
Main Course	1. The system receives the time duration of the lanes as an input. 2. The system allows the vehicles of that particular lane to passthrough for that particular time duration.
Postcondition	The vehicles of particular lanes are passed through the signalaccording to the time duration of those lanes.

Exceptions	None

IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1 Experimental Setup

The experimental setup of detection of vehicles in video feed include detection of vehicles using yolov3 configuration and weight files. COCO data set has been used to train yolov3 model to classify vehicles as car, truck, motorcycle and bus. For experimenting with yolov3 models, methodologies such as DNN has been used for detection and classification of vehicles. All the experimentation and scheduling algorithms have been implemented with python3.10. Image processing has been performed on input video feeds using opencv-python4.6.0.66. Video processing has been done on four video feeds at the same time using multiprocessing and then the priority scheduling has been performed based on the density of lanes and the presence of emergency vehicles in any lane.

For the hardware requirement, NVIDIA Geforce GTX 1650 Ti graphical processing unit has been used. OpenCv modules of version 4.5.2 have been configured with CUDA files to integrate with GPU.

The experimental setup of Pygame simulation has been conducted with python 3.10 (64 bit). The model experimentation included python Pygame 2.1.2 library which includes modules for graphic drawing, motion handling etc. The experimental model adopted uses multi-threading for efficient processing and achieving multitasking. Custom dataset has been created and incorporated which included .png images.

5.2 Experimental Analysis

5.2.1 Data

The dataset used in this program is Microsoft COCO: Common Objects in Context, a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is accomplished by get-together pictures of perplexing regular scenes containing normal articles in their regular setting. Objects are named utilizing per-case divisions to help with exact item restriction. Dataset contains photographs of 91 articles types that sounds effectively unmistakable, truly. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting

and instance segmentation. Presents a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, it also provides baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

5.3 Working of the project

5.3.1 Procedural Workflow

This project consists of two parts of simulation. In first part Pygame has been used to simulate the real time environment of traffic and then scheduling the green times of the lanes according to their respective densities. In this simulation data set has been generated with images of vehicles like: motorbike, car, truck and ambulance and images of lanes and traffic lights. Then the scheduling has been done on the basis of density. The lane with higher density has been allotted more green time and the lane with lower density has been given less green time. Priority scheduling has also been incorporated in this simulation by giving more priority to the lane that contains emergency vehicle in it. Round Robin scheduling has been implemented overall and this order gets changed only when an emergency vehicle gets detected in any lane.

Second part of the project deals with the video processing of four lanes provided as an input video feeds. It involved detection and classification of vehicles in videos. Density has been calculated of these lanes based on the following formula:

```
greenTime = math.ceil(((noOfCars*carTime) + noOfRickshaws*rickshawTime) + (noOfBuses*busTime) + (noOfTrucks*truckTime)+ (noOfBikes*bikeTime) + (noOfAmbulances*ambulanceTime))/(noOfLanes+1))
```

Here, the green times of all the four lanes have been calculated using the above formula. In this formula, number of vehicles belonging to a particular category are multiplied by the average time that vehicle of a particular category takes to the cross the intersection. All types of vehicles have been generated randomly in the pygame simulation. This simulated the real time environment.

In second part of simulation, video feeds of four lanes showing traffic have been processed. Different types of vehicles have been classified like: car, truck, bus, motorcycle and ambulance using yolov3 model configuration and weight files. This involved the concept of deep neural networks. After the detection and count of vehicles of particular type, density has been calculated using the same formula given above.

Default average times taken by vehicles to cross the intersection have been used in the calculation of density. The round robin scheduling algorithm has been used to green the signals of all four lanes.

The steps involved in the computation of green time are as follows:

1. All the videos of four lanes were processed for 10 seconds. Total number of vehicles of every type were calculated in each video going downwards. Then density was calculated based on the default average time a vehicle took to cross the intersection. In first round, first lane's signal was turned green and the time was allotted based on the density of that particular lane.

2. For the second round, all the videos of four lanes were again processed for 10 seconds and density was calculated for each lane. Now, if an ambulance would have been present in any lane other than the first lane, then the signal of that lane is turned green next time.

3. Similarly, for next two rounds green signal was given to the lane as per the order in round robin scheduling. The green timer of that particular lane is printed on the command prompt to show the results.

5.3.2 Algorithmic Approaches Used

The algorithm designed for pygame simulation has setTime() function , which is the most important piece of code in our simulation. This function is indented outside of the vehicle class and is responsible for priority scheduling and dynamic scheduling which allocates the green signal timer value based upon the density. We refer to different lanes through the currentGreen variable whose value varies from 0 to 3. Initially the count of every vehicle is 0. We have maintained a cycle array to maintain track for each cycle and in every cycle, we observe the type of vehicle that is generated and accordingly increment the count of that particular type of vehicle. Red signal timer value is calculated by summation of the present green signal lane's yellow timer and green timer. The greenTimer value is calculated by summation of product count of vehicle and its average time to cross the intersection which is initialized at the start.

The value of greenTime can vary between default minimum green time which is 10sec and default maximum which is set to 60sec. The nextGreen lane is chosen from the unprocessed lanes and is incremented by 1 i.e by following the **Round-Robin** algorithm. When the emergency vehicle is detected in any lane, then that lane is to be given the green signal next hence the **priority scheduling.**

```
function setTime():
 set maxamb=-1   #initial maximum ambulance count=-
1
 set cycle[currentGreen]="true"
 visitedlanes=visitedlanes+1
 for k=0 to 3 do
        if cycle[k]=="false" then
    for i=1 to 2 do
                for              j=0              to
len(vehicles[directionNumbers[k]][i]) do
                    set          vehicle          =
vehicles[directionNumbers[k]][i][j]        #vehicle
object has all attributes
                    if(vehicle.crossed==0) then
                         set        vclass        =
vehicle.vehicleClass
                         if(vclass=='ambulance')
then
                             noOfAmbulances += 1
            set              ambTrack              =
noOfAmbulances                 #counter variable to
keep track of ambulanace
            if(ambTrack>maxamb) then
                maxamb                             =
ambTrack                        #update the maximum
ambulance count
                nextGreen = k
 for                    j=0                      to
len(vehicles[directionNumbers[nextGreen]][0])
then        #for bikes
                set            vehicle             =
vehicles[directionNumbers[nextGreen]][0][j]
                if(vehicle.crossed==0) then
                    set          vclass            =
vehicle.vehicleClass
                    # print(vclass)
                    noOfBikes += 1
 for i=1 to 3 then
        for                 j=0                  to
len(vehicles[directionNumbers[nextGreen]][i]) then
            set              vehicle               =
vehicles[directionNumbers[nextGreen]][i][j]
            if(vehicle.crossed==0) then
                vclass = vehicle.vehicleClass
                # print(vclass)
                if(vclass=='car') then   #increment
the count of vehicles
                    noOfCars += 1
                elif(vclass=='bus') then
                    noOfBuses += 1
                elif(vclass=='truck') then
                    noOfTrucks += 1
                elif(vclass=='rickshaw') then
```

```
                    noOfRickshaws += 1
                elif(vclass=='ambulance') then
                    noOfAmbulances += 1
    set           signals[nextGreen].red          =
signals[currentGreen].yellow+signals[currentGreen]
.green
    greenTime  =  math.ceil(((noOfCars*carTime)  +
(noOfRickshaws*rickshawTime) +
    (noOfBuses*busTime)  +  (noOfTrucks*truckTime)+
(noOfBikes*bikeTime)                              +
(noOfAmbulances*ambulanceTime))/(noOfLanes+1))    #
greentime calculation

    print('Green Time: ',greenTime)
    if(greenTime<defaultMinimum)  then         #The
green timer can take value between default minimum
and default maximum
        set greenTime = defaultMinimum
    elif(greenTime>defaultMaximum) then
        set greenTime = defaultMaximum

    set signals[nextGreen].green = greenTime

    if(visitedlanes == 4) then
        for j=0 to 3 do
            set cycle[j]="false"
        set visitedlanes=0
```

Pseudo code of setTime() function

render() function is used to display the image on the screen and move() function manages the movement of vehicles according to the signals and the presence of vehicles ahead by maintaining the distance and stopping gap . The Repeat() function is a recursive function that runs for the whole time of simulation and leads the simulation.

In case of vehicle detection and scheduling of lanes using video feeds as an input, there were two important functions. Main() function contained the scheduling algorithm that controlled the lane for which the signal has to be turned green in the next round. Preprocessing of all the videos before they could be sent for scheduling was done in realTime() function. This function detected all the vehicles of each type and maintained the count for the same for all the four videos of lanes.

```
for round in range(0,4):

        processed_lane[currentGreen] = "True"



        Videos  =  ['Resources/res5_video_10.mp4'  ,
'Resources/res3_video_10.mp4'                      ,
```

```
'Resources/res6_video_10.mp4'                          ,
'Resources/res5_video_10.mp4','Resources/res7_vide
o_10.mp4']

        lane1 = multiprocessing.Value('i',0)



        ambulance_detected1=
multiprocessing.Value('i' , 0)



        total_vehicle1=multiprocessing.Value('i'
,0)



        count = 0

        for i in Videos:

            count=count+1

            if count==1:

                process                              =
multiprocessing.Process(target=realTime,
args=(i,lane1,ambulance_detected1,total_vehicle1,)
)

                process.start()

                processes.append(process)

            elif count==2:

                process                              =
multiprocessing.Process(target=realTime,
args=(i,lane2,ambulance_detected2,total_vehicle2,)
)

                process.start()

                processes.append(process)

            elif count==3:

                process                              =
multiprocessing.Process(target=realTime,
args=(i,lane3,ambulance_detected3,total_vehicle3,)
)

                process.start()
```

```
                processes.append(process)

            elif count==4 and round!=1:

                process                             =
multiprocessing.Process(target=realTime,
args=(i,lane4,ambulance_detected4,
total_vehicle4,))

                process.start()

                processes.append(process)

            elif count==5 and round==1:

                process                             =
multiprocessing.Process(target=realTime,
args=(i,lane4,ambulance_detected4,
total_vehicle4,))

                process.start()

                processes.append(process)

            else :

                pass

        for process in processes:

            process.join()



        for i in range(0,4):

            if i==currentGreen:

                print('Signal                    for
lane',i+1,'Green     and     Green   Timer   =   ',
greentimer[i].value)

            else:

                print('Signal for lane',i+1,'Red')



        counter=0

        for i in range(0,4):

            if processed_lane[i]=="False":

                counter=counter+1
```

```
                if counter==1:
                    currentGreen=i


                elif
ambulance_detected[i].value==1 and counter!=1:
                    currentGreen=i
```

This is the pseudo code for multiprocessing. Here, all the four video feeds have been processed as different processes and data has been shared by all the child processes and the main program. Video feed with an ambulance has been used in the second round of cycle in assigning green timers to all the lanes. This code also contains the for loop that checked whether a particular lane in each cycle has been turned green. If the lane has not been turned green then the value of the processing bool would be “False”. Then in the next round the lane that was not processed was turned green but the priority was given to the lane with an ambulance in it.

5.3.3 Project Deployment

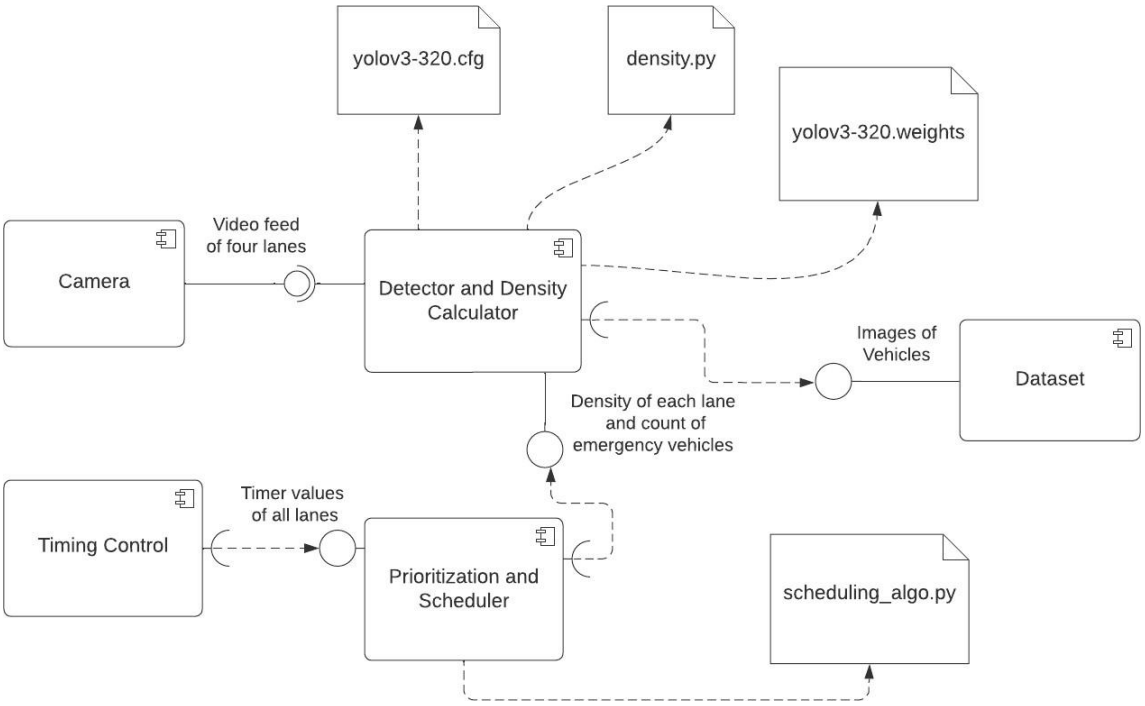
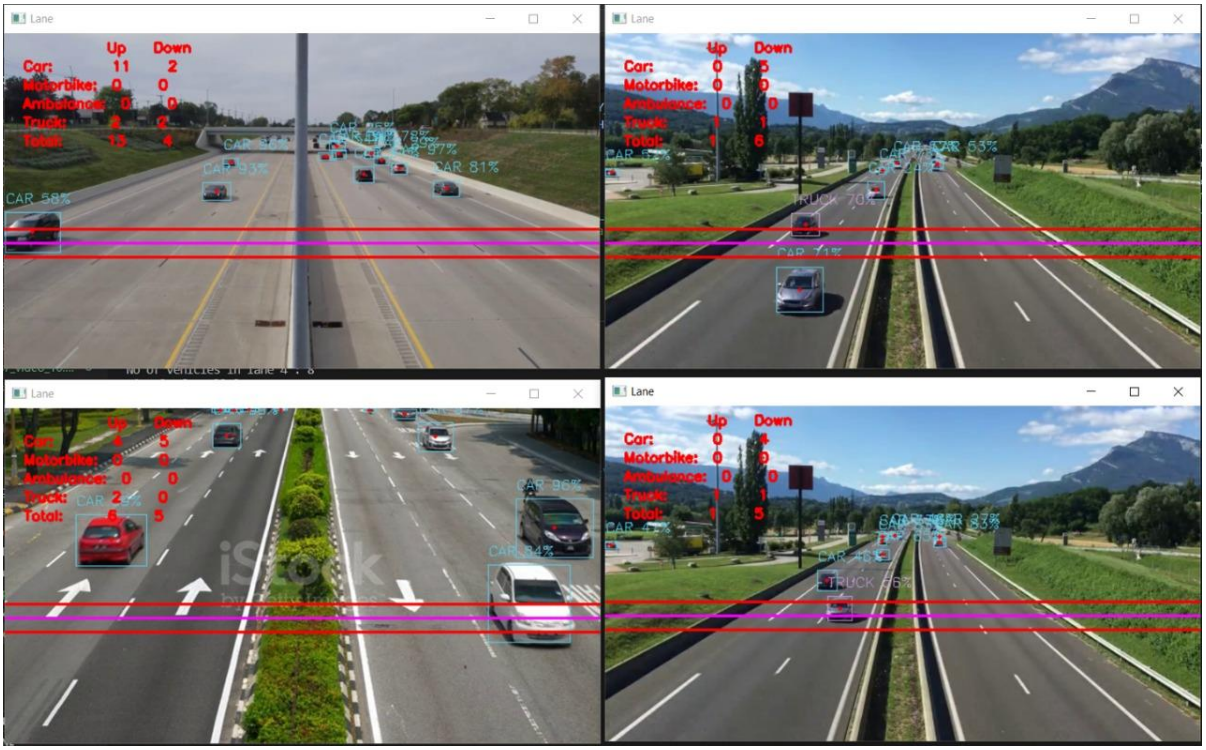


Figure 13: Project Deployment Diagram

The diagram in figure 13 contains five components. First component is camera which provides video feed of all the four lanes. Second component is Detector and Density Calculator which receives the input of video feed of all four lanes. This component then detects the vehicles on the road by using images from the dataset for reference. Then it calculates the density of each road and sends the count of emergency vehicles in all the lanes to the third component. This component uses YOLO v3-320.cfg and YOLO v3-320.weights files for detecting vehicles on road and it uses density.py fileto calculate density of vehicles on road.

5.3.4 System Screenshots

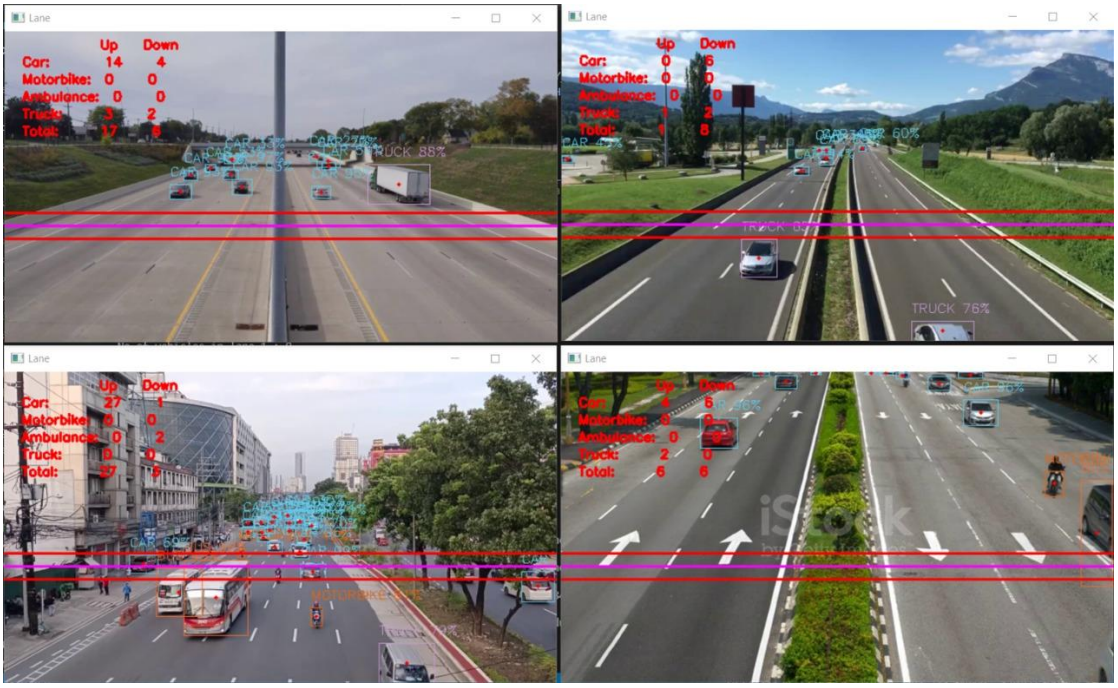
Snapshots of Working Prototype




```
No of vehicles in lane 1 : 8
No of vehicles in lane 2 : 6
No of vehicles in lane 3 : 6
No of vehicles in lane 4 : 8
Signals for all lanes
Signal for lane 1 Green   and   Green Timer =  6
Signal for lane 2 Red
Signal for lane 3 Red
Signal for lane 4 Red
For current Green 0
```

FIGURE 12 (i) Demonstration of 4-lane intersection along with green signal timings

Figure 12 (i) is basically the 4-lane intersection background for demonstration of simulation. After processing for a particular time, the vehicle count in each lane is displayed and the first lane's green signal timing is set.




```
No of vehicles in lane 1 : 8
No of vehicles in lane 2 : 6
No of vehicles in lane 3 : 6
No of vehicles in lane 4 : 3
Signals for all lanes
Signal for lane 1 Red
Signal for lane 2 Green   and   Green Timer =  5
Signal for lane 3 Red
Signal for lane 4 Red
For current Green 1
```

FIGURE 12 (ii) Simulation of vehicles turning

Simulation in figure 12 (ii) presents the number of vehicles in each lane and this time the green signal timing for the second lane is set.

```
No of vehicles in lane 1 : 8
No of vehicles in lane 2 : 6
No of vehicles in lane 3 : 6
No of vehicles in lane 4 : 8
Signals for all lanes
Signal for lane 1 Red
Signal for lane 2 Red
Signal for lane 3 Red
Signal for lane 4 Green   and   Green Timer =  6
For current Green 3
```

FIGURE 12 (iii)

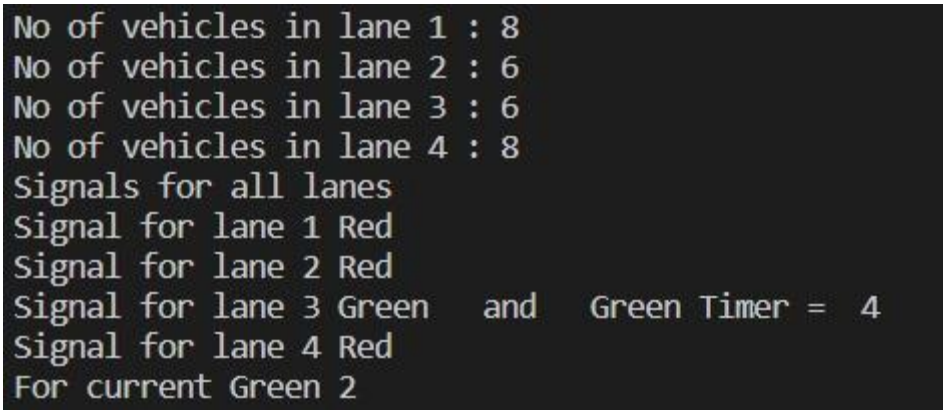


FIGURE 12 (iv) Pre-emption on the basis of density

In figure 12 (iii) and (iv), instead of setting green signal timings for the third lane, the system set the fourth lane on green because of detection of priority vehicles.

5.4 Testing Process

5.4.1 Test Plan

The test plan has been designed in close agreement with the objectives and the performance parameters. In PyGame simulation, initially the individual modules - vehicle generation, set time function are tested as a standalone unit and after resolving any bugs or errors, different modules are to be integrated and tested. Similarly, in vehicle detection through video feed, firstly object detection on video inputs has been tested. Then dynamic and prioritised scheduling algorithm incorporated in the code has been tested.

5.4.2 Features to be tested

The features to be tested for the Pygame simulation include vehicle generation module, move function and set time module. Vehicle generation module includes generation vehicle of a particular type in one of the four lanes randomly after every second. Move function includes the program that would manage the movement of vehicles and estimate the gap that they would maintain among themselves in all the four lanes. Set time module sets the green timer of each lane as per the density of each lane. It keeps the track of the number of vehicles of each type in every lane. The features to be tested for the vehicle detection in video feeds include vehicle detection module and scheduling algorithm function. Vehicle detection module contains detector that detects the presence of each type

of vehicle, and the tracker that ensures whether the vehicle in every frame of video is the same vehicle and then the counter that counts the number of vehicles.

5.4.3 Test Strategy

1. Unit Testing: At this level, every module was tested independently to validate working as per the objectives and the design of the module. This included all the features mentioned in Section 5.4.2.
2. Integration Testing: At this level, individual modules were grouped and the integrated system was then tested.
3. System Testing: After integrated testing, the whole system was checked and validated. This level of testing is known as system testing.
4. Acceptance Testing: The whole system was tested and validated by matching with the approved objectives (Section 1.7). This level of testing is termed as acceptance testing.

5.4.4 Test Cases

Test Case Number	Module	Type of test	Testing Criterion	Test Results
1.	Vehicle Generation	Functionality test	This module should ensure generation vehicle of a particular type in one of the four lanes randomly after every second	Passed
2.	Set Time Module	Functionality testing	It should set the green timer of each lane as per the density of each lane and incorporate priority scheduling. It keeps the track of the number of vehicles of each type in every lane.	Passed
3.	Move function	Functionality testing	It should manage the movement of vehicles according to the signals and the presence of vehicles ahead by maintaining the distance and stopping gap	Passed
4.	Vehicle detection which includes detector, tracker and counter in the module with video feed as an input	Functionality test	This module should ensure proper identification of vehicles and keeping track of the vehicle along with maintaining their counts	Passed

5.	Scheduling module for dynamic and prioritized scheduling	Functionality test	This module handles the main objective of the project which is scheduling based upon density of each lane and prioritizing according to the presence of the emergency vehicle	Passed

5.4.6 Test Results

The test results mentioned above have been successfully tested and giving the desired results.

5.5 Results and Discussions

Research Papers	Dynamic scheduling based on density	Density calculation using Image processing	Prioritized scheduling for emergency vehicles	Density calculation and ambulance detection using sensors
Khushi [3]	Yes	Yes	No	No
Kumar at al. [4]	No	No	Yes	Yes
Soman at al. [7]	Yes	Yes	No	No
Kanungo at al. [11]	Yes	Yes	No	No
Gandhi at al. [12]	Yes	Yes	No	No
Vogel at al. [5]	Yes	Yes	No	No
Faye at al. [13]	Yes	No	No	Yes
Zaid at al. [14]	Yes	Yes	No	No
Barbosa at al. [8]	Yes	Yes	No	No
SMRUPS	Yes	Yes	Yes	No

5.6 Inferences Drawn

Our task fulfills all the fundamental usefulness as we need according to project targets yet there is a lot to be improved and different new highlights should be added to make

the client experience more charming and surpass assumptions. The inferences drawn from the test results and by assessing the exhibition boundaries are:

- In a real-time system, the model will accomplish the different stages after taking the video input from the cameras, keep the count of number of vehicles in each lane and setting the green signal timing accordingly.
- A prototyping model with respect to the working of genuine model is finished utilizing Pygame modeling, wherein recognition of priority vehicle (mainly ambulances) in one of the cases was obvious in the functioning depictions.
- By improving the progression of traffic will diminish normal clog. That implies less holding up time at crossing points and lower emanations, expanding the air quality.
- Prioritizing traffic in view of changes in rush hour gridlock conditions progressively. This model will further develop traffic security, on the grounds that questionable velocities, weighty traffic can all bring about mishaps and passing.

5.7 Validation of Objectives

S. No.	OBJECTIVES	STATUS
1	To identify and classify various vehicles using object detection algorithms.	Successful
2	To develop a scheduling algorithm which controls the traffic light timings basedon traffic density and prioritization of emergency vehicles on each lane.	Successful
3	To create a simulated environment of a smart traffic control system.	Successful

CONCLUSION AND FUTURE SCOPE

6.1 Work Accomplished

The project idea of smart traffic control based on traffic density along with prioritized pre-emption has multiple parameters of the real time environment in play. By now we've achieved the goal of virtual simulation of the developed scheduling algorithm via Pygame. And we've also completed object detection and classification using YOLOv3 and OpenCV, further performing dynamic scheduling & prioritized pre-emption based on the vehicles detected and traffic density calculated in each lane. We've faced challenges but we're able to achieve the required objectives to reach the endgame.

6.2 Conclusions

Our proposed system aims to utilize live images from the cameras at traffic junctions for traffic density calculation using image processing and AI. It also focuses on the algorithm for scheduling the timing of traffic lights based on the vehicle density in a particular lane to reduce congestion, thereby providing faster transit to people. The system also prioritizes emergency vehicles in a lane and schedules more time for them to pass as soon as possible. The project consists of 3 parts namely Vehicle Detection and Density Calculation Module, Prioritization and Signal Switching Scheduling Algorithm and Simulation module that helps to simulate the results and change the timers.

Our proposed project tends to act as a baseline system to build more complex systems that can enrich the research in the area of smart traffic control and can promote the implementation of the system in a more presentable manner.

6.3 Environmental/ Economic/ Social Benefits

One of the most relevant and popular modern world problems is the **progressively increasing traffic congestion on the roads** of our country. With the advancement in technological development and socioeconomic boost in the automobile market, a severe impact has been observed on the traffic conditions across the nation. Considering the fact India's road network (including national highways etc.) has grown by just about a third in the last decade whereas vehicle registrations have increased by almost three times. Thus, vehicle density is increasing at a much faster pace than road length - obviously, congestion will be higher. Our proposed solution uses the technological advancements in the best possible way to ensure maximum efficiency and minimize the chances of delay in critical situations as well as introduces a upgraded level of efficiency in terms of traffic control by considering vehicle density in the process of dynamic traffic light timing scheduling. According toa report published by Times of India about 146,133 people were killed in roadaccidents in India in the year 2016. Unfortunately, about **30% of deaths are caused due to delayed ambulances. Another Indian government data shows that more than 50% of heart attack cases reach hospital late**, which can constitute unavailability of ambulances too but majority of it is due to patients stuck in traffic. These are exactly the situations where our solution would operate and help preventthe occurrence of such unfortunate events. Our solution is capable of helping anyone and everyone with an automobile in traffic congestion prone areas at the same time. That shows the scope of the solution to be really vast and valuable.

6.4 Future Work Plan

The system prototype is in its initial phase of development. Our major aim would be to get it ready for deployment on a large scale and under real-time conditions & challenges. In the coming future, we'll try to elevate the system design and try to reduce the processing time as much as we can so that our product works in real-time with an enhanced user experience in the future. Apart from that, we wish to workupon the performance and operation of our system in harsh and unreliable weather conditions with fog, mist and storms. The operation of the system is currently highly dependent on visibility, we plan to remove this constraint as well and increase the system's operation functionality for conditions with a lack of visibility. We also have plans for adding up various other functionalities like adding customized dataset for the detection of other priority vehicles, finding other techniques for density calculation in order to optimize the performance of the model, taking input as live feed from the cameras and hardware implementation of the project.

7.1 Challenges Faced

- The computational processing of the project was lowering down in a CPU and hence we have to use GPU computing
- We had to update the dataset according to our requirements as there were no emergency vehicles present earlier.
- Integrating the object detection source along with priority scheduling and hence providing corresponding green signal timings to the lights.

7.2 Relevant Subjects

- Machine Learning
- GPU computing
- Open CV
- Artificial Intelligence
- Pygame
-

7.3 Interdisciplinary Knowledge Sharing

Apart from using knowledge gained through subjects which had been taught in the course curriculum, the team has gained a lot of information as well as knowledge from other sources which are required in the completion of this project. The various other disciplines which are included in this project are regarding the usage of object detection algorithm and its integration with the Machine Learning libraries to achieve the desired functionality. The team majorly learned how to use, configure, code and integrate various code modules, to implement the system so that it can work more efficiently and provide better results than the current solutions present in the market.

7.4 Peer Assessment Matrix

		Evaluation of			
		Apurvi	Japleen	Pramit	Shaurya
Evaluation by	Japleen	5	5	5	5
	Pramit	5	5	5	5
	Shaurya	5	5	5	5
	Apurvi	5	5	5	5
Total		20	20	20	20

7.5 Role Playing and Work Schedule

The following are the roles of different team members in the development of the project.

1. Apurvi Garg
    - Simulating the traffic junction through pygame
    - Incorporating emergency vehicles prioritization
    - Design Optimization
2. Japleen Kaur
    - Study of YOLO library used for object detection
    - Simulating the traffic junction through pygame
    - Design Optimization
3. Pramitdeep Gogna
    - Applying object detection algorithm on vehicles and testing
    - Simulating the project using video feed as an input
    - Designing signal switching algorithm based on the traffic density
4. Shaurya Pratap
    - Study of YOLO library used for object detection
    - Documentation
    - Applying object detection algorithm on vehicles and testing

7.6 Student Outcomes (SO) Description and Performance Indicators (PI)

SO	SO Description	Outcome
1.1	Ability to identify and formulate problems related to computational domain	The formulation of YOLOv3 for object detection was accomplished.
1.2	Apply engineering, science, and mathematics body of knowledge to obtain analytical, numerical, and statistical solutions to solve engineering problems.	Basic principles of science and mathematics body of knowledge which will be the better way to capture video helped in solving the engineering problems
2.1	Design computing system(s) to address needs in different problem domains and build prototypes, simulations, proof of concepts, wherever necessary, that meet design and implementation specifications.	The design was optimized taking into consideration the processing time and hence GPU computing was used.
2.2	Ability to analyze the economic trade-offs in computing systems.	The economic trade-offs were kept in mind and hence the cost of sensors is reducted from the project and live video feed was used.
3.1	Prepare and present variety of documents such as project or laboratory reports according to computing standards and protocols.	We gave two presentations for this project involving all the team members and work was effectively distributed among the team members, the presentation was done quite professionally

		following all the norms as well as was conducted in a well-organized manner
3.3	Able to communicate effectively with peers in well organized and logical manner using adequate technical knowledge to solve computational domain problems and issues.	Every team member works efficiently in accordance with the task assigned and the associated deadline. Also, each team member discussed the status of each other's work progress on a weekly basis.
4.1	Aware of ethical and professional responsibilities while designing and implementing computing solutions and innovations.	The project is based on traffic control management and hence there is no violation of traffic rules and mental health caused due to the chaos in congestion is taken care of in the outcomes.
4.3	Evaluate computational engineering solutions considering environmental, societal, and economic contexts.	The simulation in Pygame is a prototype to the proposed idea, wherein the challenges faced in real life are encountered and hence the green signal timings are displayed.
5.1	Participate in the development and selection of ideas to meet established objective and goals.	Team work, accomplishing work in limited time, presentation skills, technical skills, dealing with various errors with patience will help the team in future.
5.2	Able to plan, share and execute task responsibilities to function effectively by creating collaborative and inclusive environment in a team.	The team has used various YouTube videos and online courses for the implementation of our project.
6.1	Ability to perform experimentations and further analyze the obtained results.	The testing was done at every stage and the results were analyzed thoroughly, the errors were taken into consideration and taken care of.
6.2	Ability to analyze and interpret data, make necessary judgement(s) and draw conclusion(s).	The data collection was done from verified sources and tested for complete data integrity.
7.1	Able to explore and utilize resources to enhance self-learning.	The knowledge of object detection and priority scheduling on real time video was explored and the model is trained for prioritizing emergency vehicles.

7.7 Brief Analytical Assessment

Q1. What sources of information did your team explore to arrive at the list of possible Project Problems?

Ans1. The idea for the project was induced from real-life problems which was reducing the waiting time of people while travelling and the primary reason for the same cause is the congestion in traffic. Therefore, data was collected from verified sources to understand the problem better and come up with the most efficient answer.

Q2. What analytical, computational and/or experimental methods did your project team use to obtain solutions to the problems in the project?

Ans2. The conventional traffic systems are complex and non-linear in nature because they are timing dependent and not traffic dependent. So, the idea of making a traffic control system traffic dependent was by calculating the density of traffic and accordingly providing signal timings which will be based on the density.

Q3. Did the project demand demonstration of knowledge of fundamentals, scientific and/or

engineering principles? If yes, how did you apply?

Ans3. Yes, the project demanded demonstration knowledge of fundamentals, scientific and/or engineering principles. The object detection for the real video feed was done using YOLOv3 which is the most accurate and widely used Machine Learning model. This model was then integrated with the priority scheduling source code and based on that green signal timings were awarded to the traffic lights. The priority scheduling is based on 2 major factors: the density of a particular lane and presence of emergency vehicle in any lane. This system was simulated in Pygame.

Q4. How did your team shares responsibility and communicate the information of schedule with others in team to coordinate design and manufacturing dependencies?

Ans4. The project was divided equally in four group members with effective communication at every level of the project. The team has used various YouTube videos and online courses for the implementation of the project. Peer evaluation was done after discussing the source code, the shortcomings and were worked upon in order to increase the efficiency and accuracy of the project.

Q5. What resources did you use to learn new materials not taught in class for the course of the project?

Ans5. Various research papers were used in order to understand the problem and discussions were held in order to provide the optimized solution to the problem. Learning the use of YOLOv3 to detect objects and integration of the codes was accomplished with

the help of Youtube and fellow mates. Data from verified sources was also analyzed using MS Office suite.

Q6. Does the project make you appreciate the need to solve problems in real life using engineering and could the project development make you proficient with software development tools and environments?

Ans6. Project development is the best way to solve real-life problems and the only way to make our team efficient with the software development tool environment. To bring bookish knowledge to a real-life unit, the only way is project development on real-life problems.

APPENDIX A: REFERENCES

[1] TomTom International BV, “Traffic congestion ranking: Tomtom traffic index,” Traffic congestion ranking | TomTom Traffic Index. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/ranking/?country=IN. [Accessed: 16-Dec-2022].

[2] R. Soman and K. Radhakrishnan, ‘Traffic Light Control and Violation Detection Using Image Processing’, Traffic, vol. 8, no. 4, 2018.

[3] M. S. Shinde and S. R. Jagtap, ‘Intelligent Traffic Management Systems’, A Review, 2016.

[4] “Vahan Sewa: Dashboard,” VAHAN SEWA| DASHBOARD. [Online]. Available: https://vahan.parivahan.gov.in/vahan4dashboard/vahan/dashboardview.xhtml;js%20essionid=27C8%2093696318E5EAD0D21F6220736EB1. [Accessed: 25-Dec-2022].

[5] 2019 Debabrata Mohapatra / TNN / Oct 12, “City to roll out 'smart traffic' system: Bhubaneswar News - Times of India,” The Times of India. [Online]. Available: https://timesofindia.indiatimes.com/city/bhubaneswar/city-to-roll-out-smart-traffic-system/articleshow/71546615.cms#:~:text=BHUBANESWAR%3A%20More%20than%20a%20year,busy%20KIIT%2DJayadev%20Vihar%20route. [Accessed: 25-Dec-2022].

[6] “Ahmedabad to get India's first Intelligent Traffic System,” The Economic Times. [Online]. Available: https://economictimes.indiatimes.com/news/economy/infrastructure/ahmedabad-to-get-indias-first-intelligent-traffic-system/articleshow/44758284.cms. [Accessed: 25-Dec-2022].

[7] www.ETGovernment.com, “Madhya Pradesh: Indore Smart City set to implement Intelligent Transport Management System, Government News, et government,” ETGovernment.com, 20-Oct-2019. [Online]. Available: https://government.economictimes.indiatimes.com/news/smart-infra/madhya-pradesh-indore-smart-city-set-to-implement-intelligent-transport-management-system/71670251. [Accessed: 25-Dec-2022].

[8] R. Carvalho Barbosa, M. Shoaib Ayub, R. Lopes Rosa, D. Zegarra Rodríguez, and L. Wuttisittikulkij, “Lightweight PVIDNet: A Priority Vehicles Detection Network Model Based on Deep Learning for Intelligent Traffic Lights,” Sensors, vol. 20, no. 21, p. 6218, Oct. 2020, doi: 10.3390/s20216218.

[9] Khushi, "Smart Control of Traffic Light System using Image Processing," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 99-103, doi: 10.1109/CTCEEC.2017.8454966.

[10] V. Kumar et al., ‘AI Powered Smart Traffic Control System for Emergency Vehicles’, in ICDSMLA 2020, Springer, 2022, pp. 651–663.

[11] A. Vogel, I. Oremović, R. Šimić and E. Ivanjko, "Improving Traffic Light Control by Means of Fuzzy Logic," 2018 International Symposium ELMAR, 2018, pp. 51-56, doi: 10.23919/ELMAR.2018.8534692.

[12] M. M. Gandhi, D. S. Solanki, R. S. Daptardar and N. S. Baloorkar, "Smart Control of Traffic Light Using Artificial Intelligence," 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2020, pp. 1-6, doi: 10.1109/ICRAIE51050.2020.9358334.

[13] S. Faye, C. Chaudet and I. Demeure, "A distributed algorithm for adaptive traffic lights control," 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012, pp. 1572-1577, doi: 10.1109/ITSC.2012.6338671.

[14] A. A. Zaid, Y. Suhweil and M. A. Yaman, "Smart controlling for traffic light time," 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2017, pp. 1-5, doi: 10.1109/AEECT.2017.8257768.

[15] A. Kanungo, A. Sharma and C. Singla, "Smart traffic lights switching and traffic density calculation using video processing," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), 2014, pp. 1-6, doi: 10.1109/RAECS.2014.6799542.

[16] M. Sagar, “AI-powered traffic management to become reality in Chandigarh by February 2022,” Hindustan Times, 01-Oct-2021. [Online]. Available: https://www.hindustantimes.com/cities/chandigarh-news/aipowered-traffic-management-to-become-reality-in-chandigarh-by-february-2022-101633125164747.html. [Accessed: 17-Dec-2022].

[17] N. Singh, "Nhai invites bids for Advanced Traffic Management System for delhi-mumbai E-way as work in last lap," *News18*, 12-Dec-2022. [Online]. Available: https://www.news18.com/news/india/nhai-invites-bids-for-advanced-traffic-management-system-for-delhi-mumbai-e-way-as-work-in-last-lap-6595237.html. [Accessed: 17-Dec-2022].

[18] S. Ye, 'Research on urban road traffic congestion charging based on sustainable development', *Physics Procedia*, vol. 24, pp. 1567–1572, 2012.

[19] M. Asad, R. M. Yasir, D. Nower, D. Shoyaib, and Others, 'Traffic Congestion Prediction Using Machine Learning Techniques', *arXiv preprint arXiv:2206. 10983*, 2022.

[20] S. R. Samal, P. G. Kumar, J. C. Santhosh, and M. Santhakumar, 'Analysis of Traffic Congestion Impacts of Urban Road Network under Indian Condition', *IOP Conference Series: Materials Science and Engineering*, vol. 1006, no. 1, p. 012002, Dec. 2020.

APPENDIX B: PLAGIARISM REPORT

Document Information

Analyzed document	plag checking.pdf (D153810125)
Submitted	2022-12-18 14:36:00
Submitted by	Tarunpreet
Submitter email	tarunpreet@thapar.edu
Similarity	7%
Analysis address	tarunpreet.thapar@analysis.urkund.com

Sources included in the report

W	URL: https://github.com/mihir-m-gandhi/Adaptive-Traffic-Signal-Timer Fetched: 2022-04-13 08:23:08	5
SA	**Thesis_First_Try (3).pdf** Document Thesis_First_Try (3).pdf (D136044057)	8
SA	**UBUS (Project Report) Dissertation.docx** Document UBUS (Project Report) Dissertation.docx (D138217208)	12
SA	**205214329_K.Vasanth_II MSc CS C.pdf** Document 205214329_K.Vasanth_II MSc CS C.pdf (D138688287)	1
SA	**Group7report (1) (5).docx** Document Group7report (1) (5).docx (D109198989)	1