MAJOR PROJECT ON

TRAFFIC ANALYSIS AND MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements of the degree of

(Bachelor of Engineering)

Supervised by **Dr. Suvarna Pansambal**

Under – taken by

Paras Chheda

Kedar Damani

Bhavin Gajjar

Computer Engineering

Atharva College of Engineering

(2020-21)

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: dd/mm/yyyy **(Name and Signature of Students)**

Project Report Approval for B.E.

This project report entitled **“*Traffic Analysis and Management System*”** by ***Paras Chheda, Kedar Damani, Bhavin Gajjar*** is approved for the degree of ***Bachelor of Engineering in Computer Engineering.***

Internal Examiner

External Examiner

Date: __________

Place: __________

College Seal

ATHARVA
COLLEGE OF ENGINEERING

AET'S

ATHARVA COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that

Paras Chheda

Kedar Damani

Bhavin Gajjar

Have satisfactorily completed the requirements of the B.E Project Report

On

"TRAFFIC ANALYSIS AND MANAGEMENT SYSTEM"

As prescribed by the ***University of Mumbai*** *Under the guidance of*

Guide

Dr. Suvarna Pansambal

Dr. Suvarna Pansambal	**Dr. Suvarna Pansambal**	**Dr. S.P.Kallurkar**
Project Coordinator	**Head(Computer Engineering)**	**Principal**
(April 2021)	**College Seal**	**Date:**

ACKNOWLEDGEMENT

We would like to thank our project guide **Dr. Suvarna Pansambal** for her enormous cooperation and guidance. We have no words to express our gratitude for a person who wholeheartedly supported the project and gave freely of her valuable time while making this project. All the inputs given by her found a place in the project. The technical guidance provided by her was more than useful and made this a success. She has always been a source of inspiration for us. It was a great experience learning under such a highly innovative, enthusiastic and hardworking professor. We are also thankful to our principal **Dr. S.P. Kallurkar**, project coordinators and all the staff members of the Computer Department who have provided us various facilities and guided us throughout to develop this project idea. Finally, we would like to thank the teachers of our college and friends who guided and helped us while we were working on this project idea.

ABSTRACT

As per a recent survey, the vehicle density in larger cities is about 28,380 vehicles per lakh population and 115 vehicles per kilometer road length. The total number of vehicles on the road is at 3.49 crore, of which 35.75 lakh vehicles or 10.2% are in Mumbai and its surrounding township. The traffic congestion problem is exacerbating in major cities of India. 'Traffic Analysis and Management System' aims to reduce this issue with a feasible solution. By utilizing the concepts of Image processing and object detection, we are calculating the number of vehicles at each traffic signal. This data will be further used to determine the total waiting time at each stop. While calculating the red-signal window, we used a dynamic algorithm, which can reduce the waiting time at less crowded signals. Moreover, the data collected is used for analysis purposes, as well. The information is then fed to a machine learning model, which is responsible for handling the traffic density analysis and prediction. This method requires heavy dependency on surveillance cameras but, since Mumbai and other developed cities are already equipped with CCTV cameras at most of the road junctions, it is the most practical solution for dealing with the traffic congestion problem.

INDEX

Sr. No.	Contents	Page No.
1	Introduction	6
1.1	Need and Motivation	7
1.2	Basic Concept	8
1.3	Applications	8
1.4	Market Potential and Competitive advantages	8
2	Review of Literature	9
3	Report on the Present Investigation	12
4	Aim and Objectives	13
5	Problem Statement.	14
6	Proposed System	15
7	Requirement Analysis	16
7.1	Functional Requirement	16
7.2	Non-Functional Requirement	16
8	Scope and Feasibility	17
9	Design Details	18
9.1	Context Level diagram	18
9.2	DFD Diagram	19
9.3	Sequence Diagram	20

9.4	E-R Diagram	21
10	Implementation Plan	22
10.1	Hardware Requirements	22
10.2	Software Requirements	22
10.3	Timeline	23
11	Methodology	24
12	Conclusion	26
13	Acknowledgement	27
14	Literature Cited	28

LIST OF FIGURES

Figure No.	Name of Figure	Page No.
6.0	System Representation	15
9.1	Context Level Diagram	18
9.2	Data Flow Diagram	19
9.3	Sequence Diagram	20
9.4	Entity Relation Diagram	21
10.1	Tentative Timeline	23

LIST OF TABLES

Figure No.	Name of Table	Page No.

1. INTRODUCTION

The 'Traffic Analysis and Management System' (TAMS) is a real-time application of Computer vision, focusing on solving the traffic congestion problem. Computer vision concepts give computers the ability to function as human eyes. Due to the recent advances, it's possible to integrate within laptops, mobile phones, and other electronic devices such as CCTV cameras. Computer vision allows us to implement Image processing and object detection algorithms. Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be an image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies.

Furthermore, Object detection is a computer vision technique that works to identify and locate objects within an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in (or how they move through) a given scene. Broadly speaking, object detection can be broken down into machine learning-based approaches and deep learning-based approaches.

In more traditional ML-based approaches, computer vision techniques are used to look at various features of an image, such as the color histogram or edges, to identify groups of pixels that may belong to an object. These features are then fed into a regression model that predicts the location of the object along with its label.

On the other hand, deep learning-based approaches employ convolutional neural networks (CNNs) to perform end-to-end, unsupervised object detection, in which features don’t need to be defined and extracted separately.

1.1 Need and motivation

India is one of the fastest-growing economies in the world. The average income of Indians is growing, resulting in a rising number of privately owned vehicles. Though public transport is widely available in India, it is not sufficient for the huge population. Especially in Metro cities, often public transport services are crowded. Therefore, to travel peacefully, people are opting for commuting in their own vehicles. And as a result, more vehicles are coming on roads.

According to an April 18 report released by The Boston Consulting Group and commissioned by Uber, On an average, travelers in Delhi, Mumbai, Bengaluru, and Kolkata spend 1.5 hours more on their daily commutes than their counterparts in other Asian cities during peak traffic times. Calculating the cost of congestion based on fuel burned and productivity loss, which includes the man-hours and opportunity cost, pollution, and accidents (health costs) incurred on an annual basis, India's biggest cities may be losing up to $22 billion annually to traffic congestion, and its commuters are bearing the burden.

The proposed system will help in reducing the high volume of traffic problems by controlling the traffic signal opening and closing window. As opposed to aimless and incomplete construction of roads, we plan to utilize the available resources to its utmost efficiency.

1.2 Basic Concept

We propose an easy-to-implement solution for effective management. We are using CCTV cameras to count the total number of vehicles at each signal. The lesser the vehicle count, the lower will be the waiting period for that lane. Similarly, if there's high traffic density at one particular lane, that lane will be given a more green signal window for traffic to pass. The algorithm ensures that each junction has maximum flow of vehicles travelling through it.

1.3 Application

Reduces the average waiting time vehicles have to spend at each traffic signal. Manages traffic effectively and helps in reducing the traffic congestion problem. Gathers data for further analysis. Maintains records, which can help in predicting traffic hours for each junction.

1.4 Market potential and Competitive advantages

The product has a huge potential, especially in major cities like Delhi, Mumbai and Bangalore, where traffic is an enormous obstacle for daily commuters. Since no such technique is used currently, this will be an innovative approach towards resolving the problem. The ease of implementation further contributes towards the wide adaptability of the proposed system.

2. REVIEW OF LITERATURE

J. Raj et. al. focuses on traffic density estimation using location-based sensors in this paper. While making a prediction, factors like traffic volume and time mean speed is considered and machine learning algorithms, namely k-nearest neighbor and Artificial Neural Network are used [1].

S. Dey and M. Rahman, proposed a system for calculating vehicle count and speed from a video clip. Using image processing and artificial neural network, they also proposed to predict the future trend for traffic density. Background subtraction technique is used for finding the centroid of a vehicle for easy tracking process [2].

D. Sitaram et. al. used a rating-based approach for still image monitoring. Using the 'Temporal Variance' method, they compared one image with traffic to another image without traffic. Considering the distance between two vehicles and day or night time, they assigned a traffic rating out of 10 [3].

M. S. Uddin et. al. with this paper, proposed another image processing based technique for traffic density. However, this paper used the canny edge detection method to compare empty roads with roads full of vehicles and accordingly calculated the density of traffic [4].

M. V., V. V. R. and N. A., in this paper focuses on object detection method using frame extraction followed by a box filter based background estimation. After smoothening the rapid variation, moving vehicles are detected using a pixel-wise variation, and the region based convolution neural network is implemented for vehicle recognition purpose [5].

U. E. Prakash et. al. has implemented an image processing based approach for determining traffic density and further controlling traffic signal based on the calculated density. However, image enhancement techniques are used for the detection of vehicles by separating vehicles from background roads [6].

S. Dong and Y. Zhang, proposed an effective method for the identification of traffic bottlenecks using the Max-flow min-cut theorem. While implementing this algorithm, the paper suggests to find weak parts of the road and propose to divert flow from that terminal to allow maximum flow at each point [7].

S. A. Meshram and R. S. Lande, provides us information about three potential image processing based methods for vehicle detection, namely Kalman filter, pixel method and centroid method. Moreover, the results were obtained using CCTV surveillance cameras and hence, it provides a base for our proposal of detecting vehicles from a certain inclined angle [8].

Khushi, deals with area density based traffic management system using image processing. After converting an image from RGB to Grayscale, the total extracted area is calculated and based on that the traffic density is estimated. Furthermore, using an Arduino UNO the concept of dynamic time allocation technique was demonstrated [9].

V. Cherniy et. al. developed a general traffic management system based on the collection of a vast amount of data such as geographical location, time and place, and office hours. The main takeaway from this paper is that by having a sufficient amount of data, we can make a reasonable prediction regarding the traffic problems [10].

3. REPORT ON PRESENT INVESTIGATION

After investigating previous already proposed/established system we were able to identify few shortcomings.

- First of all, sensor-based density systems tend to have less accuracy as compared to image-based models. Since it is difficult to get a precise location of multiple vehicles in real-time, this approach is not feasible to cope up with the current traffic dense system.
- Secondly, many papers have suggested the area-based density measurement technique for estimating traffic congestion. However, this method might be useful for highway surveillance at a junction it is not really convenient.
- Furthermore, the vehicle count-based method is more suited as compared to the density-based method. Considering the fact that different vehicles have different mobility rates, it is erroneous to assume that the same density traffic will require the same amount of time to travel from one point to another.
- Moreover, there is no present system that provides a correct real-time analysis on present traffic conditions. These data can be useful for prediction of traffic rush hours and possible locations of congestions, which is currently not implemented.
- Additionally, the proposed algorithm 'You-Only-Look-Once_v4 (YOLOv4)' is a state-of-the-art algorithm for real-time object detections. With the additional enhancements in the latest version, this algorithms stands on the top for the most accurate algorithm for real-time object detections with considerably higher frame rates.
- Finally, the real-time object detection technique is missing in the current traffic management system. Since all traffic signals work on a traditional hardcoded waiting time allocating algorithm, there is a fundamental flaw of extra waiting time at less crowded signals, which can be overcome by implementing the proposed system.

3.1 Comparison of present and proposed systems

References	Positives	Negatives
Application of data mining techniques for traffic density estimation and prediction (k-nearest neighbour) [1].	Uses sensor-based monitoring system. Utilises multiple factors for considering the traffic density and provides real-time estimations of traffic.	Not capable enough to compute traffic that is parallel to each other. The algorithm has low precision value since it does not provide exact numbers.
Application of Image processing and data mining techniques for traffic density estimation and prediction [2].	Provides traffic density estimation and vehicle count based calculations. Also, uses ANN based prediction model.	This might not be a practical approach for narrow roads or tightly packed junctions.
Still Image processing techniques for intelligent traffic monitoring [3].	Suitable to give fast and accurate results. Incorporates image comparison using temporal variance method	Could be difficult to implement in real-time traffic monitoring system as the system is more suitable for observing the traffic conditions.
Real-time area-based traffic density estimation by image processing for traffic control system [4].	Implements real-time and efficient image processing algorithms and uses image enhancement techniques.	It involves comparing two images for estimation of traffic density. This might give us an estimation of traffic density but still has less accuracy as compared to object counting.
A Deep learning RCNN approach for vehicle recognition in traffic surveillance system [5].	Uses RCNN based frame extraction followed by box filter-based background estimation for object detection.	Due to involvement of multiple overlapping layers the live performance induces certain extent of noise which results in performance degradation.

Density based traffic control system using image processing [6].	Capable of calculating total number of car occupying certain area and makes a reasonable prediction.	It has similar results as mentioned in previous density-based model yet might not be sufficient for implementing real-time.
Traffic surveillance by using image processing [8].	Proposes the use of CCTV camera for monitoring purpose and compares results of different algorithms.	It is capable of detecting traffic but does not proposes efficient algorithm for resolving the traffic.
Smart control of traffic light system using image processing [9].	Uses adequate algorithms for congestion and provides reasonable solution for it.	The algorithm is not adaptive in nature and hence probably not capable of producing accurate results.

Table 3.1 Comparative study of present investigations

4. AIM AND OBJECTIVES

4.1 Aim

The proposal for 'Traffic Analysis and Management System' aims to primarily reduce the unnecessary waiting time commuters have to spend at less crowded lanes, especially during the peak hours of the day at dense junctions, and provide a reliable mechanism to handle the increasing traffic congestion problem.

4.2 Objectives

- The objective of this project is to manage the traffic on the road in an effective manner.
- Implementation of this project will reduce the average traffic hours commuters have to spend on a daily basis.
- The system will be useful for monitoring as well as traffic analysis purposes and the data gathered can be further processed to get an estimated trend of traffic.

5. PROBLEM STATEMENT

The relentless growth in the number of vehicles has resulted in the traffic congestion problem. The situation is aggravated during peak hours, especially on multi-lane cross-junctions. The current traffic management system is not capable of handling the overload effectively, and hence the commuters have to spend more and more time in long traffic queues. As the existing system allows a static waiting time for each traffic signal, it is not adaptive to the present traffic condition. The situation demands an advanced solution capable of producing efficacious results.

6. PROPOSED SYSTEM

The proposed system uses the surveillance camera present at each traffic signal and captures the live frame of traffic at that place. We further use this frame to apply the Image processing technique, and then using object detection, the number of vehicles will be calculated. This data is further processed under an algorithm, which is responsible for providing an estimated red signal window at each signal. After considering the traffic density, an appropriate signal timer will be assigned such that it allows maximum flow through that point. In simple words, the lane with less traffic will have a smaller green signal window as compared to other more packed lanes. The capture data will be used for analysis purposes, as well. Using Artificial Neural Network (ANN), we will predict a trend for traffic congestion. The data collected over time will help in making a reasonable prediction, which will further improve the efficiency of this system. By implementing this system, we can ensure that the traffic congestion problem is handled without spending a significant amount of money. As major junctions in cities like Mumbai are preoccupied with CCTV cameras, the cost estimation for this method is much less. Hence, this is a feasible solution for handling the vehicle congestion problem.

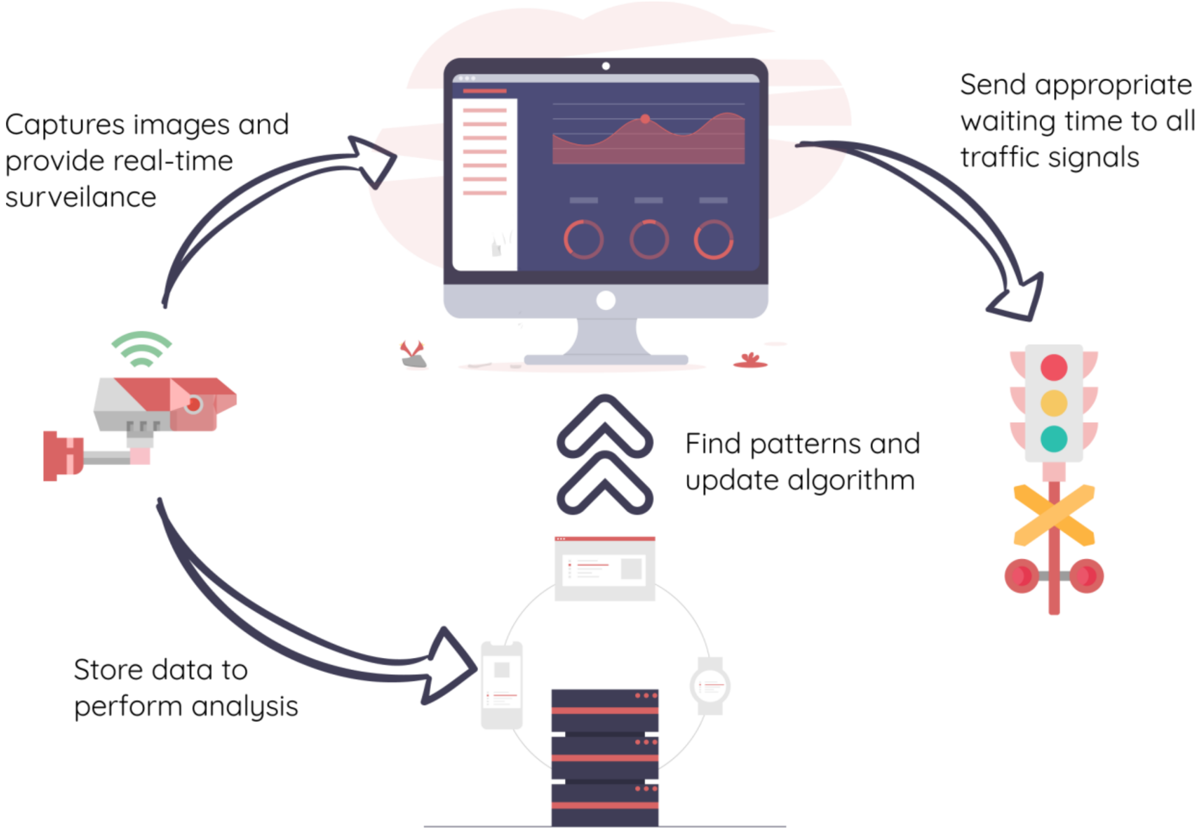

Captures images and provide real-time surveilance
Send appropriate waiting time to all traffic signals
Find patterns and update algorithm
Store data to perform analysis

Figure 6.0: System Representation

6.1 Components of the system

- CCTV Camera

  Cameras are the primary and only source of input for this system. Camera captures the real-time image of present traffic conditions and send that data to the control center for further processing. Only requirement for these cameras is an ability to capture and share high-definition images.

- Image Processing System

  This is a core component of the proposed system. It is a highly efficient system capable of performing real-time object detection. When trained over custom dataset as required, the algorithm can perform object (vehicle) counting with much accuracy and frames per second.

- Data Interpretation Unit

  This unit is mathematical brain of the system. It is responsible for gathering all the data from Image Processing System and process it accordingly. After acquiring traffic density at each signal, Data Interpretation Unit (DTU) will pass that data through optimized algorithm responsible for generating traffic signal timers. Moreover, DTU will also be responsible for analysing traffic density using various mathematical functions.

- Traffic Signals

  Traffic Signals are the result of the entire system. All the calculations will be finally sent to Traffic Signals, which will display Waiting Time (Red Signal) and Green Signal duration accordingly.

7. REQUIREMENT ANALYSIS

7.1 Functional Requirements

This section focuses on the functional requirements of the project. Functional requirements are technical devices required, algorithms to handle calculations, the bandwidth requirement for data transmission, data storing and manipulation techniques. Since the project is primarily based on computer vision techniques, the prime functional requirements are image capturing cameras and high-end computers capable of processing input images.

7.2 Non-Functional Requirements

Non-functional requirements can help to improve the quality of the product, but their absence will not affect the implementation. Clearer the image more accurate the output will be. Hence, high-definition and high-speed cameras will drastically improve efficiency. Moreover, because of high-resolution images, additional features like number plate detection and identification will be possible. Additionally, by using on-camera chips and connecting them with a cloud system, we can reduce latency between transmission.

8. SCOPE AND FEASIBILITY

8.1 Scope

The proposed system has a wide application. Due to excessive traffic, the proposed system has wide applications, especially in crowded cities like Mumbai, Delhi and Bangalore. The current traffic management system is based on static time allocation. Hence, the proposed solution is more efficient. Because of its scalability and easy implementation, it will help in reducing the traffic congestion problem.

8.2 Feasibility

8.2.1 Operational Feasibility

Operational feasibility is the ability to handle operations and supervise the process for any additional manual analysis. Most of the things will be automated and software-based hence, a single person can manage to inspect over a wide network with minimal effort.

8.2.2 Technical Feasibility

Technical feasibility involves implementing the proposed system in real-life scenarios. The popularity of Image processing algorithms is increasing day by day, hence programming communities like python are equipped with the latest libraries, which can be directly implemented in the software. Depending upon the requirement, minor modifications can be done, but overall, the existing system is quite scalable to the proposed problem without many technical obstacles.

8.2.3 Economical Feasibility

Economic feasibility is the capital centred approach. It considers the estimated cost and statistical look out of a project. The cameras required for this project are expensive, but since many cities have already adapted CCTV-based surveillance, the money needed to spend on this part is significantly reduced. On the other hand, computer devices with some minimum requirements are mandatory, but without knowing the exact configuration of current systems, it's difficult to estimate the tentative cost required for upgrades (if any).

8.2.4 Legal Feasibility

Legal feasibility is an assessment that investigates whether any aspects of proposed system conflicts with legal requirements or any laws. Since this project is mainly proposed for government agency rather than any private company, dealing with legal feasibility is not an issue. Traffic details being crucial data should not be mishandled by any third-party agency. Existing CCTV cameras are already used for traffic monitoring; hence, the proposed system merely acts as an addon and should not conflict with any legal implications.

8.2.5 Schedule Feasibility

Schedule feasibility is defined as the probability of a project to be completed within its scheduled time limits. As this is an optimization system, the more time invested in gathering real-time data and training model will tend to yield better results but using available datasets for identification can be used to start implementing the system and later on as per availability of data the system can be scaled to achieve higher accuracy.

9. METHODOLOGY

9.1 Data Generation

The data generation is the first step towards the implementation of this project. As we are dealing with the real-time scenario, previously captured images can only be used for training or experimental purposes. We propose the use of CCTV surveillance cameras as our primary source of capturing the live traffic. Each junction has a set of cameras focusing in each direction, which will work as a collective source of information. Further, majorly there are two methods for transmitting this data. The first is capturing still images and storing them for further processing, and the second is applying the algorithm on the live camera frame. The first method sounds convenient. However, it has many latency issues, which is why we preferred to apply the algorithm on live footage.

9.2 Data Processing

Data processing is the second and most important stage of this project. Once the picture frame is sent over the transmission channel. It reaches the computer system, where further processing will be done on it. We proposed the use of Image processing and object detection algorithms. Previously few initiatives were taken in determining the image-based traffic density, but most of them compare the two images - One with traffic on it and another without it. In calculating density based on this approach, the fundamental flaw is the unidentified type of vehicle and the mobility factor. Analysing two scenarios in which a large vehicle and two smaller vehicles occupy the same area on the footage, but the fact that smaller vehicles are more agile as compared to larger ones is not considered. Hence, even if in both scenarios the area occupied by vehicles is the same, but depending on the type of vehicles in the test case, the end results might alter.

Considering the above flaw, we propose the use of an object detection technique to count the number of vehicles at each traffic signal. By doing so, we can analyse the exact type to test the data we are working on, which will drastically improve the accuracy of the algorithm. Using OpenCV, an open-source computer vision library available in python, we can realize real-time object detection.

9.3 Analysis and Prediction model

The data captured through cameras will serve another purpose of analysing the traffic trend. While sharing the live footage, the information regarding vehicle count and traffic density will be stored and a separate dataset will be created. This dataset will then be fed to a machine learning model, which will analyse this data and using Artificial Neural Network (ANN) predict the congestion trends in the local area. However, this information will not provide any immediate benefit, but in the long term, this data can be used for traffic optimization and finding the fastest route between two places.

9.4 Digital Traffic Signals based on Maximum Flow

The concept of maximum flow involves finding a feasible flow through a flow network that obtains the maximum possible flow rate. In simpler terms, it is the most effective utilization of available nodes (Traffic Signals) through which maximum flow (vehicles) passes in a given period of time. This is especially helpful while dealing with multi-lane traffic, where the flow of traffic is not regulated I.e. Number of vehicles going in one direction far surpasses the other lane. If we continue to use static timers then the number of vehicles in a crowded lane will keep on rising to result in more and more congestion. However, by implementing a dynamic green signal window, we can assign more time for traffic in the jammed lane to pass.

10. DESIGN DETAILS

The design part includes the following:

a. Context Level Diagram

b. Data Flow Diagram (DFD)

c. Sequence Diagram

d. Entity Relation Diagram

10.1 Context Level Diagram

The context level diagram represents the entire system as one high-level process and shows the relationship between different entities involved in the process. The entities involved in this project are described in the following diagram, namely, Data Collecting Cameras, Data processing Systems, Digital traffic signals and Analysis and Prediction model.

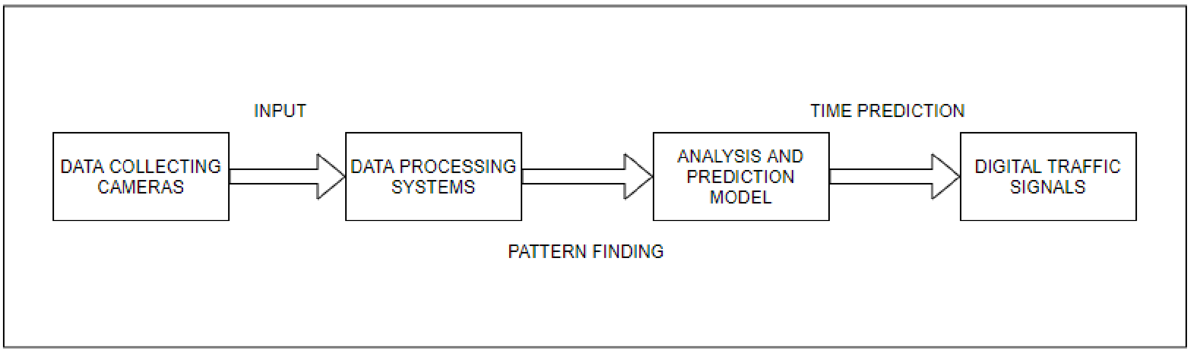


Figure 10.1: Context Level Diagram

10.2 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.

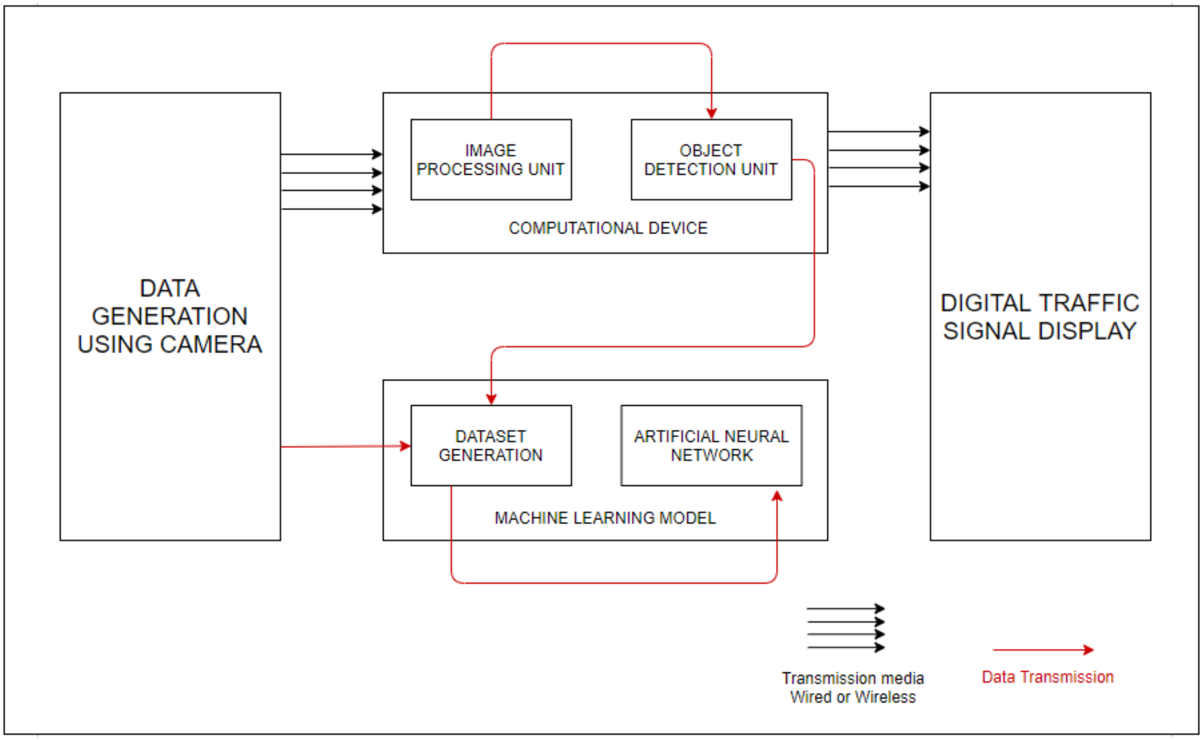

DATA GENERATION USING CAMERA
IMAGE PROCESSING UNIT
OBJECT DETECTION UNIT
COMPUTATIONAL DEVICE
DIGITAL TRAFFIC SIGNAL DISPLAY
DATASET GENERATION
ARTIFICIAL NEURAL NETWORK
MACHINE LEARNING MODEL
Transmission media Wired or Wireless
Data Transmission

Figure 10.2: Data Flow Diagram

10.3 Sequence Diagram

In the sequence diagram, the objects according to their time of execution are listed below. The sequence diagram deals with how the user will flow through the system sequentially and what are the processes that will take place. The objects interact with each other by sending messages to each other. As shown in the figure, the camera generates data by capturing the frame and this frame is passed through systems and processed further until an accurate result is obtained, which is sent to digital traffic signals.

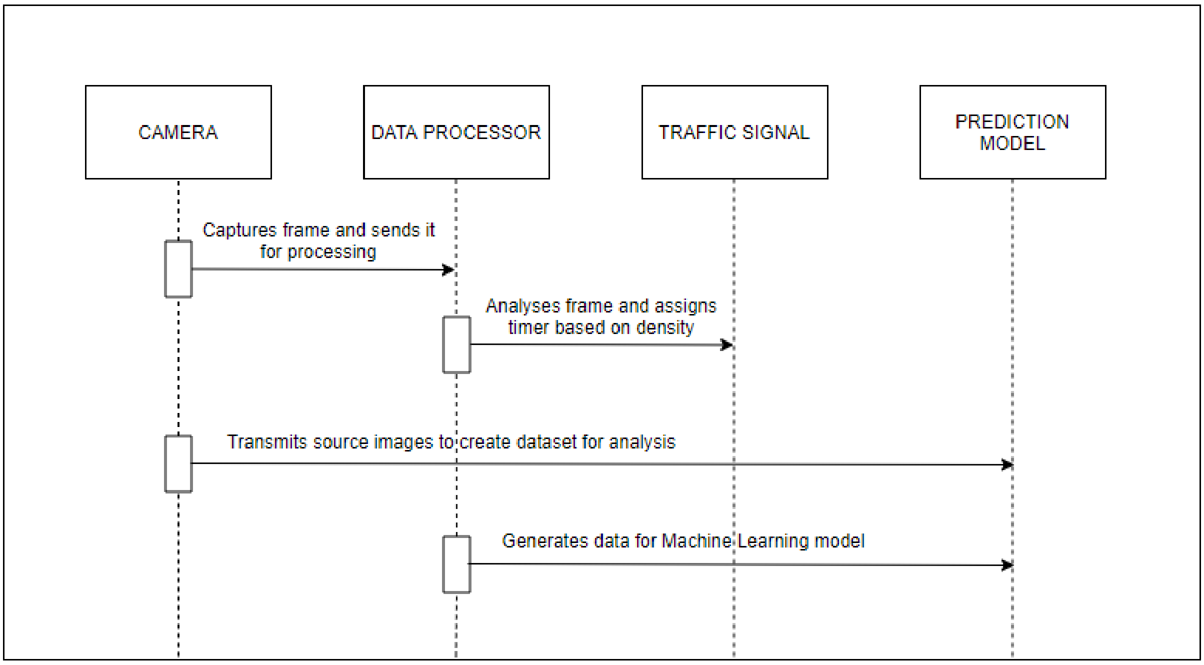

CAMERA
DATA PROCESSOR
TRAFFIC SIGNAL
PREDICTION MODEL
Captures frame and sends it for processing
Analyses frame and assigns timer based on density
Transmits source images to create dataset for analysis
Generates data for Machine Learning model

Figure 10.3: Sequence Diagram

10.4 Entity Relation Diagram

The main entities and how they are related to the other is shown in the diagram below. The entities and their key attributes are defined and what entities are interacting with each other for what purposes.

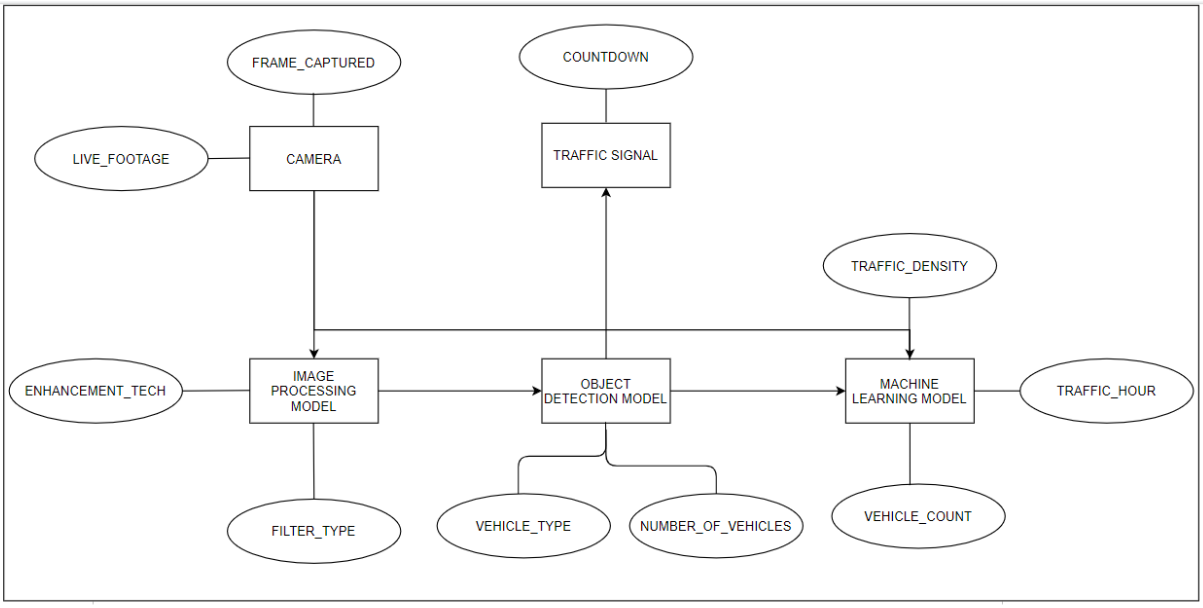


Figure 10.4: Entity Relation Diagram

11. IMPLEMENTATION PLAN

11.1 Hardware Requirements

Operating Systems: Windows 10 or Ubuntu

System RAM: 8 GB

System ROM: Preferably cloud storage or 1TB per slot

System GPU: Minimum GTX 710 or equivalent, Recommended GTX 1650 or higher

System Processor: Intel i5 or higher

Bandwidth Requirement: 5.0 GHz up to 1GBPS transmission rate

Camera: 720p resolution or higher

Traffic Signals: Synchronous traffic signals

11.2 Software Requirements

Programming language/Environment: Python 2.7.x/3.x

Python Dependencies: NumPy, OpenCV, Matplotlib, Darknet, TensorFlow, Keras

11.3 Operational Requirements

Capturing and transmission of High-quality images over the network.

Real-time object detection with high frame rate for object detection.

Efficient algorithm capable of adaptively adjusting signal timer to meet traffic meets.

System to analyse current traffic trends and maintain data over the time.

Traffic signal’s ability to synchronise with other signals at the same junctions.

11.4 Dataset and Connectivity

11.4.1 Dataset used

Microsoft COCO: Common Objects in Context [18]

The dataset is created by gathering images of complex everyday scene containing common objects in their natural state. Objects are labelled using per-instance segmentations to aid in precise object localization. The dataset contains photos of 80 objects class that would be easily recognizable by a 4-year-old. With a total of 2.5 million labelled instances in 328k images.

11.4.2 Implementation platform

Google Colab [19]

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

11.4.3 Framework Used

Darknet [20]

Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.

11.4.5 Object Detector

YOLOv4 [11]

Yolo_v4 is a Convolutional Neural Network based state-of-the-art object detector.

11.5 Pseudo Code

11.5.1 Implementing Object Detection on Google Colab

```
Import required dependencies like matplotlib, opencv, numpy

Import training dataset (COCO)

Import darknet framework

Import suitable convolutional weights

Enable GPU on the colab notebook

Verify the CUDA cores

Create Darknet directory in session storage

Select Images for training and validation along with required weights

Set training parameters for model

Train model

Run model on required Image/ Video
```

11.5.2 Enabling Camera Capture on Google Colab

```
Import required dependencies like display, javascript, b64decode

Import camera-capture snippet from code snippets

Select an input source for camera

While (camera == active)

    Show camera output

    OnPress of Capture button

        Return current frame

        break
```

11.5.3 Designing optimal algorithm for traffic signal timers

```
Import python dependencies like Numpy, Matplotlib

Create a 3D matrix of size (n, 4, 4)

Initialize first instance of matrix

Create four list s1_list, s2_list, s3_list, s4_list ( one for each signal lane in a classic 4-lane junction )

Randomize each list with ‘n’ entries

Define signal_timing function:

        If ( threshold_1 < detected_obejcts < threshold_2):

                Return ( green_signal_timer , 0)

        Elseif ( detected_objects > threshold_2):

                Return ( green_signal_timer, detected_objects – threshold_2)

Append data from random_list to 3d_matrix

For each data in 3d_matrix:

        current_count = data + remaining_vehicle[data-1]

        [Green_signal, remaining_vehicles] = Signal_timing(current_count)

Display 3d_matrix
```

11.5.4 Average waiting time calculations at each signal

```
create two empty list named signal_list and waiting_time_list

For current_count in 3d_matrix:

    For x in range of (current_count):

        Signal_list.append([1,0])

    If (len(signal_list) < = max_permitteed_count):

        For x in signal_list:

            X[1]+=1

            temp_set = [x[1] for x in signal_list]

            temp_set = list(set(temp_set))

            waiting_time_list.append(temp_set)

            signal_list.clear()

        elif (len(signal_list)>(maxi[count,n,2]*2)):

            for x in signal_list:

                x[1]+=1

            temp_set = [x[1] for x in signal_list[:(maxi[count,n,2]*2)]]

            temp_set = list(set(temp_set))

            waiting_time_list.append(temp_set)

            signal_list = signal_list[(maxi[count,n,2]*2):]

        count+=1

waiting_time_list = list(np.concatenate(waiting_time_list).flat)

waiting_time_list = [int(x)*m for x in waiting_time_list]

average = sum(waiting_time_list)/len(waiting_time_list)

return average
```

11.5.5 Average Number of Vehicles at each signal calculations

avg_vehicle_ideal = [sum(s1_list), sum(s2_list), sum(s3_list), sum(s4_list)]

avg_vehicle_dyn = [sum(s1_total_vehicles), sum(s2_total_vehicles), sum(s3_total_vehicles), sum(s4_total_vehicles)]

avg_vehicle_std = [sum(s1_total_vehicles_std), sum(s2_total_vehicles_std), sum(s3_total_vehicles_std), sum(s4_total_vehicles_std)]

avg_vehicle_ideal = [x/n for x in avg_vehicle_ideal]

avg_vehicle_dyn = [x/n for x in avg_vehicle_dyn]

avg_vehicle_std = [x/n for x in avg_vehicle_std]

11.6 Timeline

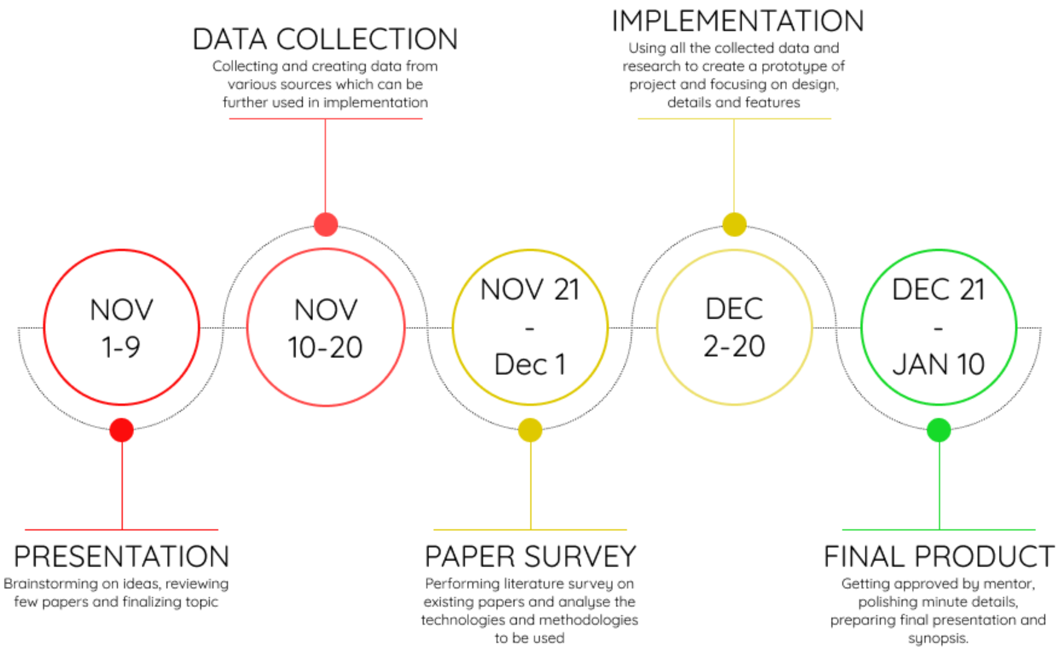

DATA COLLECTION
Collecting and creating data from various sources which can be further used in implementation
IMPLEMENTATION
Using all the collected data and research to create a prototype of project and focusing on design, details and features
NOV 1-9
NOV 10-20
NOV 21 - Dec 1
DEC 2-20
DEC 21 - JAN 10
PRESENTATION
Brainstorming on ideas, reviewing few papers and finalizing topic
PAPER SURVEY
Performing literature survey on existing papers and analyse the technologies and methodologies to be used
FINAL PRODUCT
Getting approved by mentor, polishing minute details, preparing final presentation and synopsis.

Figure 10.6: Tentative Timeline

11.7 Simulation and Working Environment

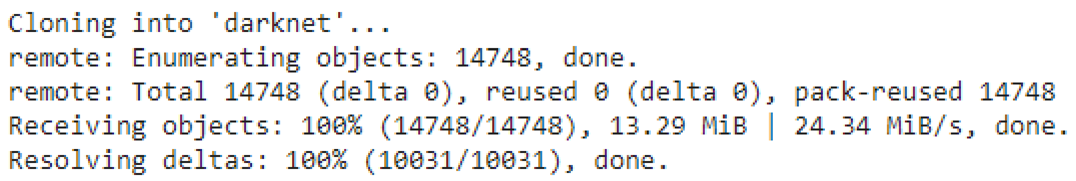
12. Testing

12.1 Verification Testing

12.1.1 Framework Testing

We are using the Darknet framework for our object detection model. By cloning the open-source repository of ‘darknet’ available on GitHub we can import the framework in our system.

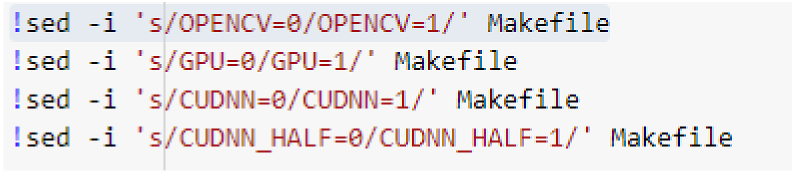
Command: !git clone https://github.com/AlexeyAB/darknet



```
Cloning into 'darknet'...
remote: Enumerating objects: 14748, done.
remote: Total 14748 (delta 0), reused 0 (delta 0), pack-reused 14748
Receiving objects: 100% (14748/14748), 13.29 MiB | 24.34 MiB/s, done.
Resolving deltas: 100% (10031/10031), done.
```

12.1.2 GPU & OpenCV Testing

Having a GPU enabled on Colab notebook is crucial for this system to run. By enabling GPU not only the training time will reduce substantially but also the accuracy of model will increase. On the other hand by default OpenCV is not fully operational on cloud platform hence we have to explicitly create a file to make use of python library to it’s full extent.

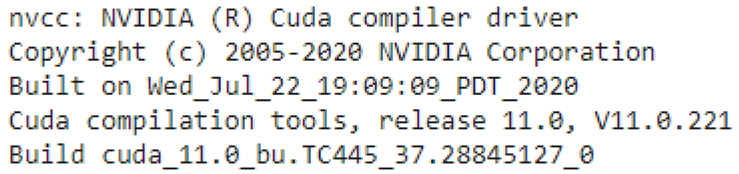


```
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
```

12.1.3 CUDA Verification

‘Compute Unified Device Architecture (CUDA)’ is a way in which you can achieve parallel computing and yield most out of your GPU power in an optimized way, which results in much better performance while executing tasks.

Command: !/usr/local/cuda/bin/nvcc --version

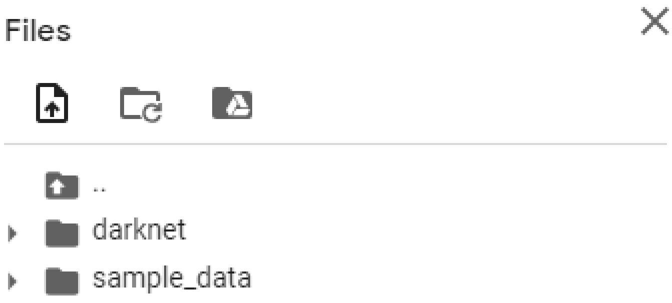


```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Wed_Jul_22_19:09:09_PDT_2020
Cuda compilation tools, release 11.0, V11.0.221
Build cuda_11.0_bu.TC445_37.28845127_0
```

12.1.4 Darknet Verification

After cloning the framework, next thing is to deploy the framework on our system so that we can access its executable files and make required changes as per our need.

Command: !make

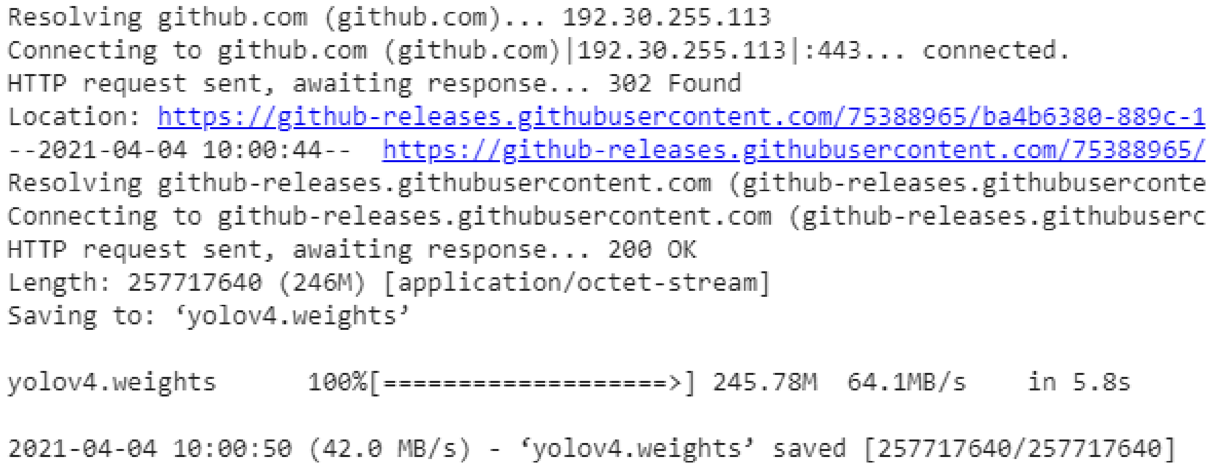


12.1.5 Yolov4 weights verification

Weights are backbone of the system. Using pre-trained weights, we can customize our object-detector as required. Weights decide how much influence the input will have on output.

Command:

!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.weights



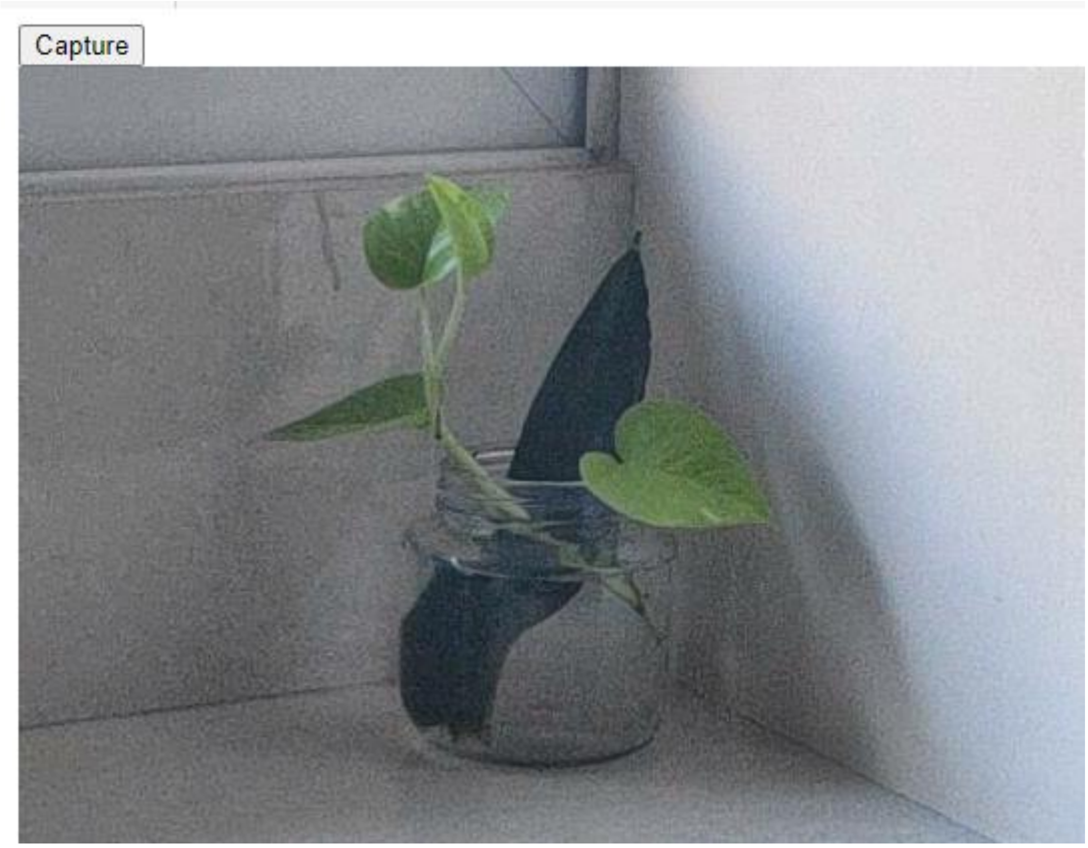
```
Resolving github.com (github.com)... 192.30.255.113
Connecting to github.com (github.com)|192.30.255.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/75388965/ba4b6380-889c-1
--2021-04-04 10:00:44--  https://github-releases.githubusercontent.com/75388965/
Resolving github-releases.githubusercontent.com (github-releases.githubuserconte
Connecting to github-releases.githubusercontent.com (github-releases.githubuserc
HTTP request sent, awaiting response... 200 OK
Length: 257717640 (246M) [application/octet-stream]
Saving to: ‘yolov4.weights’

yolov4.weights      100%[===================>] 245.78M  64.1MB/s    in 5.8s

2021-04-04 10:00:50 (42.0 MB/s) - ‘yolov4.weights’ saved [257717640/257717640]
```

12.1.6 Camera Detection

Camera works as the source of acquiring images for further processing. Here we have used Smartphone camera for demonstration purpose.



12.1.7 Algorithm Testing

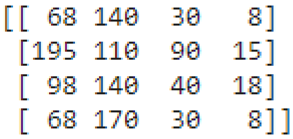
In order to test the proposed algorithm, we have created a classic 4-lane junction scenario. In the following image,

Column1 represents the total number of vehicles present at that signal.

Column 2 represents the Red Signal Time allotted to that signal.

Column 3 represents the Green Signal Time allotted to that signal.

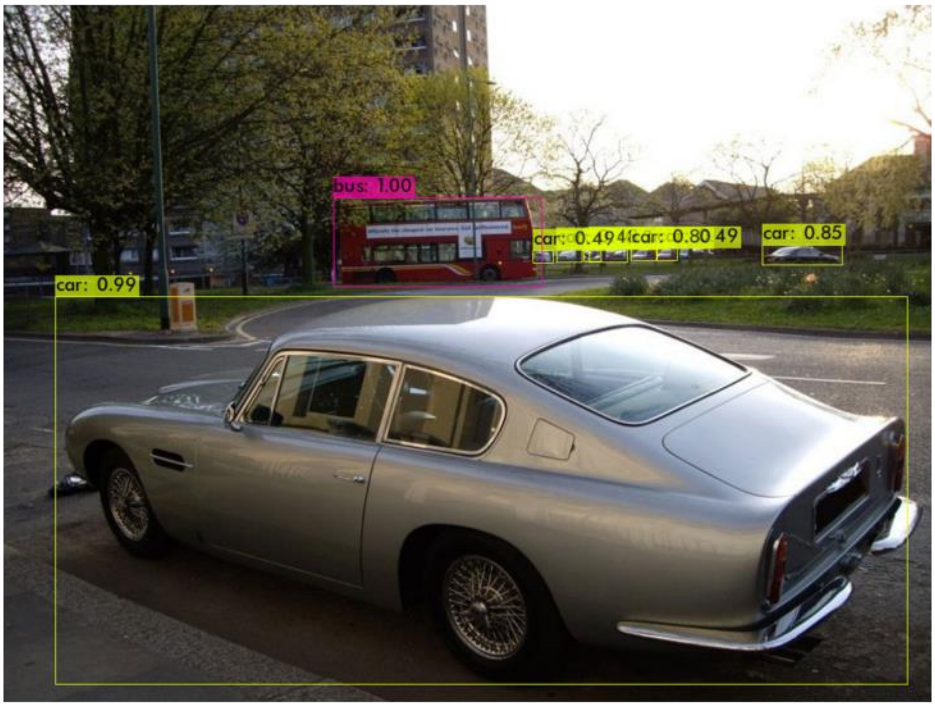
Column 4 represents the remaining number of vehicles after Green signal is over.

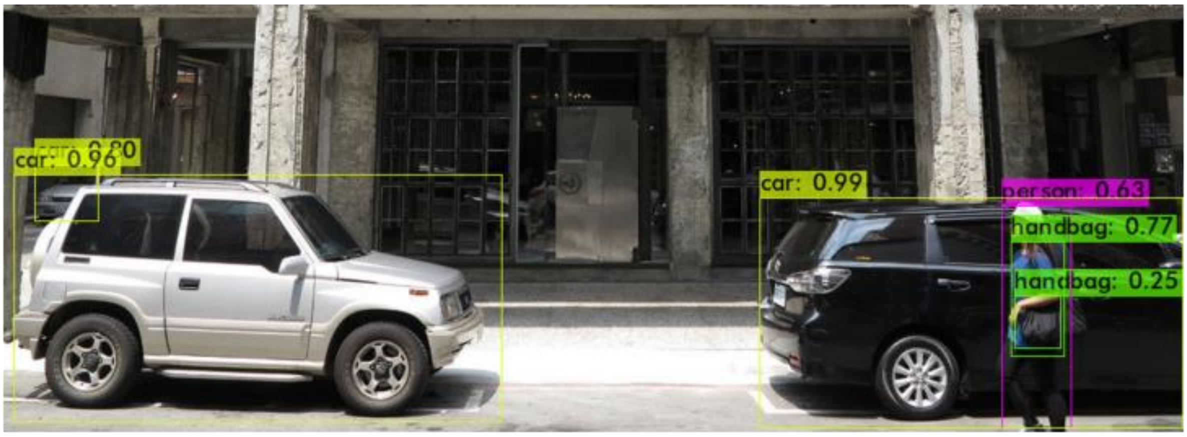


```
[[ 68 140  30   8]
 [195 110  90  15]
 [ 98 140  40  18]
 [ 68 170  30   8]]
```

12.2 Validation Testing

The validation testing is used to evaluate a given model. In object-detectors we use this testing to fine-tune the model hyperparameters. Hence the model occasionally sees this data, but never does it Learn from this. We use the validation test results, and update higher level hyperparameters. So, the validation set affects a model, but only indirectly. The validation test is also known as the Dev test or the Development test. This makes sense since this dataset helps during the development stage of the model.



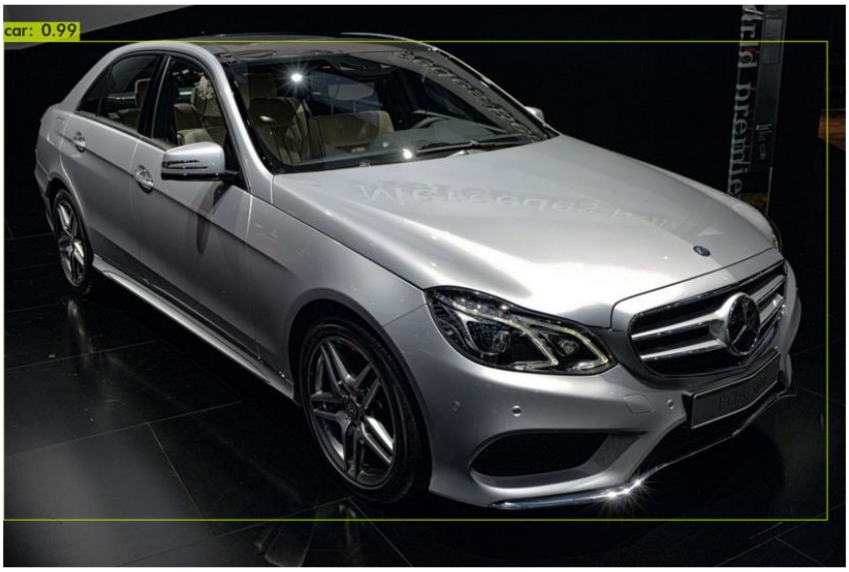


12.3 Test Cases

Test ID	Test Scenario	Expected Outcome	Actual Outcome	Pass / Fail
1	Single Object Detection	Detector should be able to identify object and label it correctly	Detector detected the object and labelled it correctly.	Pass

Test Data

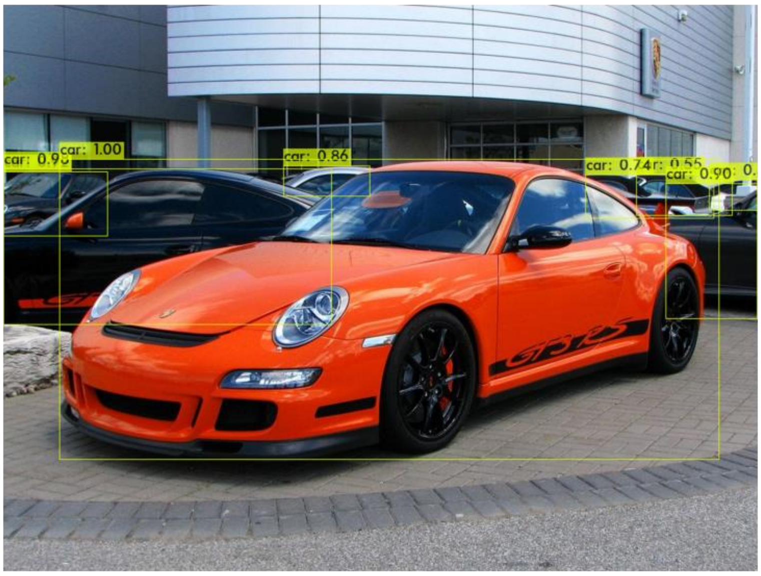
Outcome



Test ID	Test Scenario	Expected Outcome	Actual Outcome	Pass / Fail
2	Single Class Multi Object Detection	Detector should be able to identify multiple object and label it correctly	Detector detected multiple objects and labelled it correctly.	Pass

Test Data

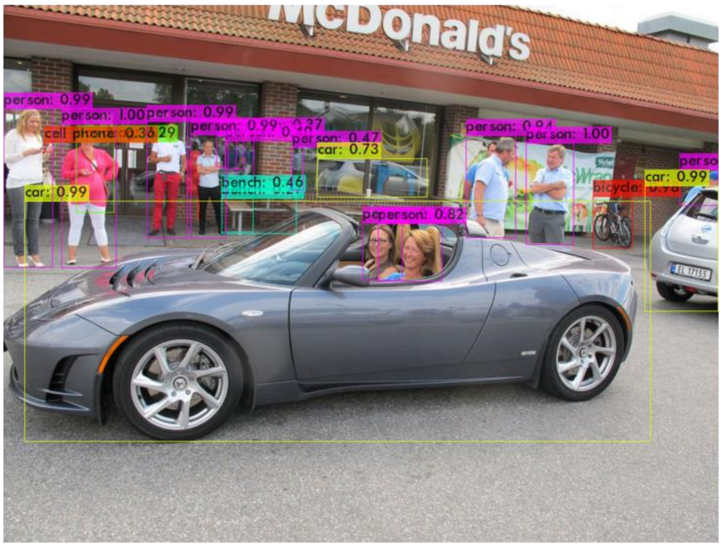
Actual Outcome



Test ID	Test Scenario	Expected Outcome	Actual Outcome	Pass / Fail
3	Multi Class Multi Object Detection	Detector should be able to identify multiple objects of different classes and label it correctly	Detector detected multiple classes objects and labelled it correctly.	Pass

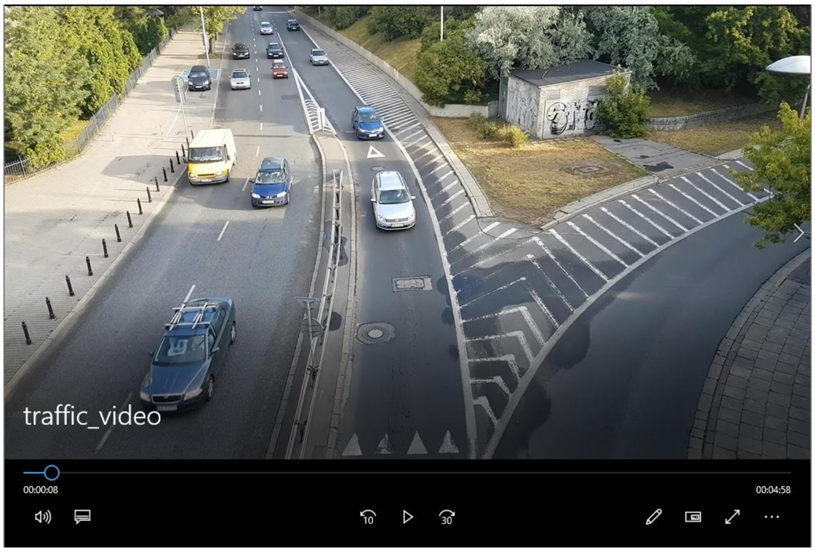
Test Data

Actual Outcome

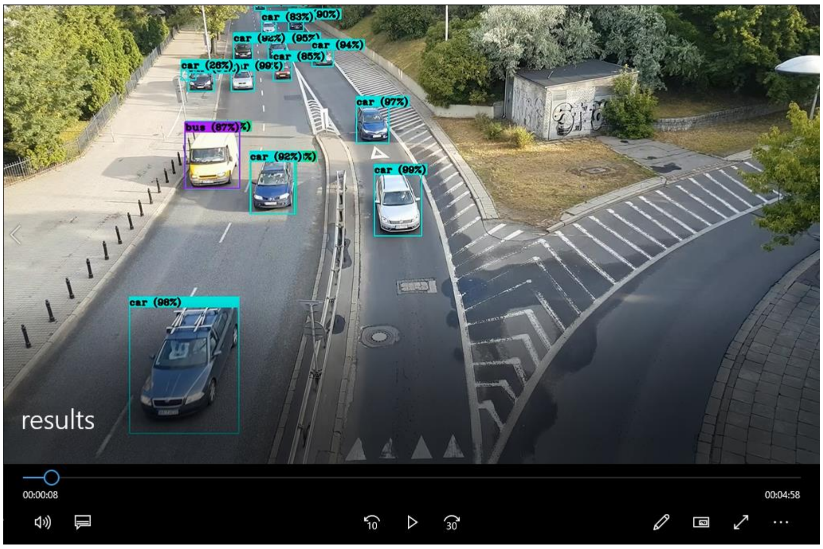


Test ID	Test Scenario	Expected Outcome	Actual Outcome	Pass / Fail
4	Multi Class Multi Object Detection in a real-time video	Detector should be able to identify real-time objects of different classes in a video.	Detector detected real-time objects accurately.	Pass

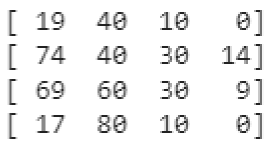
Test Data



Actual Outcome

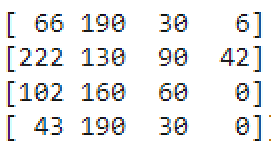


Test ID	Test Scenario	Expected Outcome	Actual Outcome	Pass / Fail
5	Traffic Signal Algorithm Testing over 50 iterations.	Remaining vehicles at the end should be as least as possible.	Vehicles at the end of test were lesser as compared to standard method.	Pass



```
[ 19  40  10   0]
[ 74  40  30  14]
[ 69  60  30   9]
[ 17  80  10   0]
```

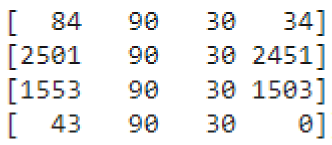
Initial Condition



```
[ 66 190  30   6]
[222 130  90  42]
[102 160  60   0]
[ 43 190  30   0]
```

Final Iteration

Proposed Method



```
[  84   90   30   34]
[2501   90   30 2451]
[1553   90   30 1503]
[  43   90   30    0]
```

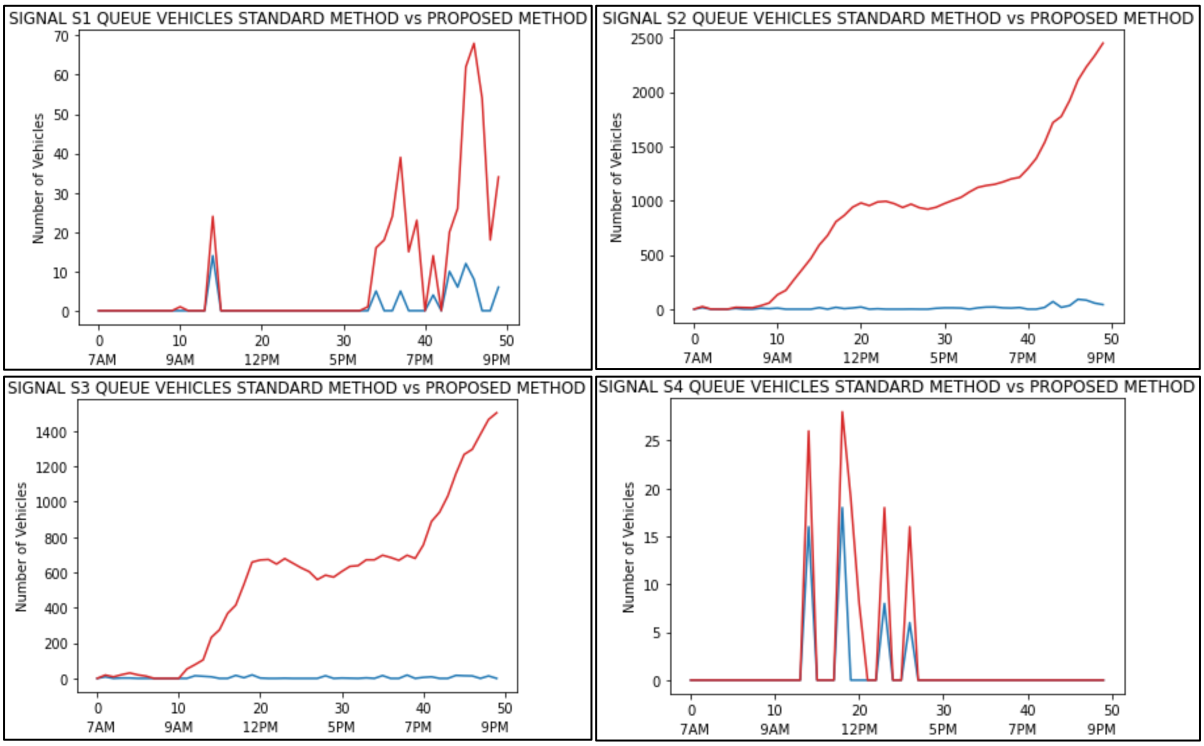
Final Iteration

Standard Method

13. Result Analysis

13.1 Number of Queued vehicles

Number of queued vehicles is a parameter to identify the congestion at each signal by the end of 50 iterations. As seen in the charts below, in the standard method traffic density keeps on increasing after congestion occurs but in the proposed method the density does not differ marginally. Hence, it proves that proposed system can efficiently manage the same amount of traffic without causing long queues.

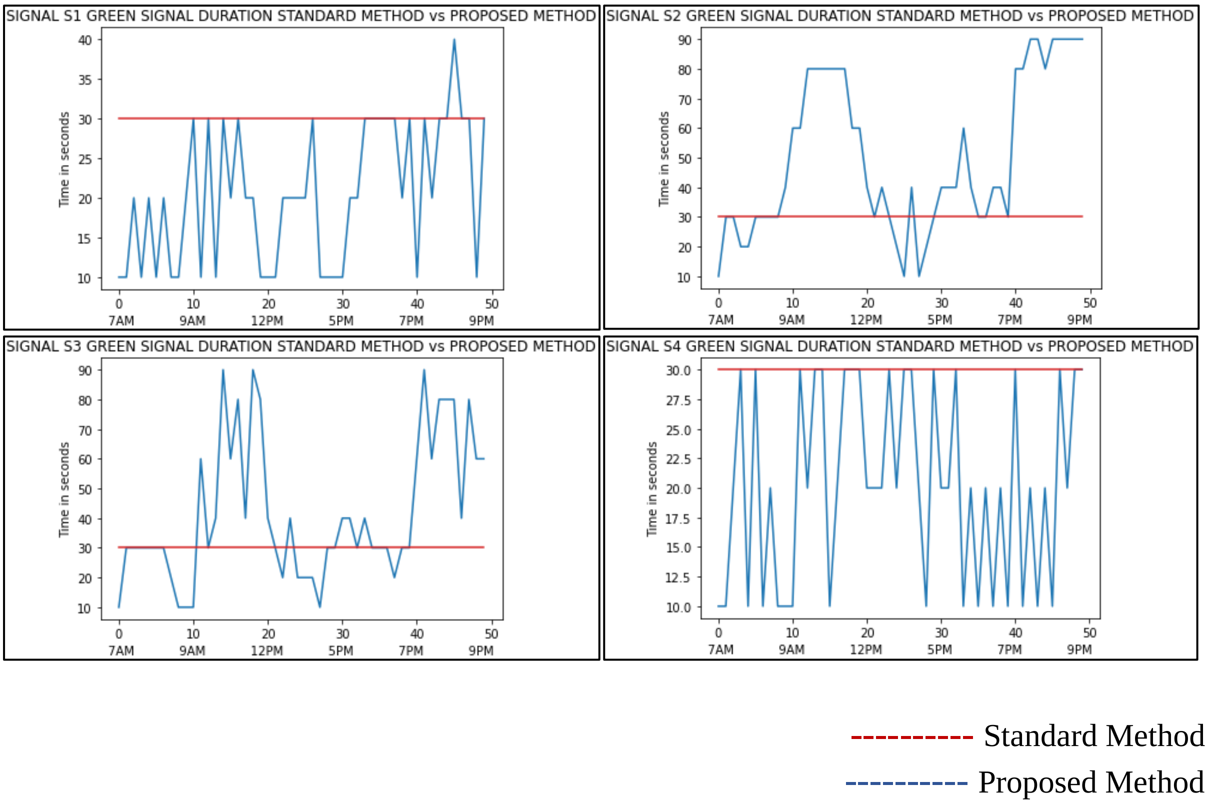


---------- Standard Method

---------- Proposed Method

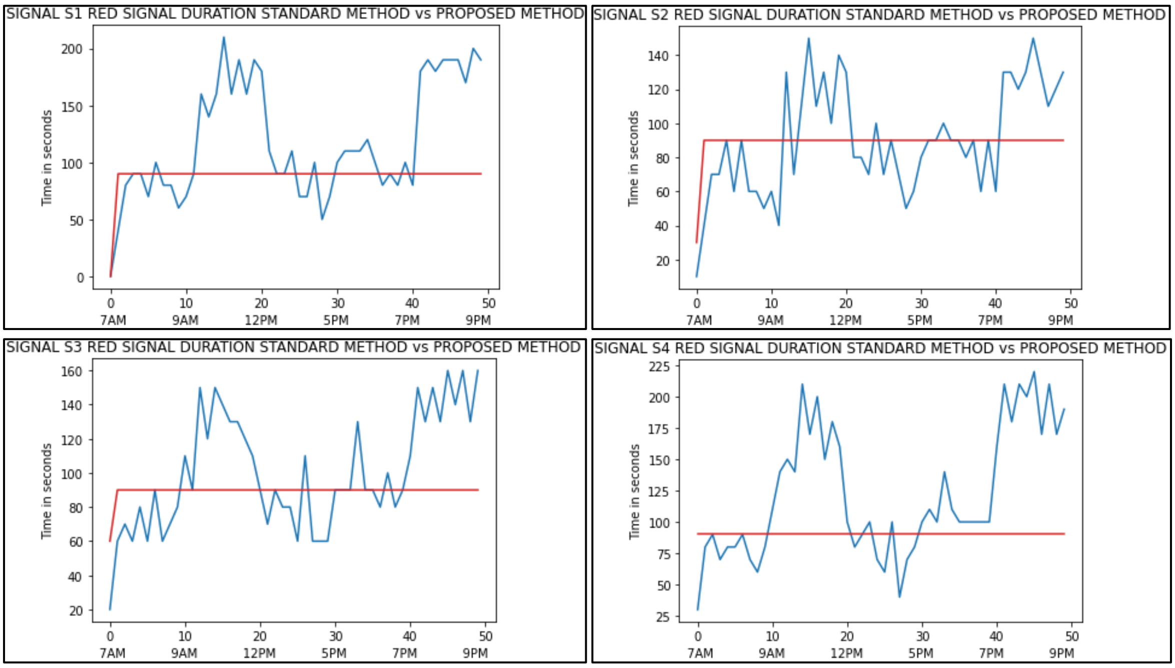
13.2 Green Signal Duration

Green Signal Duration is the data collected during each iteration. As per the proposed system the algorithm will consider vehicle count at each signal and adaptively assign a green signal timer. As shown in the following charts at less congested signals like S1 & S4 the algorithm will allot less opening time. Similarly, at high volume signals like S2 & S3 the opening window increases to meet the present situation.



13.3 Red Signal Duration

Red signal duration is the amount of time vehicles has to stop at each signal for the next turn. As shown in the diagram rather that providing same red signal for each interval our algorithm dynamically changes the window size such that maximum number of vehicles can pass through without causing congestion. In a nutshell, the system borrows green signal time and assign it to high density lanes and similarly takes waiting time from congested lanes and assign it to less volume streets.



SIGNAL S1 RED SIGNAL DURATION STANDARD METHOD vs PROPOSED METHOD

SIGNAL S2 RED SIGNAL DURATION STANDARD METHOD vs PROPOSED METHOD

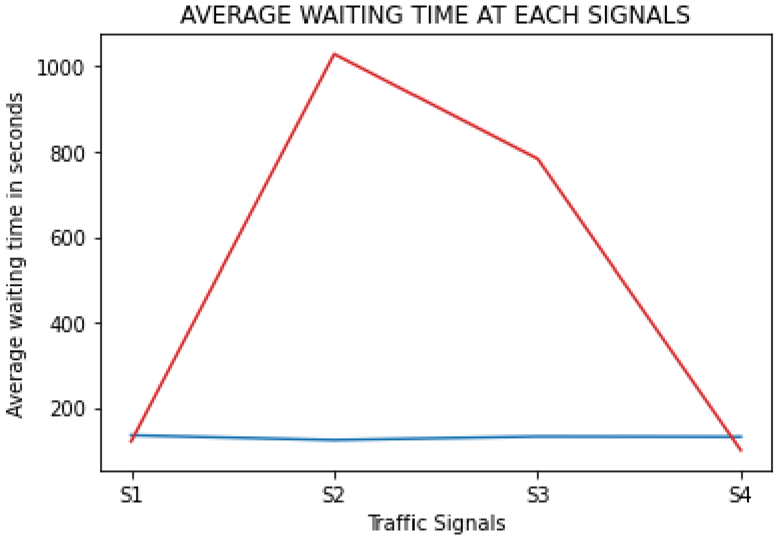
SIGNAL S3 RED SIGNAL DURATION STANDARD METHOD vs PROPOSED METHOD

SIGNAL S4 RED SIGNAL DURATION STANDARD METHOD vs PROPOSED METHOD

---------- Standard Method
---------- Proposed Method

13.4 Average Waiting Time

Average waiting time is the time that each vehicle has to spend at every signal. Average Waiting Time is an important parameter for this system as the ultimate goal is to reduce it as much as possible. In less congested lanes waiting time is not an issue but as the traffic increases the duration people have to wait at each signal also increases. As seen in the chart below for the signals S1 and S4 where traffic density is relatively less the average waiting time is less but as the traffic increases overtime and at its peak when congestion occurs, this waiting time increases substantially. But this is not the case in proposed system. Although, average waiting time at S1 and S4 in proposed system is relatively more, the same at other two signals it is considerably less, which is a trade off we can afford to have.

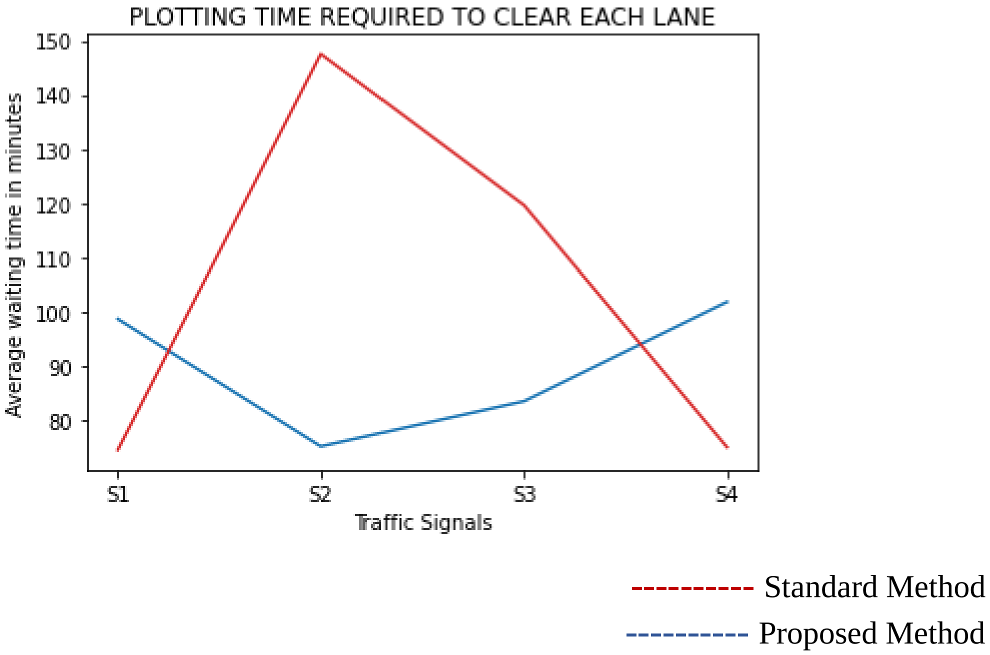


---------- Standard Method

---------- Proposed Method

13.5 Average Clearing Time

Average clearing time is the time required to clear the entire lane. In other words, it is time required for all vehicles present in the system to pass from each signal. This parameter will help us to decide whether the system is actually efficient or not. As seen in the chart below it takes about two and a half hours just to clear the traffic in existing lane for signal S2. However, that is not the case in the proposed system. The time required to clear the densest lane is the least. On the contrary for other two lanes S1 and S4 this time has increased by about 20 minutes. By changing few parameters and equipping the algorithm with more suitable dataset we can adjust the line so that it becomes homogeneous for all four signals.



14. Advantages and Limitations

14.1 Advantages

- The primary advantage of this system is that it will help in reducing traffic congestion during peak hours of the day.
- The system will adapt to the present situation and finds the best solution to avoid long queues at each signal.
- The system serves as a source of vast dataset of each junction which is useful for a centralised storage.
- By analysing the data retrieved from each source the system can help in predicting the peak hour traffic at each junction throughout the system.
- Since vehicle does not have to stand idea for long time it improves the efficiency and reduces the petrol consumption.
- Another benefit of less traffic is reduced pollution both air and sound.
- Commuters can reach the destination quickly and without much hassle.
- This will work as an upgrade from current system and will help in creating more suitable environment for traffic management.

14.2 Limitations

- Since the system is majorly dependent on CCTV cameras, it will become difficult or more expensive to implement in cities having no pre-installed surveillance system.
- Another limitation is lack of available data. Although, there are many datasets available for vehicles in general, none of them are specifically designed keeping Indian scenarios in mind.
- Since this is an object-detecting system, the computational requirements are quite high and might not be suitable for a normal computer system. Hence, making an upgrade in term of processing power could be proves an expensive but valuable investment.
- Lastly, if the cameras can not obtain a clear image then any further processing is not possible at all. In other words, heavy rain can hinder the quality of the captured image and object detection might produce less accurate results.

15. Application and Future Enhancements

16. CONCLUSION

The proposed system will help to manage the traffic more efficiently, which will eventually result in reduced traffic hours and the annual money spent on traffic congestion. As compared to many other possible solutions, our proposal is much more feasible and easier to implement. As the system is dynamic in nature, it is capable of easily adapting to occasional festive traffic without additional modifications. As a results of reduced traffic congestion, average number of vehicles present at each signal is also reducing significantly. Since, vehicles are not waiting in ideal mode, it is also consuming less petrol and as a result pollution is also decreasing. Apart from solving the congestion problem, this is an effective tool for analysing traffic trends in cities. With further modification and expansion, it can scale up to serve as a traffic command center of entire city and hence an effective and much-needed system for developing countries

14. LITERATURE CITED

[1] *J. Raj, H. Bahuleyan, L.D. Vanajakshi*. (2014, December). Application of Data Mining Techniques for Traffic Density Estimation and Prediction. 11th Transportation Planning and Implementation Methodologies for Developing Countries.

[2] S. Dey and M. Rahman, "Application of Image Processing and Data Mining Techniques for Traffic Density Estimation and Prediction," 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 2019, pp. 1-6, doi: 10.1109/ICACCP.2019.8882878.

[3] D. Sitaram, N. Padmanabha, Supriya S and Shibani S, "Still image processing techniques for intelligent traffic monitoring," 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, 2015, pp. 252-255, doi: 10.1109/ICIIP.2015.7414775.

[4] M. S. Uddin, A. K. Das and M. A. Taleb, "Real-time area based traffic density estimation by image processing for traffic signal control system: Bangladesh perspective," 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, 2015, pp. 1-5, doi: 10.1109/ICEEICT.2015.7307377.

[5] M. V., V. V. R. and N. A., "A Deep Learning RCNN Approach for Vehicle Recognition in Traffic Surveillance System," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0157-0160, doi: 10.1109/ICCSP.2019.8698018.

[6] U. E. Prakash, K. T. Vishnupriya, A. Thankappan and A. A. Balakrishnan, "Density Based Traffic Control System Using Image Processing," 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), Ernakulam, 2018, pp. 1-4, doi: 10.1109/ICETIETR.2018.8529111.

[7] S. Dong and Y. Zhang, "Research on Method of Traffic Network Bottleneck Identification based on Max-flow Min-cut Theorem," Proceedings 2011 International

Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), Changchun, 2011, pp. 1905-1908, doi: 10.1109/TMEE.2011.6199586.

[8] S. A. Meshram and R. S. Lande, "Traffic surveillance by using image processing," 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), San Salvador, 2018, pp. 1-3, doi: 10.1109/RICE.2018.8627906.

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

[10] V. Cherniy, S. Bezshapkin, O. Sharovara, I. Vasyliev and O. Verenych, "Modern Approach to the Road Traffic Management in Cities of Ukraine: Case Study of Kyiv Municipal Company "Road Traffic Management Center"," 2020 IEEE European Technology and Engineering Management Summit (E-TEMS), Dortmund, Germany, 2020, pp. 1-6, doi: 10.1109/E-TEMS46250.2020.9111757.

[11] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, " YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv:2004.10934 , 23 April 2020.

[12] https://www.thehindu.com/news/cities/mumbai/city-gets-99-more-private-vehicles-in 2019/article28023565.ece#:~:text=As%20per%20the%20Economic%20Survey,Mumbai%20and%20its%20surrounding%20townships.

[13] https://fullscale.io/blog/computer-vision-concepts/

[14] https://sisu.ut.ee/imageprocessing/book/1

[15] https://www.fritz.ai/object-detection/

[16] https://www.statista.com/statistics/608392/automobile-industry-domestic-sales-trends-india/

[17] https://scroll.in/article/876209/traffic-jams-in-cities-arent-just-frustrating-theyre-also-expensive

[18] https://www.tomtom.com/en_gb/traffic-index/mumbai-traffic/

[19] https://cocodataset.org/#home

[20] https://colab.research.google.com/

[21] https://github.com/AlexeyAB/darknet

[22]