S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

March 2018

Project Title	**Traffic prediction using vehicle inflow data captured by sensors**
Student Name	**M.H.Kulatunge**
Registration No. & Index No.	**2015/MCS/041** **15440411**
Supervisor's Name	**Dr H.E.M.H.B.Ekanayake**

For Office Use ONLY

Traffic prediction using vehicle inflow data captured by sensors

A dissertation submitted for the Degree of Master of Computer Science

M.H.Kulatunge

University of Colombo School of Computing

2018

UCSC

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: **M.H.Kulatunge**

Registration Number: **2015/MCS/041**

Index Number: **15440411**

Signature: Date:

This is to certify that this thesis is based on the work of Ms. M.H.Kulatunge under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: **Dr.H.E.M.H.B. Ekanayake**

Signature: Date:

Abstract

As the problem of urban traffic congestion spreads, there arises the need for the use of advanced technology to provide information about traffic congestion. Although many types of traffic sensors are currently in use, all have some drawbacks, and the deployment of such sensor systems has been difficult due to high costs. Sensors such as Inductive loop detections, Infrared sensors, Magnetic sensors are hard to deploy and inaccurate when comes to detecting traffic. Ultra sonic sensors and video cameras are used in this research due to the reasons that ultra-sonic sensors are cost effective, easy to deploy and video cameras can be easily installed and supports scalability. Within the scope of this research, three major contributions are presented. First is an approach to detect traffic by capturing vehicle speed using ultra sonic sensors. The second contribution is using video processing to detect traffic of urban routes in Colombo by capturing vehicle speed and count. As the third contribution, a machine learning approach was used to detect traffic in urban routes of Colombo. Moreover, this research has used a statistical based approach to predict traffic using traffic data captured by the machine learning approach.

Experiments to verify the suitability of ultra sonic sensors to capture vehicle speed was carried out in a prototype environment. Whereas the evaluation of the video processing to capture vehicle speed and count was carried out in a real world environment. For the machine learning approach a model was trained with over 4000 images and evaluated the accuracy of traffic detection with video feed taken from urban areas of Colombo.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor Dr. H.E.M.H.B. Ekanayake of UCSC for the continuous support of my research, for his patience, motivation, and immense knowledge. Dr. Ekanayake was always available whenever had doubts, questions about my research or writing. He consistently allowed this paper to be my own work, and steered me in the right the direction whenever he thought I needed it.

I would also like to thank my friends, for supporting and motivating me when I needed it the most. Finally yet importantly, I would like to thank my parents for encouraging me and for their wise counsel. This thesis would not have been possible without your love and kindness.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IDE – Integrated Development Environment

CCTV – Closed Circuit Television

CNN – Convolutional Neural Network

ResNet – Residual Network

LSTM – Long Short Term Memory

RNN – Recurrent Neural Network

GPU – Graphical Processing unit

CUDA- Compute Unified Device Architecture

1 INTRODUCTION

Due to advancements in electronics, sensor nodes become smaller and cheaper. Modern communication technology made the sensor node to communicate in a better way by developing several efficient protocols and algorithms.

Sensor networks can be used effectively for home automation, industrial control, military and health applications. Nevertheless these sensor technologies has become popular in transportation context as well mainly with the intension of traffic detection, vehicle parking, traffic light operations. People try out various new solutions with the intension of solving traffic problems.

In the last decades, we witnessed a large increase in traffic and transport demand that has created and aggravated capacity problems in the infrastructure causing traffic congestions and delays. Heavy traffic conditions not only affect the welfare of citizens from the economic point of view but they are also related to their health status speaking both psychologically, due to stress accumulated during their travels, and physically, due to high air pollution levels.

This research has focused on the two major areas namely, **Traffic data collection using sensor technologies** and **traffic prediction.** After a thorough study of sensor technologies Ultra sonic sensors and video Image processing was chosen to calculate traffic data such as vehicle speed and count.

There has been several studies conducted to see how sensors can be used in relation to traffic. Most of these attempts involve trying to detect vehicle presence, classification, speed, count. There are several existing intelligent traffic prediction solutions but are more generalized to transport contexts globally. In Sri Lanka, we have a completely different traffic context that is quite different from traffic scenarios in other countries. Less disciplinary driving with vehicles not navigating on the correct lanes, pedestrians crossing from all over the place are some of the common behavior we observe. Hence, with such mixed traffic context in Sri Lanka it is quite tricky to predict traffic congestion. There are hardly any traffic prediction solutions that cater traffic conditions considering the transportation context of Sri Lanka. Having such a traffic prediction solution can have tremendous impact on one’s personal life, career and future. Hence, the attempt here is to provide a traffic prediction solution to support Sri Lankan traffic context.

This research is an attempt in solving the above-mentioned problem by trying to find answers to the following research questions.

1. How can sensors be used to detect vehicle speed and count in a flow of traffic for an urban route in Colombo?
2. How can the flow of traffic be determined using traffic data collected by sensors?
3. How can the current traffic flow in an urban route of Colombo be used to predict the future traffic?

1.1 Objectives

The objectives listed below are considered with respect to the context of urban route in Colombo.

- To identify a method to capture the speed of moving vehicles using selected sensors.
- To identify a method to capture the vehicle count using selected sensors.
- To determine the current traffic flow of a selected route.
- To predict the future traffic by using information on the current traffic flow.

1.2 Scope

The scope of the project can be explained in three sections,

1. Traffic route properties
2. Collecting traffic parameters
3. Traffic prediction

1.2.1 Traffic route properties

Road structures and properties differ from one another across different cities. This research will only focus on predicating traffic for routes that fulfills the following characteristics.

- A traffic route that has a uni-directional traffic flow.
- A route that has a maximum number of 3 lanes.
- A road section that has uniform geometric characteristics along the road length.
- A road where vehicle inflow from bi-roads adjoins to a main road.

1.2.2 Collecting of traffic data

This research focuses on obtaining vehicle speed and vehicle count as traffic parameters using one or more selected sensor technologies.

1.2.3 Traffic Prediction

- Traffic prediction is done to accommodate the time period of 6am to 6pm in predicting the traffic.
- A textual categorization of the traffic status is provided as the prediction.

1.2.4 Delimitations

1. Extreme weather conditions are not considered.
2. Traffic route with interactions, roundabouts, overhead bridges are not considered.
3. Several factors naturally affect the traffic conditions in a normal route. This study will not consider the below mentioned scenarios.
    - **Road clearance due to emergency vehicles**
    - **Road construction**
    - **Road Accidents**
    - **Road closure**
4. The study will not address external factors in the environment that affects wireless sensors.
    - **Weather conditions**
    - **Illumination conditions**

1.2.5 Assumptions

- Government rules for having a separate lane for buses will continue to be valid until the research is completed.
- The choice of route will not subject to changes in road properties such as extended road lanes, overhead bridges.

1.3 Structure of this document

The upcoming chapters of this document discusses about the Background of the research context where existing work in this area is critically analyzed. Next the Analysis and Design of this research with the approaches and methodology used are discussed in details. Followed by the Implementation chapter, finally the Evaluation chapter describes the experiments carried out and their results. Finally, the Conclusion and Future work are discussed.

2 BACKGROUND

In this research there are number of areas that considered for obtaining a feasible and most suitable solution. As the topic revolves around detecting and predicting traffic using Senor Technologies, first and foremost the available sensor technologies in the context of vehicle detection are discussed in this section. In addition, a comparison of existing work on sensor based traffic detection to identify the challenges and limitations are listed. Finally sensor, traffic simulation software, technologies are discussed.

2.1 Overview of sensors technologies in traffic detection.

Traffic detection technologies contains three components, the transducer, a signal processing device, and a data processing device. The transducer detects the passage or presence of a vehicle or its axles. The signal-processing device typically converts the transducer output into an electrical signal. The data-processing device usually consists of computer hardware and firmware that converts the electrical signal into traffic parameters. Typical traffic parameters include vehicle presence, count, speed, class, gap, headway, occupancy, weight and link travel time. The data processing device may be a part of the sensor, as with devices that produce serial output data, or may be controllers external to the sensor as utilized with sensors that have optically-isolated semiconductor or relay outputs.

The sensors can be classified into intrusive and non intrusive based on the way they are mounted. Intrusive sensors are installed under or across the pavement. They provide accurate traffic information, but the installation and maintenance of these sensors can cause traffic disruption. Some of the Intrusive sensors are inductive loop detectors, pneumatic tubes, piezoelectric sensors. On the other hand, Non-intrusive sensors are installed above or to the side of the road, ensuring minimal disruption to traffic flow. They include microwave radar, infrared, video, ultrasonic systems and acoustic sensors. [1]

2.2 Comparison of different types of sensors

The following Table 2.1 based on information in articles [1] [2] [3] discusses the different types of sensors that can be used for vehicle detection, their advantages and disadvantages.

Table 2.1 - Different sensor types and their usages

Device	Description	Advantages	Disadvantages
Video Image Processors	A combination of hardware and software with extracts information with the use of a camera, infrared camera	Can detect presence over a number of lanes/zone. Vehicle detection zones can be configured easily. Rich array of data available. Provides wide-area detection when information gathered at one camera location can be linked to another.	Need to overcome shadows, weather, and reflections from the roadway surface. Performance affected by inclement weather such as fog, rain. Reliable night-time signal actuation requires street lighting.
Infrared Detectors • Active • Passive	***Active IR*** "Operate by transmitting energy from either a light emitting diode (LED) or a laser diode" [2] ***Passive IR*** "Detects energy emitted by objects in the field of view" [2]	***Active IR*** Detection possible in both day & night. Transmits multiple beams for accurate measurement of vehicle position, speed, and class. Multiple lane detection available. ***Passive IR*** Multizone passive sensors measure speed.	***Active IR*** • Sensitive to weather conditions and ambient light • Installation and maintenance, including periodic lens cleaning, require lane closure ***Passive IR*** Vehicle sensitivity in weather conditions such as heavy rain, snow and dense fog. Some models not recommended for presence detection.

Ultrasonic detectors	"Operate by transmitting ultrasonic energy and measuring the energy reflected by the target." There are two types of ultrasonic sensors available, one for detecting the vehicle presence-only and the other for speed measuring.	Can be easily mounted on the roadside. Multiple lane operation available Capable of over height vehicle detection.	Should be mounted facing perpendicular as possible for better detection. Environmental conditions such as temperature change and extreme air turbulence can affect performance. Large pulse repetition periods may degrade occupancy measurement on freeways with vehicles travelling at moderate to high speeds.
Microwave/ Millimeter wave radar	"They operate by measuring the energy reflected from target vehicles within the field of view" [2]	Uses mature technology Can detect over multiple lanes Insensitive to inclement weather.	Possibility of false detections. Continuous wave (CW) Doppler sensors cannot detect stopped vehicle
Piezoelectric	Consists of a long strip of piezoelectric material enclosed in a protective casing. Vehicles need to pass on top of it.	Accurate vehicle detection on when and where a vehicle passed.	Doesn't detect stationary vehicles Permanent installation on pavements. Problem arises in road repairs.
Inductive loop detectors	one or more loops of wire embedded in the pavement and connected to a control box	Reliable detection for parameters like count, occupancy compared to other sensors. Insensitive to weather conditions such as rain, fog, and snow.	Installation overhead. Eg: Need to reinstall if a pavement is replaced. Installation and maintenance require lane closure. Wire loops subject to stresses of traffic and temperature.
Magnetic Detectors • Active	Operative with large metal object	***Active***	Need to install multiple detectors to detect small vehicles

• Passive	disturbing a magnetic field. ***Active*** (Magnetometer) "It measures the change in the magnetic field caused by the passage of a vehicle." ***Passive*** "Measures a change in the flux of the earth's magnetic field caused by the passage of a vehicle"	Less susceptible than loops to stresses of traffic. Insensitive to inclement weather such as snow, rain, and fog. Some models transmit data over wireless radio frequency (RF) link ***Passive*** Detection over multiple lanes	***Active*** Installation requires pavement cut. Improper installation decreases pavement life. Installation and maintenance require lane closure. ***Passive*** Can't detect the idle vehicles.

2.2.1 Type of parameters that can be obtained for each sensor type

The Table 2.2 shows the different types of traffic parameters that can be gathered using each sensor type.

Table 2.2 - Parameters that can be obtained using different types of sensors

Sensor	Count	Speed	Classification	Presence
Video Image Processors	Yes	Yes	Yes	Yes
Infrared Detectors - Active	Yes	Yes +	Yes	Yes
Infrared Detectors - Passive	Yes	Yes +		Yes
Ultrasonic detectors	Yes	Yes		Yes
Microwave/Millimeter wave radar	Yes	Yes	Yes&	Yes &
Inductive loop detectors	Yes	Yes*	Yes@	Yes
Magnetic Detectors	Yes	Yes*		Yes

+ : Possible only with multi detection region

& : Need to have a suitable processing unit

* : Two sensors are required

@ : Require specific electronic devices

2.2.2 Using a combination of sensors to capture traffic parameters

Using two or more different types of sensors can yield more accuracy. Also a combination of sensors is necessary when one type of a sensor cannot provide all the traffic parameters that are required. One such example for using a combination would be to combine passive infrared sensor with ultrasound or Doppler radar. The passive infrared-ultrasonic combination provides enhanced accuracy for presence and queue detection, vehicle counting, and height and distance discrimination. Also another example is the passive infrared-Doppler radar sensor. It is designed for presence and queue detection, vehicle counting, speed measurement, and length classification.

2.2.3 Cost of Sensors

It is commonly known in the sensor world that types such as Infrared, ultrasonic, microwave are inexpensive. However, this could vary depending on the application that is being built. The reason for this would be the number of sensors required the cost for mounting the device in the desired location. For instance, Video image processer would be wiser choice when it comes to detecting the speed of a vehicle that would require a couple of IR sensors to acquire the same data. Hence, the cost is always a choice between the type of sensor and the number of sensors required.

2.3 Sensors selected for the research

As the choice of sensor for this research **Ultra Sonic** sensors was chosen. The Intrusive sensors such as inductive loop detectors, pneumatic tubes, piezoelectric, magnetic detectors will not suit for implementation as it requires mounting the sensors under the road or pavement, which not suitable incase of a road maintenance or construction. And it is common in Sri Lankan roads for frequent road maintenance hence the above mentioned intrusive sensors are not suitable. Moreover, the installation of these sensors will cost much more than the outcome of work. Among the non intrusive sensors, the IR sensors alone cannot accommodate traffic parameter detection, it will require several of IR sensor or a IR sensor with combination of another sensor. This will be more costly and also adding more complexity. Hence Ultra Sonic sensors were selected.

In this research, for the purpose of capturing traffic parameter a second sensor type **Video processing** was also chosen. Considering the different types of parameters that can be obtained from sensors in Table 2.2 their disadvantages as mentioned in Table 2.1 and considering the cost perspective, as the second choice video image processing is chosen to capture the needed traffic parameters. As video processing allows clear and distinct object detection, apart from obtaining only the traffic parameters it is possible to detect how further a particular lane has queued up to and also support human activity detection.

2.4 Existing software and tools used for Ultrasonic sensors and video processing

This section discusses various software and tools that are currently available to configure and implement solutions using Ultra sonic sensors and Video processing.

2.4.1 Software and tools for video processing

There are a number of software's that facilitates video processing. Among them MATLAB® and Simulink® provides functions to "acquire images and video from imaging hardware, use graphical tools to visualize and manipulate images and video, develop new ideas using libraries of reference-standard algorithms , migrate designs to embedded hardware" However this is a commercialized software and requires knowledge on C/C++ to work with. There are disadvantages in terms of cost, portability and speed/performance.

OpenCV (Open Source Computer Vision Library) is another software that can be used to video/image processing. It's available for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV is good in terms of computational efficiency and is very much suitable for real-time applications.

2.4.2 Software and tools ultra sonic sensors

The Arduino project a commonly used software provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. This open souse software is easy to use and provides flexibility in configuring electronic sensors. Hence, this software is ideal for the purpose of configuring and using the ultra sonic sensors

2.5 Related Work

Here Related work of traffic detection using ultra sonic sensors and video processing technologies are discussed separately.

2.5.1 Work related to ultra Sonic sensors

There are several studies that has attempted vehicle detection using ultra sonic sensors, which of some involving ultra sonic sensors mounted on the road side, and some mounted into vehicles itself. However, there is hardly any study found attempting to detect vehicle speed solely based on ultrasonic sensors.

In [4] their attempt is using a combination of infrared sensors and ultra sonic sensors to identify the vehicle classification, speed using a statistical approach of Bayesian Networks. The combination of sensors has yielded positive results in vehicle detection, and 5kmph error for speed detection. Even though this study has showed valuable results, implementation of this will be costly as it involves a combination of sensors and proper configuration.

Research done in [5] uses ultrasonic sensors to capture traffic related information. The systems described is capable of measuring traffic on multiple lanes, identifies vehicle classification, speed. The paper [5] discusses their own device called a wireless ultrasonic sensor mote (WUSM), which is said to be small and easy to install and can be adapted for use in a variety of real road environments.

Most ultrasonic sensors detect vehicles by measuring from top to bottom or from side to side diagonally. However, these approaches require that a detector be installed for each lane because each detector measures only one lane on a road. Furthermore, ultrasonic sensors require considerable infrastructure on a road. [6]

Also the approach in [6] seems problematic with an area with high noise because the ultrasonic sensor receiving noise data that can lead to errors in the vehicle detection procedure. Especially when the noise is not only generated by multiple vehicles passing by, but also due to environmental facts.

Another study using ultra sonic sensor and traffic is in article [7] , detects traffic violations in a highway using two ultrasonic sensors. Even though the work in [7] is not directly related to detecting traffic flow, the approach used to detect traffic violation behaviors is a good input for traffic development detection. However limitation in [7] is that there are cases where the vehicle count hasn't been quite accurate.

For [8] the system uses ultrasonic sensors installed on the side of a road which measures the initial traffic information by a lightweight vehicle detection algorithm. One of the focused areas in the study [8] is to minimize the power consumption, which they have been able to achieve by the designed three-phase routing protocol. Transferring of traffic data has been tried out in method of adaptive segmentation which has not been quite successful when transmitting data in shorter duration of time typically less than 30 sec intervals which has led to missing detection of vehicles. However, vehicle data has been much accurate when the data transmission interval is constant.

2.5.2 Work related to video processing

Applying image processing technologies to vehicle detection has been an area of interesting the recent research filed related to Intelligent Transportation Systems (ITS).

Much earlier study done in [9] resulted in the autoscope video detection systems that are widely used in today's traffic detections and surveillance around the world. However modern studies has been carried out throughout using different approaches in terms of algorithms, vehicle classification via computer vision have occurred.

An attempt to detect vehicles using video feed done in [10] is developing a real-time vehicle detection system for low-resolution traffic video feed using existing algorithm by Harris-Stephens. The developed system determines the total and lane based vehicle counts and average speed of the vehicle for a given segment of the roadway. Also includes a vehicle detection application that has an advanced warning of congestion and queues at work zones and on freeways during special events. This warning sign indicated the traffic situation considering the vehicle speed, count gathered as real time data using Radio frequency. Even though the attempt in [10] has been successful up to an extent there are deviations when a single camera is used to cover a number of lanes (3-4) the placement of the video equipment has an impact on the obtained video feed which claims to be skewed.

The study carried out in [11] has suggested their own image processing algorithms for traffic counting, queue length, speed measurement and vehicle classification. The algorithm adopted in [11] is based on changes in pixels values of the video in the middle of traffic lanes. The pattern of these pixels values are used to measure the queue length, length of individual vehicle and to detect the position of a particular vehicle within a short interval of time.

Both [10]and [11] has considered obtaining the queue length of a road to predict accurate traffic conditions, which is advantage compared to other traffic detection systems that uses sensor technologies such as inductive loops.

Study carried out by [12] has demonstrated that accurate vehicle dimension estimation could be performed through the use of a set of coordinate mapping functions. Although the solution in [12] estimated vehicle lengths to within 10% in every instance, the method has required camera calibration in order to map image angles and pixels into real-world dimensions.

The proposed solution in [13] is to develop a unique algorithm for vehicle data recognition and tracking using Gaussian mixture model and blob detection methods. The approach is to differentiate the foreground and background frames and also eliminating the disturbances occurred through noise. The foreground detector detects the object and a binary computation is done to define rectangular regions around every detected object. Then the final counting is done by tracking the detected objects and their regions which has produced significant results.

This however does not focus on the traffic prediction aspect, but rather on obtaining vehicle related data using image processing. This will be a useful input if the vehicle classification information is considered when developing a traffic prediction system using video processing.

The system developed in [14] presents a density analyzer scheme based on counting the number of vehicles in the present image. The density detection algorithm searches for connecting pixels, which is then identified as a vehicle based on their own defined threshold.

A fuzzy based controller and morphological edge detection techniques are proposed in [15]. Similar to [14] this is also based on measuring traffic density, but in here it has been done by correlating the live traffic image with a reference image.

Although several commercial video image processing systems have been developed for traffic data collection, these systems are typically subject to several major problems including complicated calibration processes, poor detection accuracy under certain weather and lighting conditions, etc. Nonetheless, these previous investigations provide valuable insights to the video-based vehicle detection and classification problems to be addressed in this study.

3 ANALYSIS AND DESIGN

The research focuses on the two main areas, **collecting traffic inflow data and traffic prediction using the current vehicle inflow.** In this chapter the analysis and design for the above two areas are discussed.

3.1 Collecting traffic inflow data

In this research two-sensor technologies Ultra sonic sensors and video Image processing was chosen to see how accurately can traffic parameters such as vehicle speed and vehicle count can be obtained in traffic contexts of urban areas in Colombo. This section will first describe how ultra sonic sensors was used to capture traffic data. Secondly the usage of image processing for the same task is described.

3.1.1 Approach used to collect traffic parameters using ultra sonic sensors

3.1.1.1 *Characteristics of ultra sonic waves*

Ultrasound is an acoustic wave with a very high frequency, which is beyond human hearing. The audible frequency of humans are said to be between 20Hz and 20kHz, the ultrasound means acoustic waves above 20kHz.

Ultrasound has several characteristics which make it so useful and that have led to its use in many electronics applications. Being inaudible to humans is one advantage since it will not be annoying to humans. Ultrasound waves are a compressional vibration of matter. Also they have a lower propagation speed than light or radio waves.

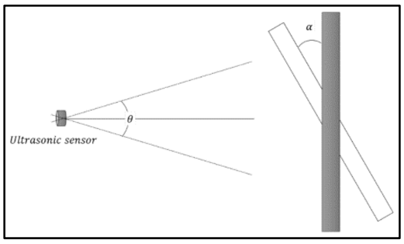


Figure 3.1 - Angle of the Ultra sonic sensors

The beam angle of an ultrasonic sensor depends on the ultrasonic frequency. As illustrated in Figure 3.1 for instance, the beam angle θ increases as the ultrasonic frequency decreases.

In addition, to detect the reflected ultrasonic wave, we require α<θ2, where α is the angle between an ultrasonic sensor and the detection object.

3.1.1.2 *Features of the ultra Sonic sensor used*

The HC-SR04 ultrasonic sensor offers non-contact range detection with high accuracy and stable readings from 2cm to 400 cm. This sensor is chosen for the research as it suitable in size and scale of the prototype model.

The specification of the sensor is as below,

Power Supply :+5V DC

Quiescent Current : <2mA

Working Current: 15mA

Effectual Angle: <15°

- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

The Figure 3.2 below shows how an ultra sonic sensor detects an object.

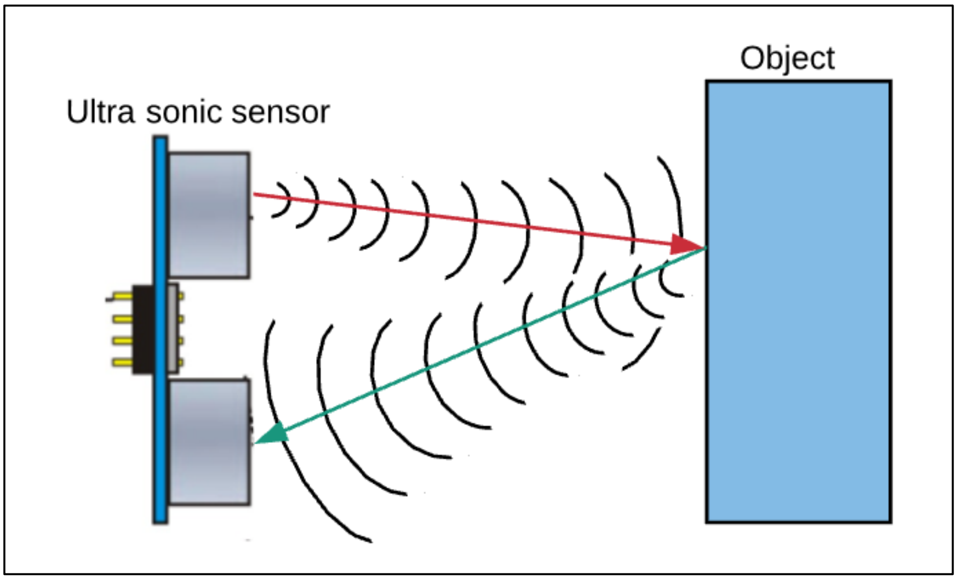


Figure 3.2 - Object detection using ultra sonic sensor

As shown in Figure 3.2, from one opening of the ultra sonic sensor a high frequency sound pulse is sent, the other opening receives ultra sonic waves, which was reflected by hitting an object.

3.1.1.3 *Placement of the ultra sonic sensor*

There are several ways a sensor could be mounted. These ways are illustrated in Figure 3.3.

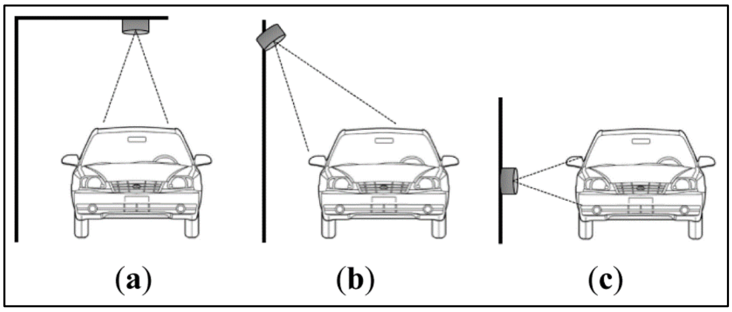


Figure 3.3 - Ways and ultra sonic sensor can be mounted

Where (a) ,(b) ,(c) in Figure 3.3 refers to,

(a)The overhead mount.

(b)The angled mount.

(c)The horizontal mount.

In this research the horizontal mount is chosen to place the ultra sonic sensors. As this mounting position is the most suitable to detect vehicles traveling in a single lane road. In addition, this mounting position required less effort for installation.

3.1.1.4 *Detecting the speed of a vehicle using Ultra sonic sensors*

This research attempts to capture the vehicle speed using two ultra sonic sensors of a single lane road. The diagram in Figure 3.4 illustrates a model of how a the sensors are placed for the purpose of capturing vehicle speed.

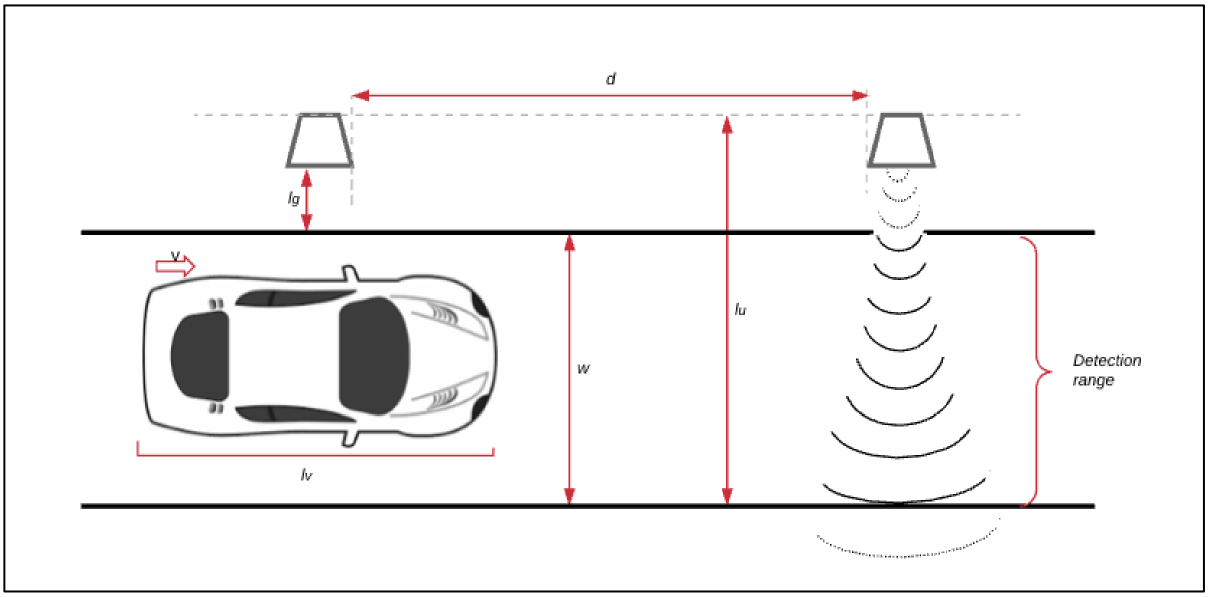


Figure 3.4 - Placing ultra sonic sensors on the road

As shown in Figure 3.4 we consider a vehicle with length l_v traveling with a constant speed of v along the road which is of a width of w. Both ultrasonic sensors are placed at a distance of l_g from the roadside. The distance between the ultra sonic sensors is d. Let the ultrasonic wave spreads at constant speed v_u and with angle θ. And the width of the ultrasonic wave l_d. Even though the wave detection range of the ultra sonic sensors spreads outs beyond the road length, it is configured to only detect for the range of the road length to avoid unnecessary detection long the pavement area. The detection range will be l_u- l_g

The model shown in Figure 3.5 illustrates the detection of a vehicle.

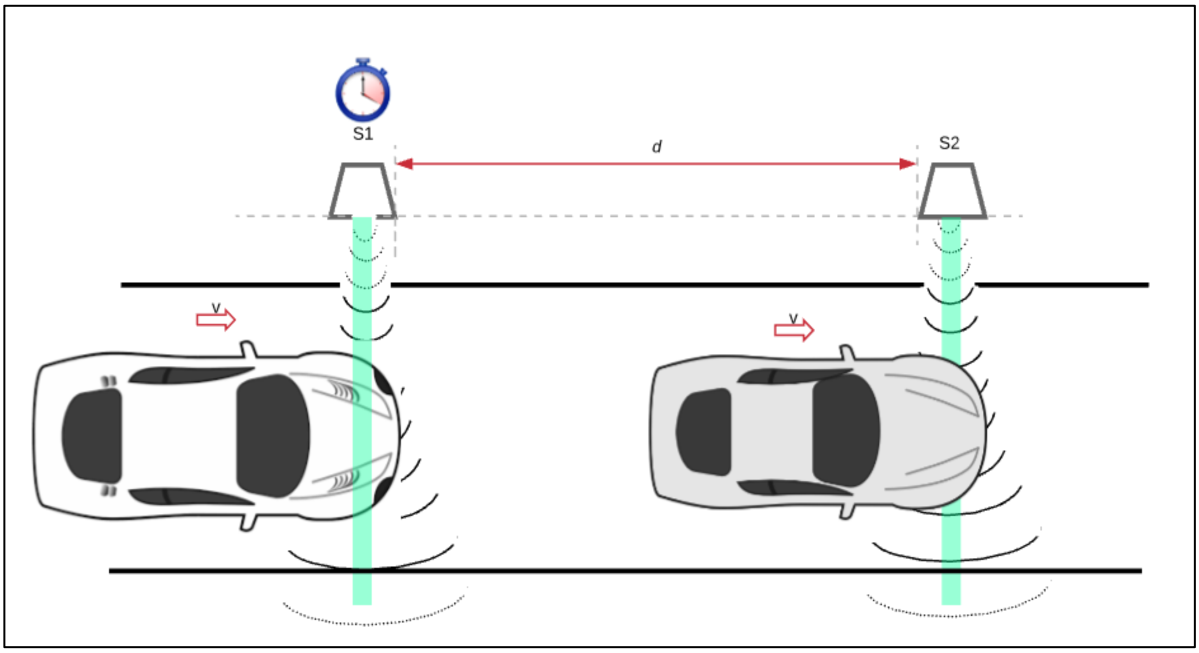


Figure 3.5- Detecting a vehicle

Once the front edge of the vehicle hits the detection range of sensor S1 a programmed timer will be switched on. The vehicle is then considered to move within the length of the road. Once the edge of the vehicle hits the detection range of S2, the timer will be switched off. And the time obtained for the vehicle to pass between the sensors are taken as *t*. The speed of the vehicle is then taken as using the simple formula,

$$v = \frac{d}{t}$$

Where *v* is the speed of the vehicle, distance between the sensors *d* and the time for the vehicle to pass the distance as *t.*

Ultrasonic sensors generate distance data by measuring the time taken to receive reflected ultrasonic waves. The time is directly proportional to the distance between the ultrasonic sensor and the detection object. The delay of triggering timer due to the time taken to receive reflected wave from the ultra sonic sensor can be neglected as it will occur when the timer is switched on and switched off.

3.1.2 Approach used to collect traffic parameters using video processing

Video/image processing can be applied in various contexts of traffic. Traffic surveillance and control, traffic management, road safety and road development of transport policy are some of the examples. This section describes how video processing can be used to capture traffic data such as vehicle speed and vehicle count.

3.1.2.1 *Object detection using video image processing*

There are several ways to process a video footage. Some of commonly used methods are mentioned below.

1) Haar- Cascade method
2) Optical Flow method
3) Background subtraction method

The Haar- Cascade method falls under the Machine Learning (ML) category. Haar-Classifier is basically a supervised pre trained way of recognizing and detecting objects according to detectors on many thousands of selected training images for each view of the object. This method can be accurate as it is pre trained on basis of the images of the objects to be identified. It is required to have two folders of images as Positive and Negative Images. The positive images folder should consist of images which includes the object that has to be identified. The negative images should contain any images without the objects. It is seen in many resources that this methods outcome has several issues such as Detecting unwanted objects in the background and producing garbage values.

The Optical Flow method is used to track moving objects which is done with the velocity of each pixel in the frame or, equivalently, some displacement that represents the distance a pixel has moved between the previous frame and the current frame. The accuracy may vary as the lighting conditions differ as similar as the background extraction part. This method will be a disadvantage when tracking an object of the same color where the same pixel will be present when it's moved.

The choice for this project was the Background subtraction method is very much accurate in detecting moving vehicles and in obtaining the traffic parameters.

Using Background Subtraction method

Background subtraction (BS) is done using the technique of generating a foreground mask which is a binary image containing the pixels belonging to moving objects in the scene by using static cameras. The process is done by subtraction between the current frame and a background model. When given a threshold value to subtract the current frame will consider the previous frame as the background and it will perform the subtraction. Figure 3.6 below shows a visualization of the background subtraction process.

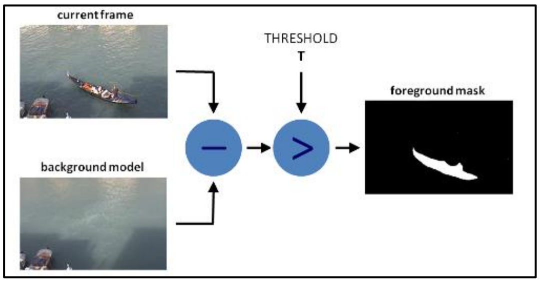


Figure 3.6 – How backgroud substration works

The image Figure 3.7 shows the original image with the background subtraction is going to be applied to. Figure 3.8 shows how background subtraction can be used to extract blobs with less shadow effects. And Figure 3.9 shows how background subtraction can be used to extract blobs with more shadow effects.

Figure 3.7- Original image before Background subtraction-

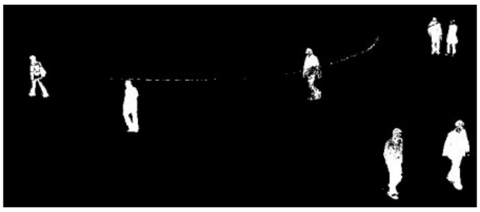


Figure 3.8-Image with background subtraction applied with less shadow effect

Figure 3.9- Image with background subtraction applied with shadow effect

3.1.2.2 *Using Background Subtraction method to detect vehicle speed and count*

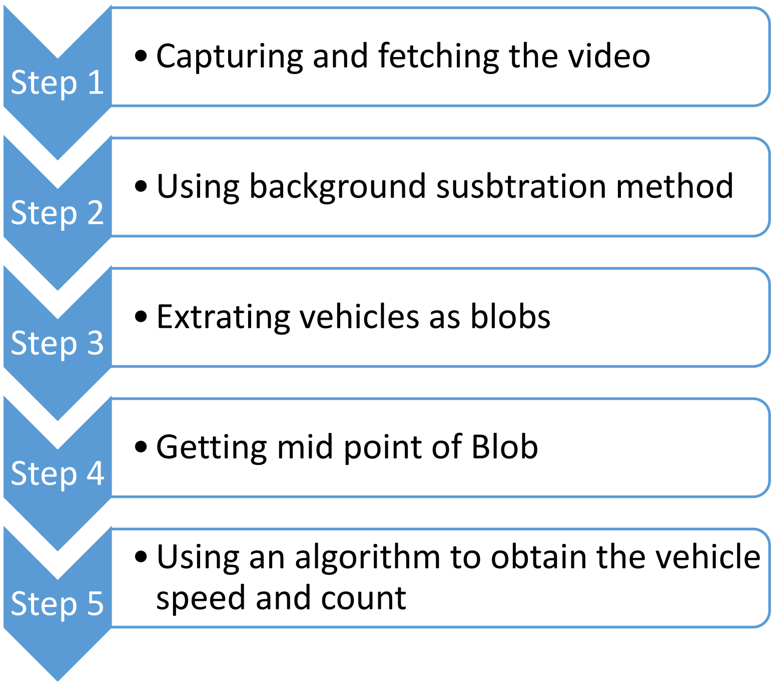


Figure 3.10 – Process followed to calculate vehicle speed and count

Figure 3.10 illustrates the process undergone to reach the calculation of the speed and count using video processing. The process includes video captured is being converted to grayscale inside the code itself where the grayscale video will be used for the background subtraction method. The blob which will be detected is then undergoes several calculations to find the midpoint of it. And finally use an algorithm to obtain the intended parameters. Details of each of these steps are discussed in the following sections of the document.

3.1.2.3 *Capturing the traffic video*

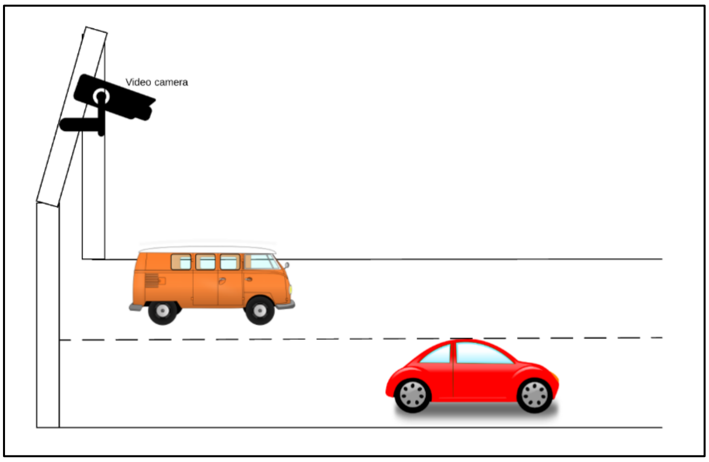


Figure 3.11- Mounting the video camera

As shown in Figure 3.11 the camera must be mounted in such a way that the road can be clearly visible and in a position where less noise / disturbances are caused because vibration from the road or surrounding objects will create unnecessary noise to the video feed. In this research the camera is mounted so that the vehicle flow must be going away from the camera rather than towards the camera. Here the reason to select an overhead mount is to be able to capture more than one lane of the road. The other option is to mount the video camera on the side of the road. This is not suitable as it will not be able to detect the vehicles travelling in the parallel lanes.

3.1.2.4 *Extracting vehicles as blobs*

The purpose of using BLOB extraction is to isolate the vehicles in the binary image. A blob consists of a group of connected pixels. The process of Blob detection is described below.

1. Thresholding: Convert the source images to several binary images by thresholding the source image with thresholds starting at minThreshold. These thresholds are incremented by thresholdStep until maxThreshold.

2. Grouping: In each binary image, connected white pixels are grouped together which considered as the binary blob.

3. Merging: The centres of the binary blobs in the binary images are computed, and blobs located closer than minimum distance between blobs are merged.

After extracting the objects as blobs the next step is to determine which of the detected objects will be considered as vehicles. For this purpose only the blobs that fits a **kernel** (convolution matrix) that fits the shape of an ellipse structure is considered as a vehicle. Here an average size for the ellipse must be defined carefully, since if it's too small unnecessary objects will be detected as vehicles and also larger vehicles will be considered as multiple vehicles. On the other hand, if it is too large and small vehicles such as bikes are not detected.

3.1.2.5 *Finding the mid point of the blobs*

To find the midpoint of a blob first task is to identify the contours of the blob. After finding the contours the Moments in an Image helps to calculate features like center of mass of the object and area of the object.

Image moment is a certain particular weighted average (moment) of the image pixels' intensities, or a function of such moments. In this research, the center of the blob is taken by the general formula used in moments to calculate centroid. Here the centroid is a point which is a pixel value.

Centroid point: $\{ x, y \} = \{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \}$

3.1.2.6 *Algorithm to count vehicles and obtain vehicle speed*

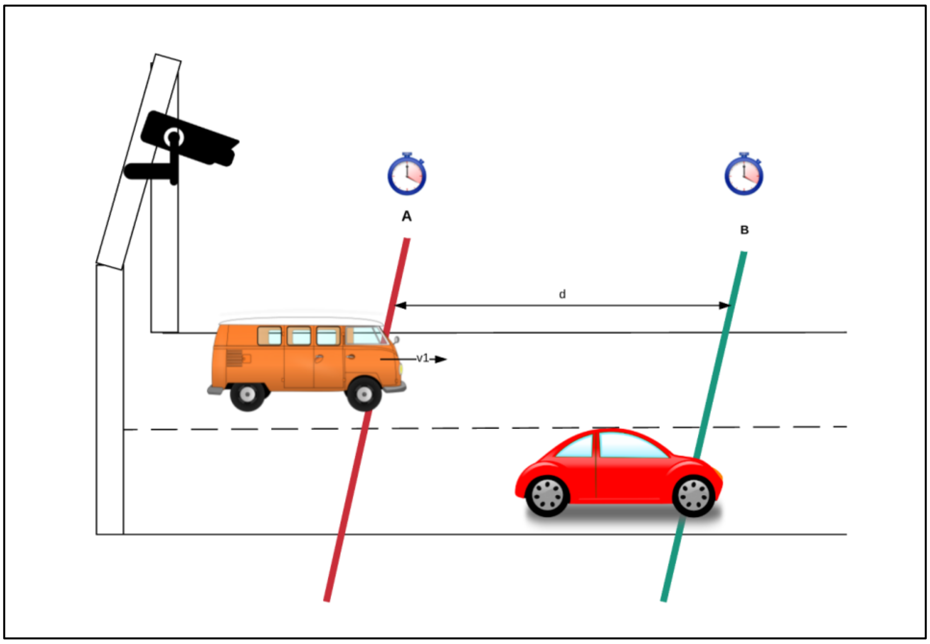


Figure 3.12- Approach used to detect vehicles

As shown in Figure 3.12 two straight lines A and B that complies with the general equation of a straight line $y = mx + c$ are maintained at a distance of d. After extracting vehicles as blobs and determining their centroids, the centroid of the object is used in calculating the speed and count of vehicles.

The mathematical theory "Lines which are horizontal have zero gradient" is used to determine that a detected object has passed the lines A and B.

The gradient of the line A is define as

$$M_A = \frac{y_2 - y_1}{x_2 - x_1}$$

The gradient of line B is defined as,

$$M_{B\,=} = \frac{y_4 - y_3}{x_4 - x_3}$$

As both lines A and B are horizontal, their gradient M_A and M_B is zero.

When an object with its centroid point x, y has entered at line A which does not exist at the frame l_n-1 then appears at the next frame l_n satisfies the equation on the straight line A, the system creates a record counting the object as a vehicle. Simultaneously a timer is switched on at the time t_0.

At line B when the object which exist at the frame l_n-1, with its centroid point x, y that satisfied the equation of line B disappears at the next frame l_n the timer is switched off and the time at that point is taken as t_n.

The speed of the vehicle is then calculated as,

$$\textbf{Speed (v)} = \frac{d}{t}$$

Where,

d – The distance between the lines A and B

$t = t_n - t_0$; The total time taken for the object to travel a distance ***d***

3.1.2.7 *Storing vehicle speed and count obtained using video processing*

Using video image processing the vehicle speed, count are obtained. The average speed of vehicles and the total count of vehicles that passed during each 2mins, is fed in to the database. These information are stored in the database for the purpose of using them in traffic prediction. Figure 3.13 shows how this is done.

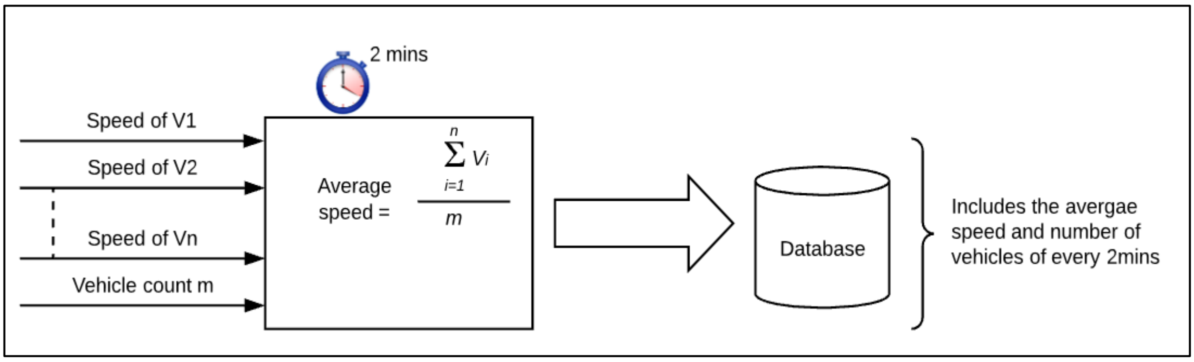


Figure 3.13- Model of storing information captured by video processing

3.2 Traffic prediction using the current vehicle inflow

This research attempts on predicting the occurrence of a traffic, by analyzing the current traffic conditions. A machine learning approach is used for the purpose of identifying traffic using a video feed. These results are then used in a prediction algorithm to determine the future occurrence of the traffic. The solution can be broken down into two sections, where Section 1 is about selecting and training the model to detect traffic and Section 2 is about using the model to predict traffic. These sections are discussed in detail.

3.2.1 Section 1: Selecting and training the model to detect traffic

The training mechanism and the model is illustrated by Figure 3.14.

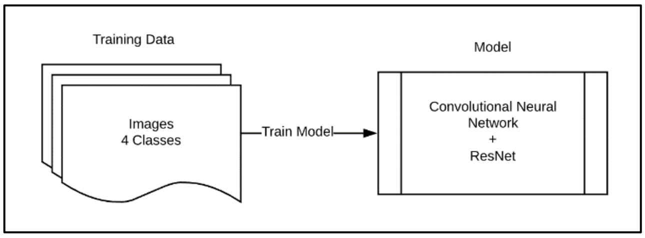


Figure 3.14- Training the model

3.2.1.1 *Training data*

For the purpose of training the model a set of training data is required. In this research a data set of 4000 images of road traffic were taken with images sizes 1440x1080 pixels. These images were then grouped into 4 categories which are mutually exclusive. The categories are listed below.

1. High Density
2. People crossing
3. Stopped vehicles
4. Vehicles from bi-roads

Sample images for each of the above categories are shown by Figure 3.15, Figure 3.16, Figure 3.17 and Figure 3.18.

Figure 3.16- Image of High density

Figure 3.15- Image of people crossing

Figure 3.18 -Image of stopped vehicles

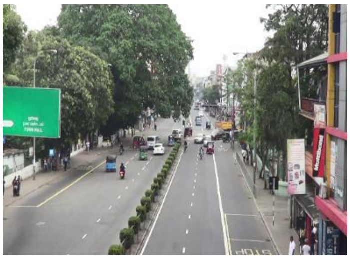


Figure 3.17- Image of vehicles entering from Bi-roads

3.2.1.2 *The Model*

As the training algorithm of the model ResNet-18 is used. This model has good performance to run on a personal computer that has a GPU (Graphical processing unit). The following section explains the process of selecting the above model.

3.2.1.3 *Selecting Convolutional Neural Networks*

It is known that Regular Neural Nets do not scale well to full images. A regular neural network is manageable for images of smaller size, for eg: 32x32x3 (32 wide, 32 high, 3 color channels), but larger the image greater the number of weights and parameters would add up quickly. Thus will cause overfitting very quickly.

As images from traffic scenes are quite large (chosen image size 1440x1080), Regular Neural networks does not suit the job. Hence CNN is chosen as it is a neural network designed to handle graphical content. Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way.

3.2.1.4 *How Convolutional Neural Networks work*

Unlike a regular Neural Network, the layers of a convolutional network have neurons arranged in three dimensions: **width, height, depth**. This allows the neurons in a layer to connect only to a small region of the layer before it instead of having a fully connected manner like in regular neural networks. As shown in Figure 3.19 the convolutional network architecture residues a fill image into a single vector of class scores arranged along the depth dimension.

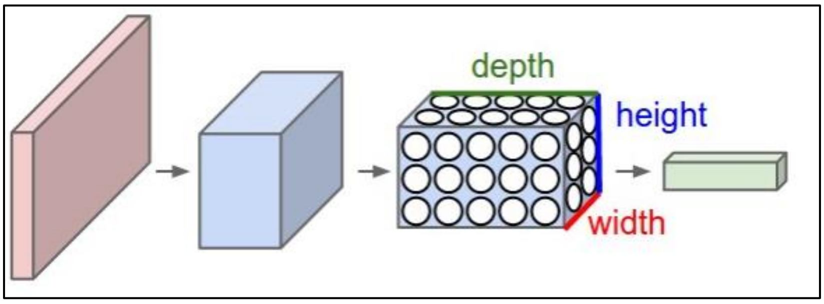


Figure 3.19- Concept of Convolutional neural networks

3.2.1.5 *Selecting the activation function for the Convolutional Neural network*

Some of the popular activation functions are,

1. Sigmoid or Logistic
2. Tanh—Hyperbolic tangent
3. ReLu -Rectified linear units

Sigmoid Activation function is easy to understand and apply but it has major reasons it is not suitable for the purpose is due to the Vanishing gradient problem, saturate and kill gradients also have slow convergence. Hyperbolic Tangent function is preferred over sigmoid function. However still it suffers from Vanishing gradient problem. On the other hand, ReLu has become very popular in the last few years. The activation is simply thresholder at zero. It rectifies vanishing gradient problem. Hence it is very much used in deep learning Models. Besides have the limitation of being able to use only within hidden layer, this was chosen as the activation function for the purpose of this research.

3.2.1.6 *Residual Networks*

The deeper the network, the more difficult it becomes to train. In this research, the CNN is chosen as the backbone of the model and it consists of a large number of layers, which makes it difficult to train and the accuracy will be compromised and has a possibility for higher training error.

To minimized the error and for the purpose of gaining more accuracy, ResNet is used to support Residual networking. ResNet provided by Microsoft is a residual learning framework to ease the training of networks that are substantially deeper than those used previously. It explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. [16] The main difference in ResNets is that they have shortcut connections parallel to their normal convolutional layers. Using the residual network architecture proposed by [16] several ResNet models were developed supporting different number of layers by each model. ResNet-18, ResNet-200 are some of the models available.

3.2.1.7 *Using Residual Network ResNet 18*

ResNet-18 is an implementation of the architecture provided by Microsoft on ResNet. This is an already trained model with ImageNet. This Residual network consists of 18 layers. This model allows training on a custom image set. In this research the data set of traffic images are trained as mention in section 3.1.2.1. Two directories, Train and Val (Validation) needs to be maintained. Where under each contains sub folders with the four categories High Density, People crossing, Vehicles from bi-roads and Vehicles stopped.

3.2.2 Section 2: Using the model to predict traffic

Figure 3.20 illustrates the process of traffic detection to traffic prediction. This involves a number of steps. These steps are discussed in detail in the upcoming topics.

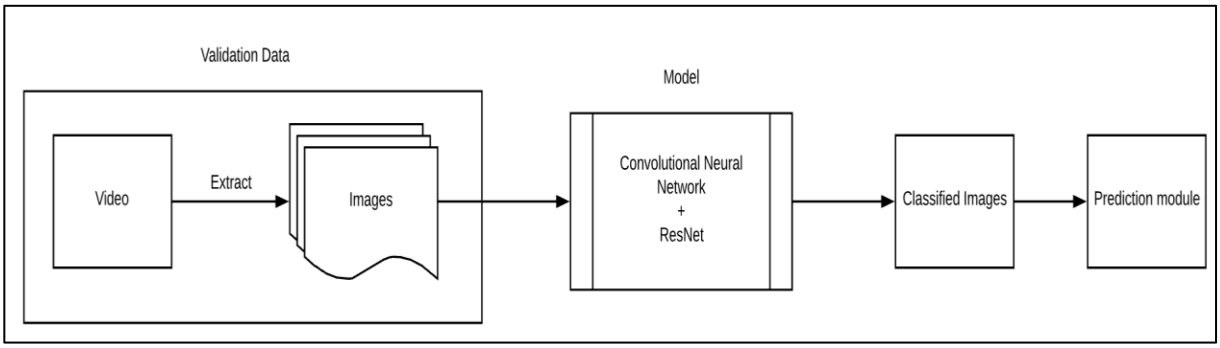


Figure 3.20- Using the model to predict traffic

3.2.2.1 *Validation Data*

For the purpose of evaluation, first the video stream needs to be converted to a number of images. This is done using a video decoder. The video stream is converted to 60 images per minute by capturing snapshots frame per second. These images then becomes input to the ResNet-18 trained model.

3.2.2.2 *Classification and Prediction*

After the images from the validation data set is processed by the model a classification label is assigned to each of those images. These data are then stored for the purpose of using them in the traffic prediction. The prediction module shown in Figure 3.20 is further elaborated in Figure 3.21below.

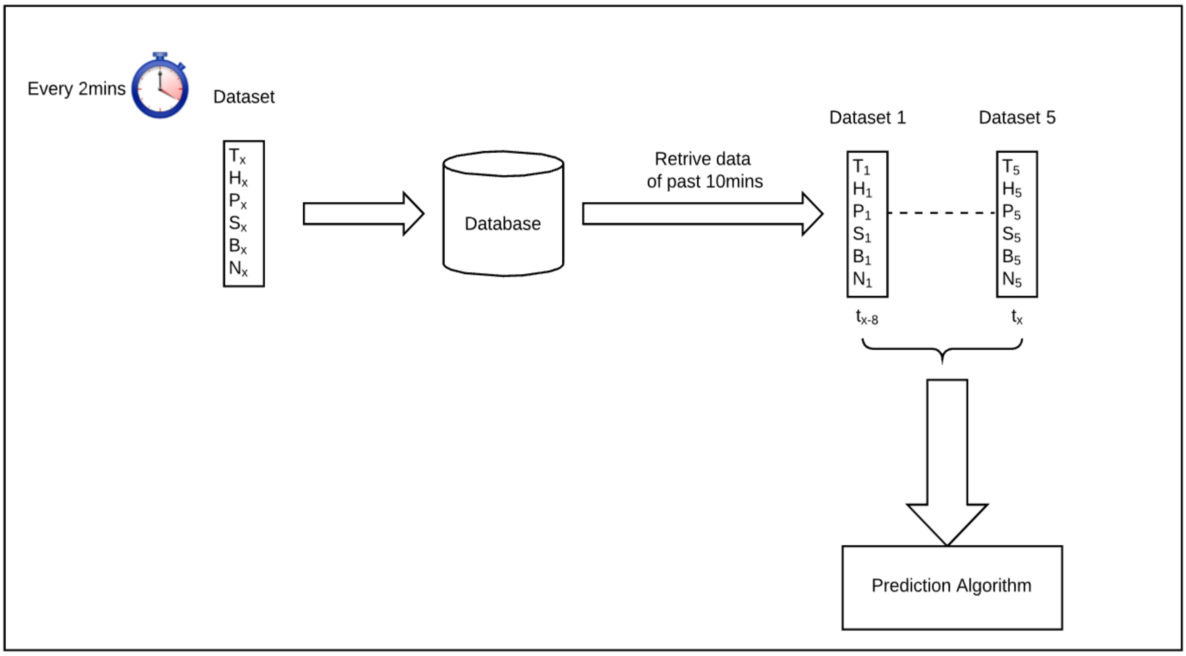


Figure 3.21- Process of storing traffic detection data

As shown in Figure 3.21, every 2 minutes a dataset with the following data are stored in the database.

1. Total number of frames that has traffic congestion
2. Total number of frames that doesn’t have traffic congestion
3. Number of frames labelled as high density
4. Number of frames labelled as people crossing
5. Number of frames labelled as stopped vehicles
6. Number of frames labelled as vehicles from bi-roads

At the same time the traffic prediction is done every 2 minutes considering the traffic data obtained within the last 10mins from the current time. This means the past 5 datasets stored for each time slot t_x , t_{x-2} , t_{x-4} , t_{x-6} , t_{x-8} where t_x is the current time are taken for the calculations. The output of the traffic predication gives the traffic congestion details for the next 5-10minutes.

Here initially a buffer of 10 minutes is required as we need to have our first 5 datasets to make the very first prediction. Hence initially the prediction happens after 10mins but there after the prediction happens every 2 minutes.

The reason to consider 2mins is to be have a reasonable amount of frames to process. A vehicle travelling at average speed of 40-50kmph consumes about 10 frames. Hence, to have a proper amount of data with minimal variation in traffic, 2mins was considered. Larger the duration, the variation in traffic is high and would lead to an inaccurate prediction.

Next with all the data being available, the percentages for each of the data categories are calculated. The calculations of these values are shown below.

Total traffic % (**T**) = $\frac{\sum_{i=1}^{5} T_i}{Total\ No.of\ Frames}\ X\ 100\%$

Less or no Traffic % (**N**) = $\frac{\sum_{i=1}^{5} N_i}{Total\ No.of\ Frames}\ X\ 100\%$

High density traffic % (**H**) = $\frac{\sum_{i=1}^{5} H_i}{Total\ No.of\ Frames}\ X\ 100\%$

Pedestrian crossing % (**P**) = $\frac{\sum_{i=1}^{5} P_i}{Total\ No.of\ Frames}\ X\ 100\%$

Stopped vehicles % (**S**) = $\frac{\sum_{i=1}^{5} S_i}{Total\ No.of\ Frames}\ X\ 100\%$

Vehicles from bi roads % (**B**) = $\frac{\sum_{i=1}^{5} B_i}{Total\ No.of\ Frames}\ X\ 100\%$

Where $\mathbf{T_i}$,$\mathbf{N_i}$,$\mathbf{H_i}$,$\mathbf{P_i}$,$\mathbf{S_i}$ and $\mathbf{B_i}$ are data obtained from the i^{th} dataset.

The calculated values then traverse through the following algorithm in Figure 3.22 and the traffic output gives one of the following labels.

- Very high traffic during next 10mins
- High traffic during next 10mins
- High traffic during next 5mins
- Medium traffic during next 10mins
- Less traffic during next 10mins

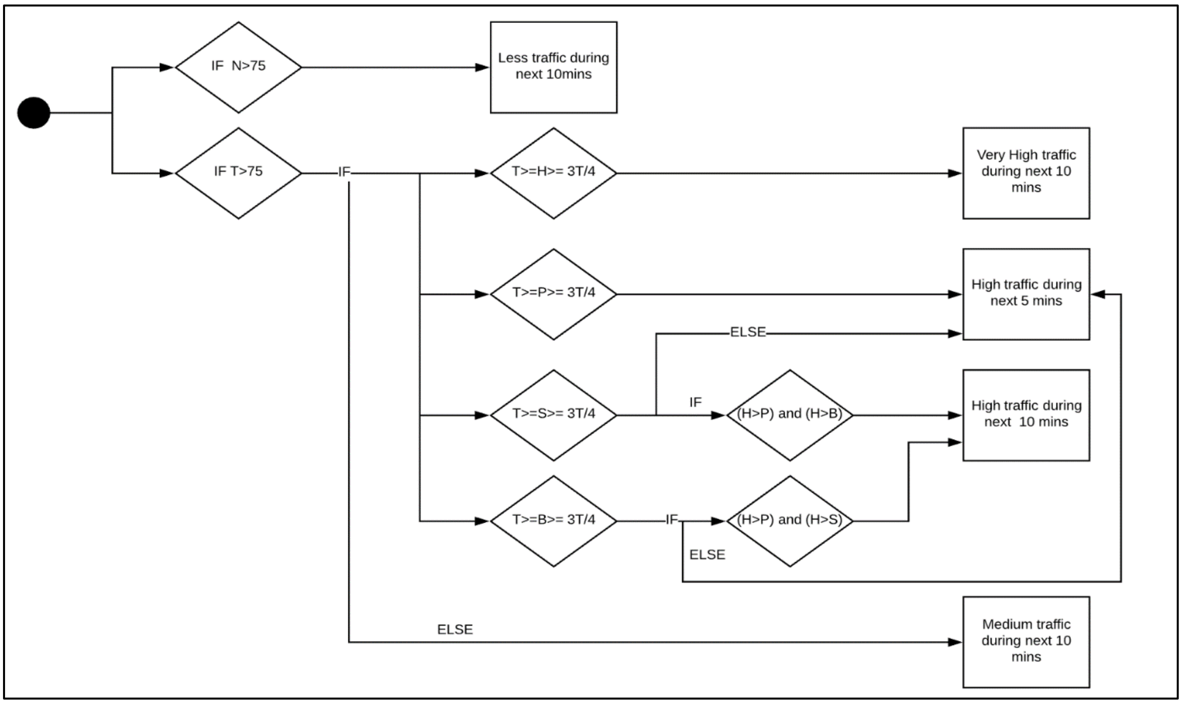


Figure 3.22- Traffic Prediction Algorithm

As shown in Figure 3.22 the output is given for 5 or 10mins based on an analysis done for the such variations. As an example traffic generating scenarios such as people crossing, the duration of the prediction ahead is given for 5 minutes and not 10mins. Even though ideally we take of the past 10mins and predict the data of the next 10mins ahead in most cases, this does not apply to some of these scenarios.

For instance, an average time take for a pedestrian is considered as 12sec, this average time is taken through observation from crossing times allowed in various traffic signals around Colombo.

Such a small delay does not become a reason to cause a traffic congestion. But, with the increase of this time, results in traffic generation. If we consider the average time for vehicle to travel a distance d is t , and the delay from pedestrian crossing is z, and an average pick up time for the vehicle is p, The total time T a vehicle takes to travel the same distance d becomes T= t+z+p. With the value of z increasing, T also increase which results in generation of traffic. Also as seen in Figure 3.22Figure 3.22 the initial benchmark value of 75 was chosen to because as it is the 3rd quartile of the maximum percentage.

4 IMPLEMENTATION

The implementation is explained in two areas namely, **collecting traffic inflow data and traffic prediction using the current vehicle inflow.**

4.1 Collecting traffic inflow data

4.1.1 Collecting traffic inflow data using ultra Sonic sensors

4.1.1.1 *Implementation software and language*

Arduino an open source software IDE was used to implement the setup of ultrasonic sensors.

4.1.1.2 *Setting up the sensors*

Sensors used :HC-SR04 ultrasonic sensor

The sensors were connected to the Arduino board (shown in Figure 4.1) and implementation of the vehicle detection and speed detection was done using the Arduino software. The *Trig* pin will be used to send the signal and the *Echo* pin will be used to listen for returning signal. It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board.

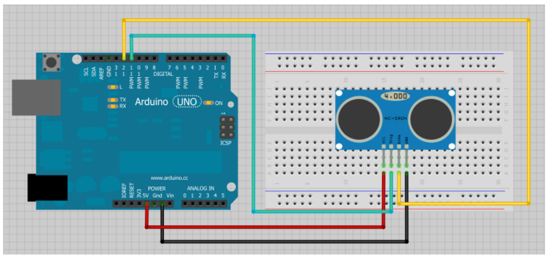


Figure 4.1 - Arduino board

4.1.1.3 *Computations*

Computations were carried out as with the formulas given in 3.1.1.4. The distances are taken in centimeters and the time is considered in milliseconds as a prototype was used to evaluate the result. The ultimate output of the vehicle speed was given in kmph.

4.1.2 Collecting traffic inflow data using video processing

4.1.2.1 *Implementation software and language*

OpenCV (Open Source Computer Vision Library) an open source computer vision and machine learning software library is used for video image processing.

4.1.2.2 *Computations*

For the purpose of background substraction the following methods of openCv was used.

1) cv2.BackgroundSubtractorMOG()
2) cv2.BackgroundSubtractorMOG2()

Basically the MOG function gives the output image just the blobs without much of the effect of the shadows in the frame but MOG2 gives an output with the shadows in the frame which can be shown as Noise. The MOG method was used in this research.

To calculate the mid point of the blob the moments of the images was taken using the function cv2.moments(c). There by taking the Mid point if the blob using

```
center_x = int(M["m10"] / M["m00"])
center_y = int(M["m01"] / M["m00"])
```

For structuring elements the method **cv2.getStructuringElement()** was used get the desired kernel. IN this research the shape Elipse was used and the size was specified as 5 x 5

4.2 Traffic prediction using the current vehicle inflow

The implementation of machine learning approach to detect traffic is described in this section.

4.2.1 Implementation software and platforms

Torch is a scientific computing framework with wide support for machine learning algorithms that puts GPUs first. It has an underlying C/CUDA implementation. Hence in this research Torch platform is used to support and run the ResNet-18 model. Also *CUDA* which is a parallel programming platform is need train the ResNet model. As the scripting language Lua is used. For the purpose of converting videos to a set of images Video-Decoder available in Torch Toolbox is used.

4.2.2 Computer requirements

A personal computer with a GPU with CUDA support is needed. A GPU is required to accelerate deep learning process.

5 EVALUATION AND RESULTS

The evaluation is explained in two areas namely, **collecting traffic inflow data and traffic prediction using the current vehicle inflow.**

5.1 Collecting traffic inflow data

5.1.1 Collecting traffic inflow data using ultra Sonic sensors

A prototype model was assembled to take form of a road and traffic environment similar to the real world. Remote control cars were used to generate traffic conditions.

The prototype built is illustrated by Figure 5.1

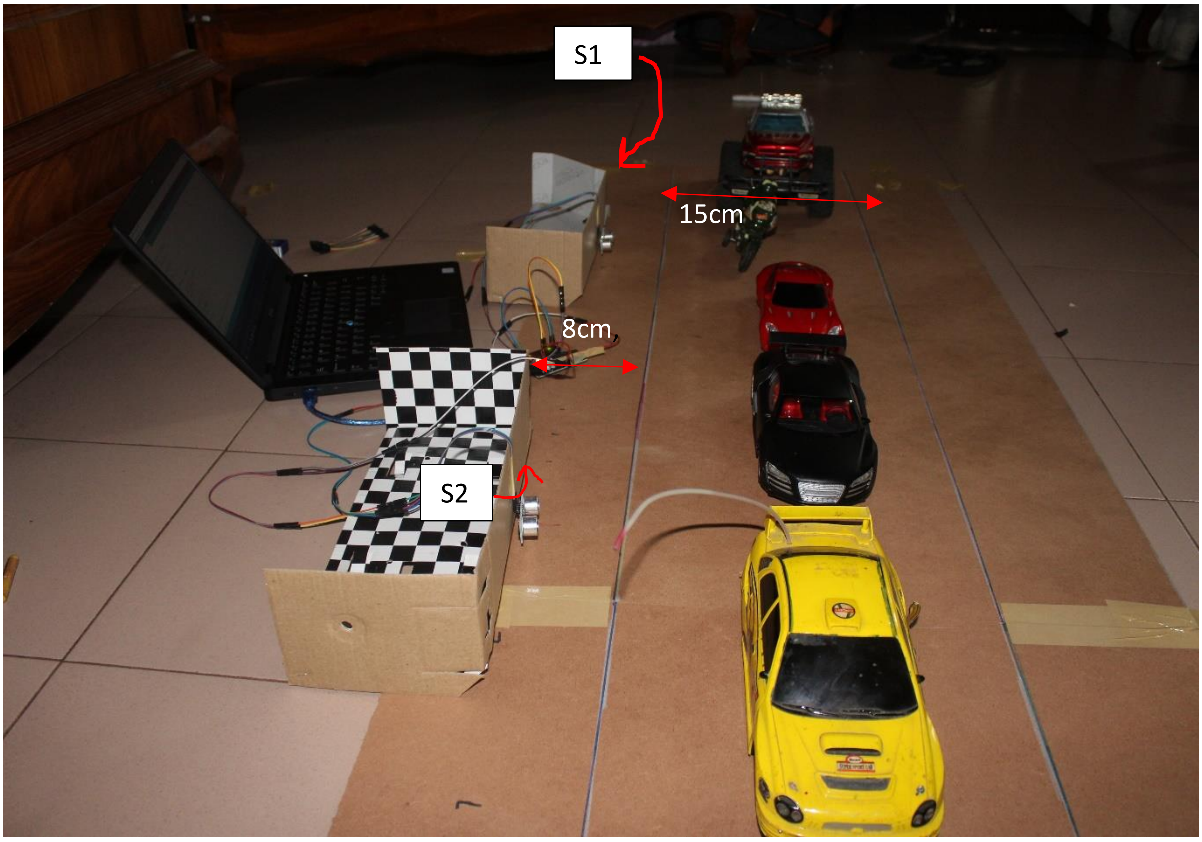


Figure 5.1 - Prototype for evaluating traffic using ultra sonic sensors

The traffic environment was set up on a wooden board. The road was drawn with a width of 15cm and the length of the road segment was taken as 135cm to represent a single lane road. The ultra sonic sensors were placed 8cm away from the road line. S1 was placed 30cm away from the start position of the road. And the initial distance between the ultra sonic sensors was taken as 30cm. The reason to select such distance was considering the size of the largest toy vehicle taken for the experiment. Three remote control cars, a jeep and bike shaped vehicle was taken for this experiment. The remote control cars are assumed to have a constant speed.

The following aspects were evaluated,

1. Accuracy of vehicle speed when the distance between the sensors is increased.
2. Accuracy of multiple vehicles passing
   - Vehicles moving parallel - Even though it’s a single lane road, in Sri Lankan traffic conditions smaller vehicles will take the opportunity to squeeze in the least available space.
3. Random pedestrian crossing

The Table 5.1 shows the symbols used to identify each vehicle.

Table 5.1 - Vehicles used in the prototype

Vehicle	Representation Symbol
Yellow car	V_1
Truck	V_2
Red car	V_3
Black car	V_4
Motor bike	V_5

5.1.1.1 *Evaluating Accuracy of vehicle speed when the distance between the sensors is increased.*

Hypothesis: If the distance between the sensors are increased the accuracy of the speed increases.

One car at a time was sent along the road and the vehicle speed detected from the ultra sonic sensors were recorded. While sending the cars speed detecting device was used to capture the speed of each passing vehicle. There after the speeds obtained from the ultra sonic sensor and the speed detector was compared. This experiment was carried out to verify the hypothesis of "If the distance between the sensors are increased the accuracy of the speed increases"

Initially to start off two ultra sonic sensors were placed at a distance of 30cm, which was the size of the largest toy vehicle taken for the experiment. The experiment was carried out by increasing the distance between sensors by 15cm to see if there was any effect in the accuracy of the traffic parameters obtained. The experiment was repeated 6 times by increasing the distance upto 105cm between S1 and S2.

Graphs in Figure 5.2, Figure 5.3, Figure 5.4, Figure 5.5and Figure 5.6 represents the speeds obtained with the varying distance of the ultrasonic sensors for vehicles V_1 , V_2 , V_3 , V_4 and V_5 respectively.

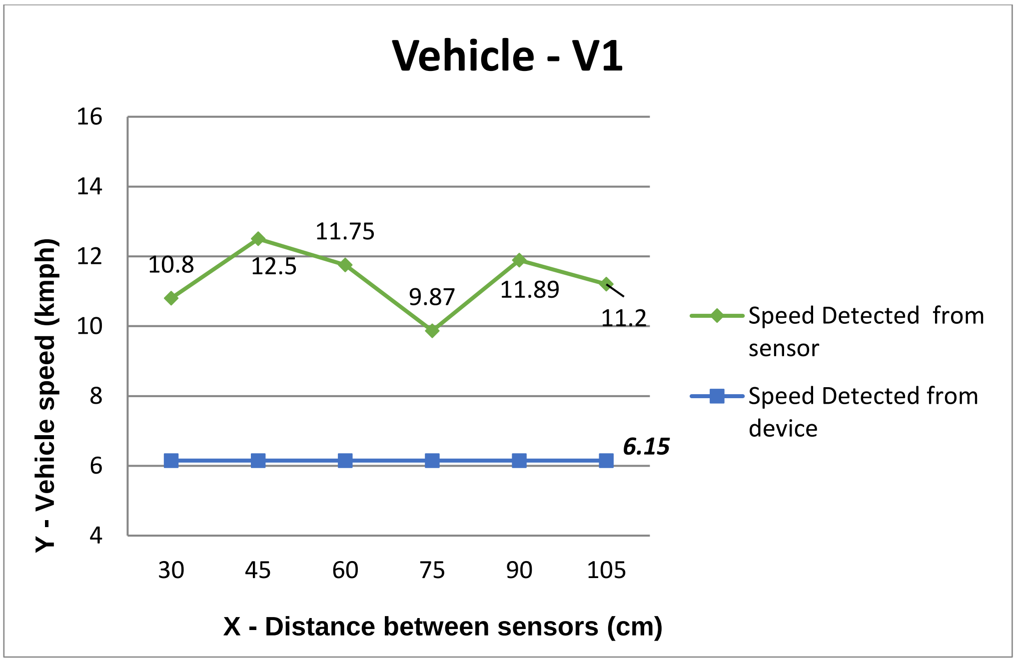


Figure 5.2 – Speed detected for V1 with different distances between the sensors

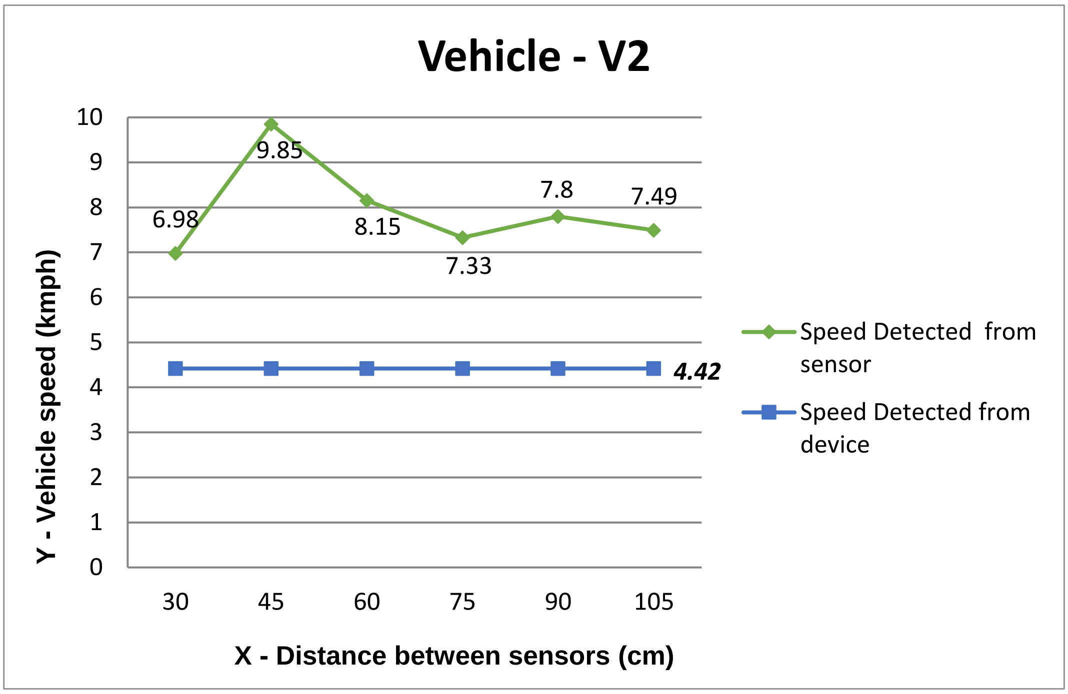


Figure 5.3 - Speed detected for V2 with different distances between the sensors

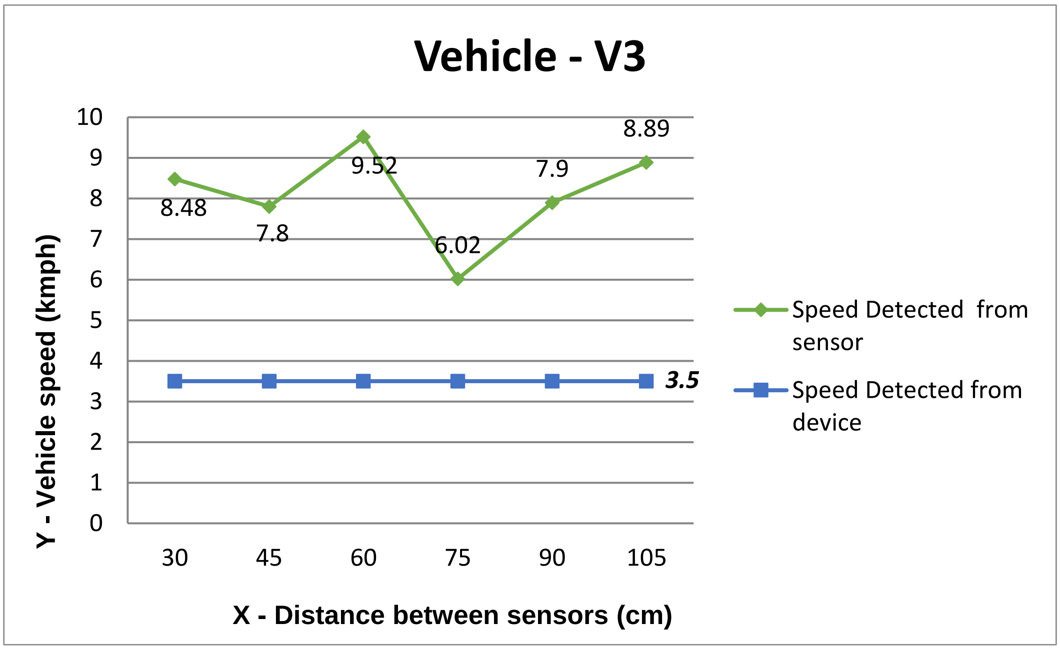


Figure 5.4- Speed detected for V3 with different distances between the sensors

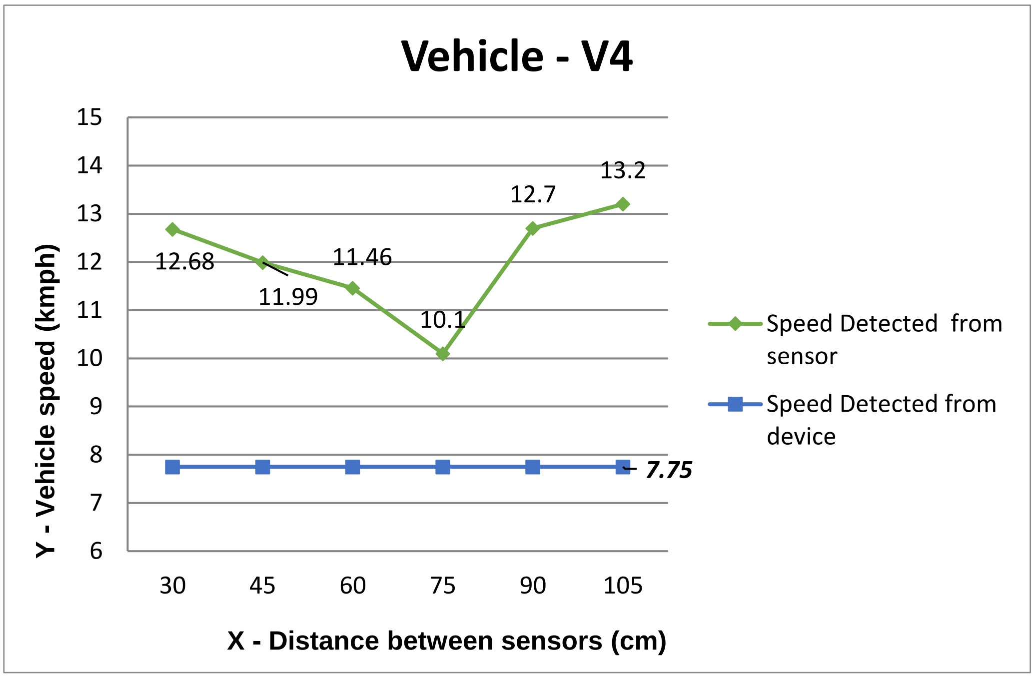


Figure 5.5- Speed detected for V4 with different distances between the sensors

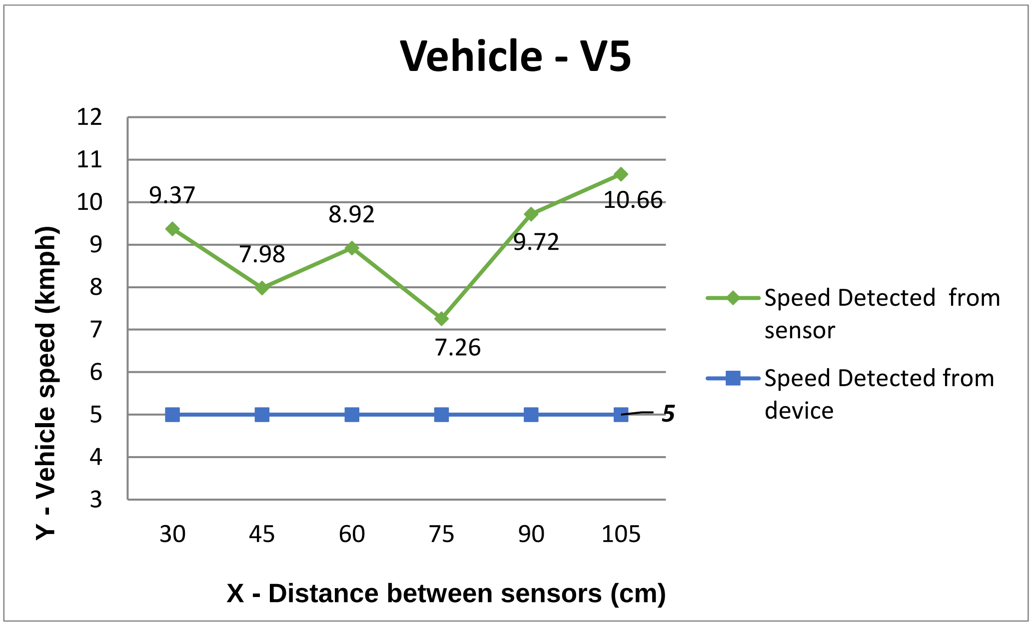


Figure 5.6 - Speed detected for V5 with different distances between the sensors

By comparing the results of Figure 5.2, Figure 5.3, Figure 5.4, Figure 5.5 and Figure 5.6 for each distance the sensors were kept, the average error in speed are plotted in the graph shown by Figure 5.7.

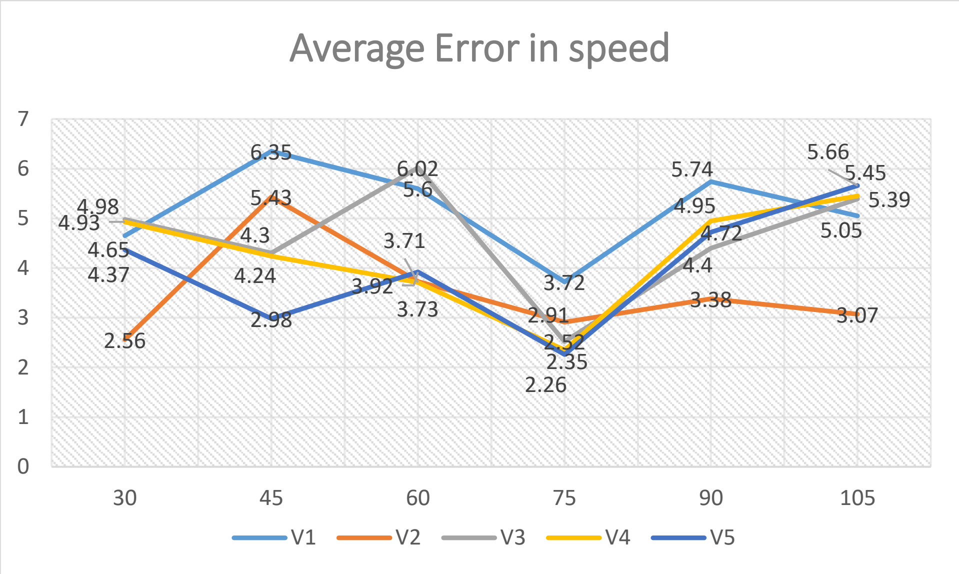


Figure 5.7 - Average error in speed calculated by ultra sonic sensors

After obtaining the speeds detected at each distance, the error between the actual speed vs the speed detected from the sensors were compared. By the above graph it is shown that the least error occurs at the distance of 75cm which has an average error in speed +2.752 kmph. It was observed that when the distance of the sensors were less, the accuracy of the speed detected was not very accurate as well as once the distance between the sensors were increased the accuracy of the speed has decreased. Therefore the hypothesis can be considered can be stated as true, with the following finding.

Using this information an ideal distance between the sensors in relation to the length of the road segment chosen.

% distance between sensors $= \frac{distance\ between\ sensors\ that\ gave\ most\ accurate\ results}{Length\ of\ the\ road\ segment}$ x 100%

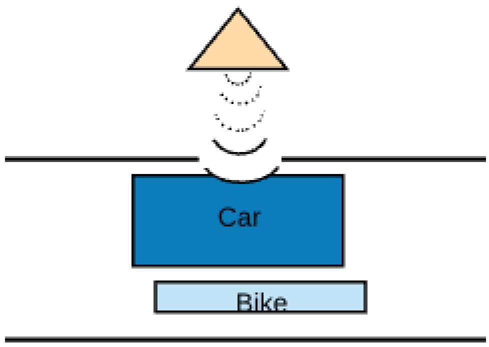
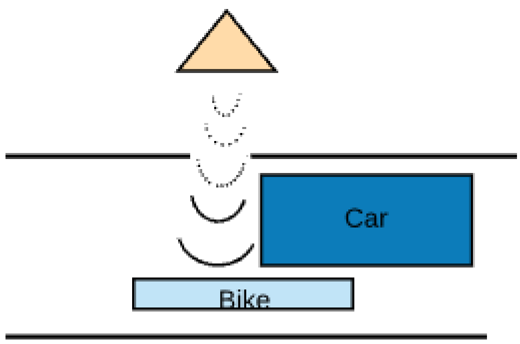
% distance between sensors $= \frac{75cm}{135cm}$ x 100% = 55.556%

When distance between the sensors was 55.556% from the length road segment the vehicle speed is said to be most accurate.

5.1.1.2 *Evaluating multiple vehicles passing*

As per the previous experiment the distance of 75cm between the sensors yielded the most accurate speed predictions was considered. Vehicles moving parallel was considered therefore a few test scenarios were run to see how accurately the set up will capture the traffic parameters. The Table 5.2 shows the results,

Table 5.2 - Results from evaluating multiple vehicles passing

Scenarios	No of vehicles detected	Comment
The car approaching first Figure 5.8  *Figure 5.8- Sensors test Scenario 1*	1	The car which approach the sensor first and is also on the side of the sensor was detected but the bike was not detected.
The bike approaching first Figure 5.9  *Figure 5.9 - Sensors test scenario2*	1	The bike was detected first as it crossed the detection range of the sensor first. But here the speed of the bike was not accurate due to partial echo received as the car obstructed the reflected wave.

By this experiment it was observed that the sensor failed to identify vehicles when two vehicles pass the sensor detection point at the same time. As the detection signal hits the first vehicle it fails to capture the second vehicle that passing the sensor at the same time. Hence this set up will be inaccurate for such scenarios. Therefore, we can conclude that vehicles travelling in parallel will failed to identify by this set up.

5.1.1.3 *Evaluating Random pedestrian crossing*

As it is a common behavior in Sri Lankan road, pedestrians often cross from places other than the designated pedestrian crossings. Experiment was carried out to capture the results of this.

Case 1: Pedestrian crossing outside the detection range of S1 and S2 (shown by Figure 5.10)

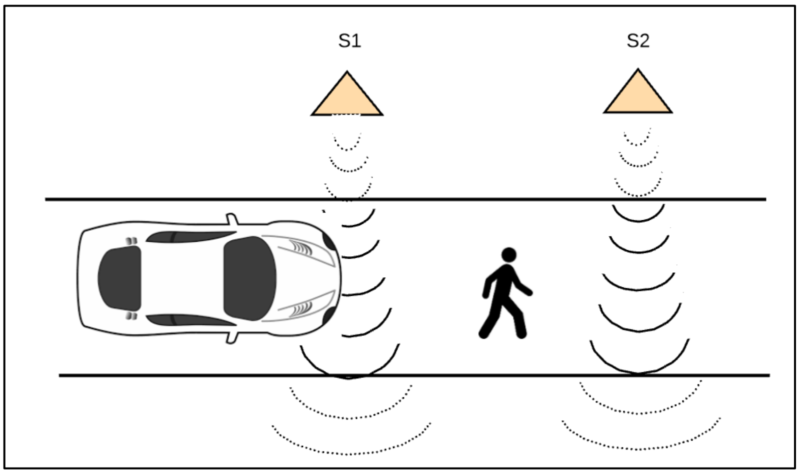


Figure 5.10- Person crossing outside the detection range

Here it was observed that there was no effect on the accuracy of the speed of the moving vehicle. The speed of the vehicle was calculated correctly by the sensors.

Case 2: Pedestrian crossing in the detection range of S1 (shown by Figure 5.11)

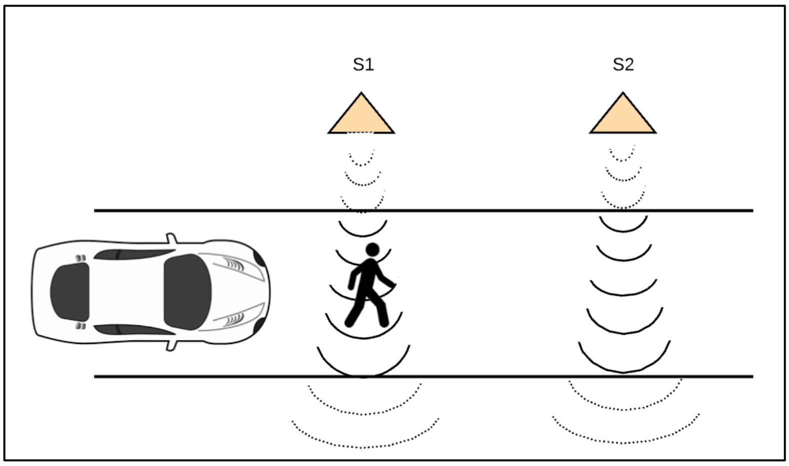


Figure 5.11- Person crossing within the detection range of S1

Here the ultra sonic sensor S1 captures the moving pedestrian as a vehicle and starts the timer and since the pedestrian does not cross over the S2 all the vehicles passing after the pedestrian crossing yield wrong speeds.

Case 3 : Pedestrian crossing in the detection range of S2(shown by Figure 5.12)

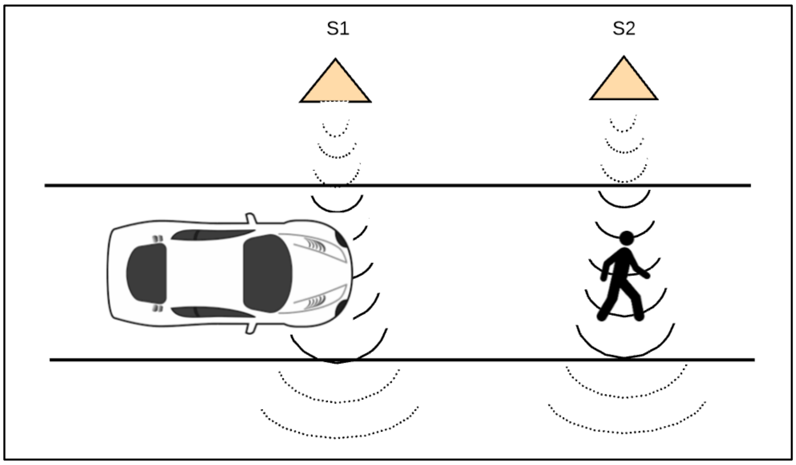


Figure 5.12- Person crossing within the detection range of S2

Here it was observed that if no vehicles are within the detection range of S1, S2 or in between the two sensors, there is No effect from the pedestrian crossing over the detection range of S2.

However if a vehicle is within the detection range of S1 or between the sensors S1 and S2 the, once the pedestrian hits the detection range of S2 , the ultra sonic sensor considers as a vehicle has now crossed S2 and hence calculates a speed. Which is incorrect since the vehicle has actually not passed S2 yet.

5.1.1.4 *Summary of the evaluation carried out using ultra sonic sensors*

Considering the usage of Ultra sonic sensors to detect vehicle speed can be determined as not suitable for Sri Lankan transportation context. This set up yielded accurate speeds when the only objects on the road was vehicles, and when no parallel vehicles moving on the road.

If we consider a common behavior in Sri Lankan road context, we see pedestrians crossing all over the place, not necessarily using the crossing. This set up provides a way to avoid pedestrians in the pavement area, but limiting the detection range of the sensor only to the width of the road.

But if pedestrian crosses the road right in front of the sensor, there will be a false detection as the pedestrian now falls within the detection range. Moreover, this setup was proven not suitable when vehicles travel parallel even in a single lane road. The above facts of wrong pedestrian crossing and parallel driving leads to the conclusion that ultra sonic sensors cannot be used for traffic data collection considering in Sri Lankan traffic context.

5.1.2 Collecting traffic inflow data using video processing

The evaluation was carried out using several video feed captured in the real world environment.

5.1.2.1 *Obtaining the video*

The Pandura town area road was selected as the traffic route to capture traffic video for the purpose of evaluating the results.

The camera was mounted in the passenger crossing overhead bridge and several videos of traffic were obtained.

A total of 4 videos with a duration of 15mins each were obtained during week days and a Sunday during to capture both peek and non-peak traffic flows. Video 1, Video 2 were taken on a weekday. Whereas Video 3 and Video 4 were taken on a weekend.

For the calculation of speed and count the two lines Red and Green were maintained. The distance between the two lines were taken as 5m.

5.1.2.2 *Evaluation the vehicle count*

The vehicle count obtained by each of the traffic videos are represented below. The accuracy of them are determined by comparing with the actual count of vehicle which was counted manually.

Graph with Actual count vs Detected count is shown in Figure 5.13.

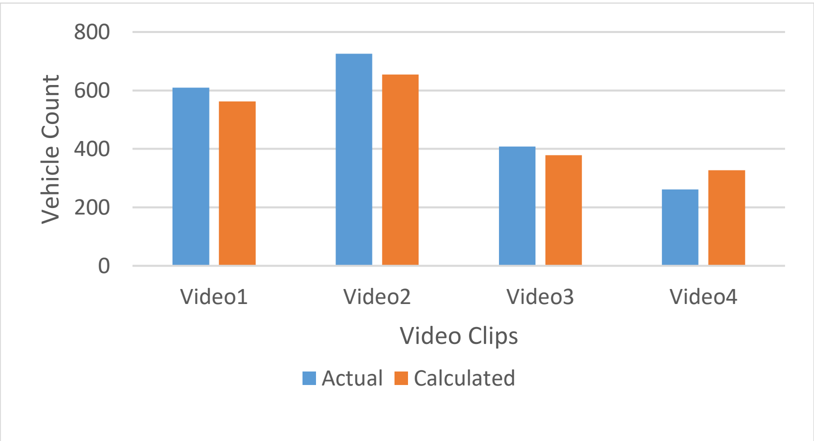


Figure 5.13- Results of vehicle count detected using video processing

By analyzing the obtained results it was noted that with video feeds 1,2 and 4 it was observed that the count of the vehicles detected through image processing is more than the actual count of vehicles. Reasons for the inaccuracy as stated below.

1. Vehicles travelling in the middle of the two lanes.

When a vehicle travels in the middle of the two lanes the centroid of the vehicle lies on both the lines. Therefore the vehicle is counted twice, once by each lane. An example for this is shown in Figure 5.14. Even though the actual count at that point was 29, it is counted as 30 because a three-wheeler in front was counted twice.

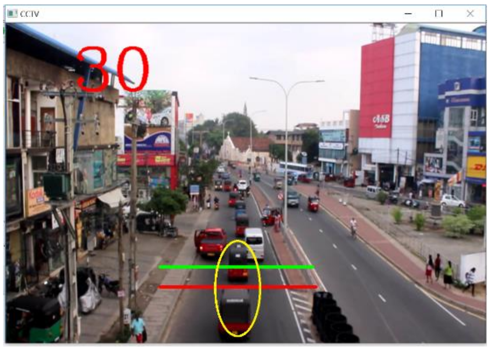


Figure 5.14- Vehicles travelling in the middle of two lanes

2. Larger and lengthier vehicles

Another reason for the detected vehicle count to be more than the actual count of vehicles is that larger vehicles such as trucks are detected as multiple vehicles rather than one. This is due to the kernel shape and size defined. An example for this situation is given in Figure 5.15.

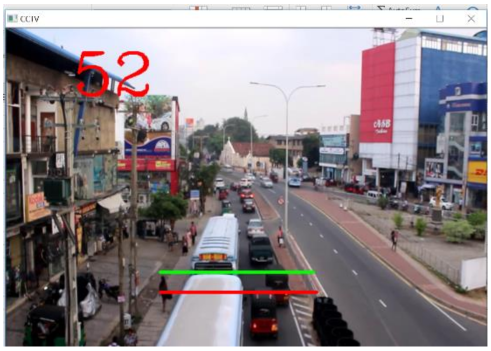


Figure 5.15- Detecting larger vehicles

3. People crossing randomly

Another scenario where the count calculated was more than the actual count was, due to incorrectly detecting pedestrians that cross the red line (shown in Figure 5.16). These Pedestrians are also identified as blobs, and if these blobs satisfies the kernel shape and size, hence it is miscalculated as a vehicle as they cross the red line.

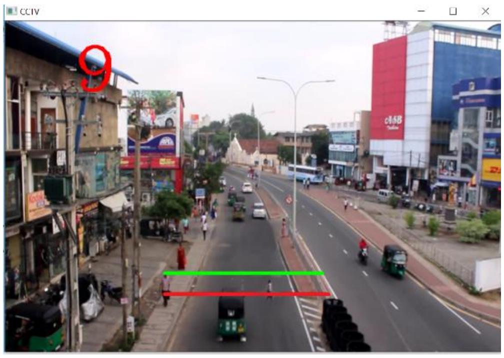


Figure 5.16 - People crossing randomly

Also, it was observed that in video feed 3, the detected vehicle count was less than the actual count. The reason for this was that smaller vehicles, for instance push cycles/ scooters are not detected as vehicles as they do not fit into the kernel size and shape that is not defined. The Figure 5.17 illustrates a scenario where the vehicle 2 is gone undetected.

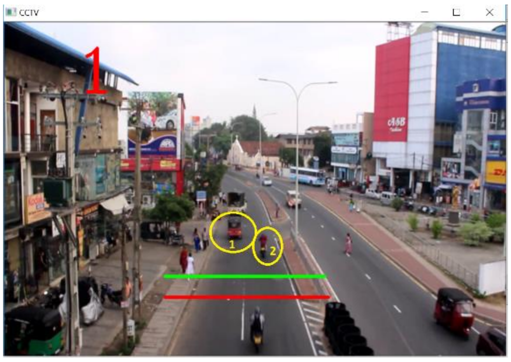


Figure 5.17- Smaller vehicles undetected

5.1.2.3 *Evaluation on vehicle speed*

For this purpose, 2 personal vehicles were driven several times through the traffic route at known fixed speeds. The speed of these were compared with the speed calculated.

The actual speed vs the speed detected by video image processing are depicted in the following graphs, Figure 5.18, Figure 5.19, Figure 5.20 and Figure 5.21

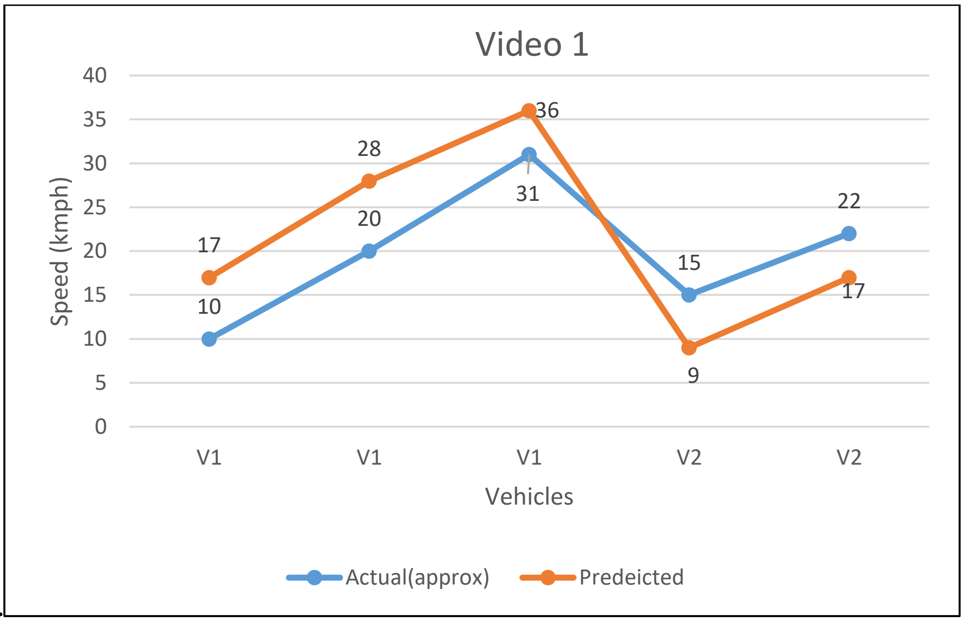


Figure 5.18 - Speeds obtained by video 1

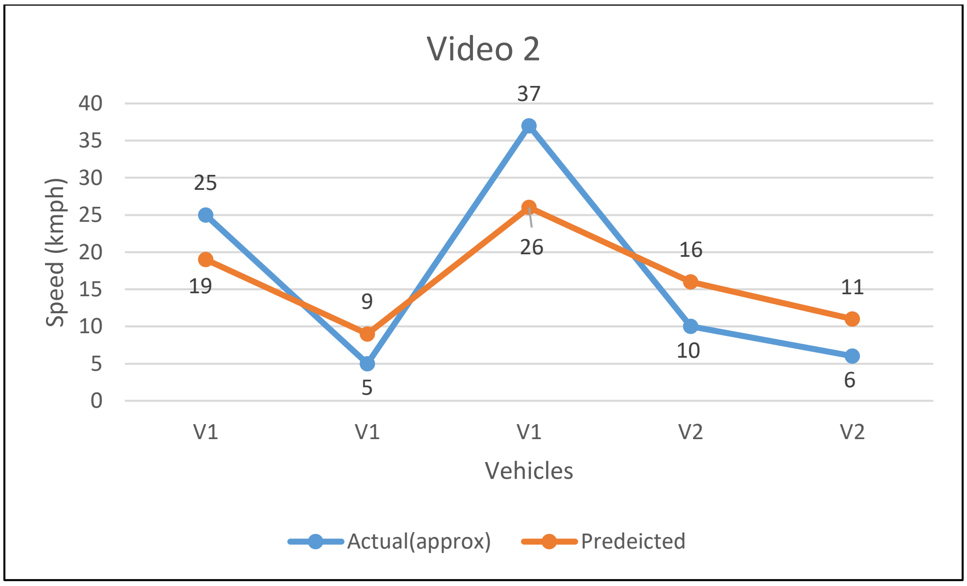


Figure 5.19 - Speed obtained by video 2

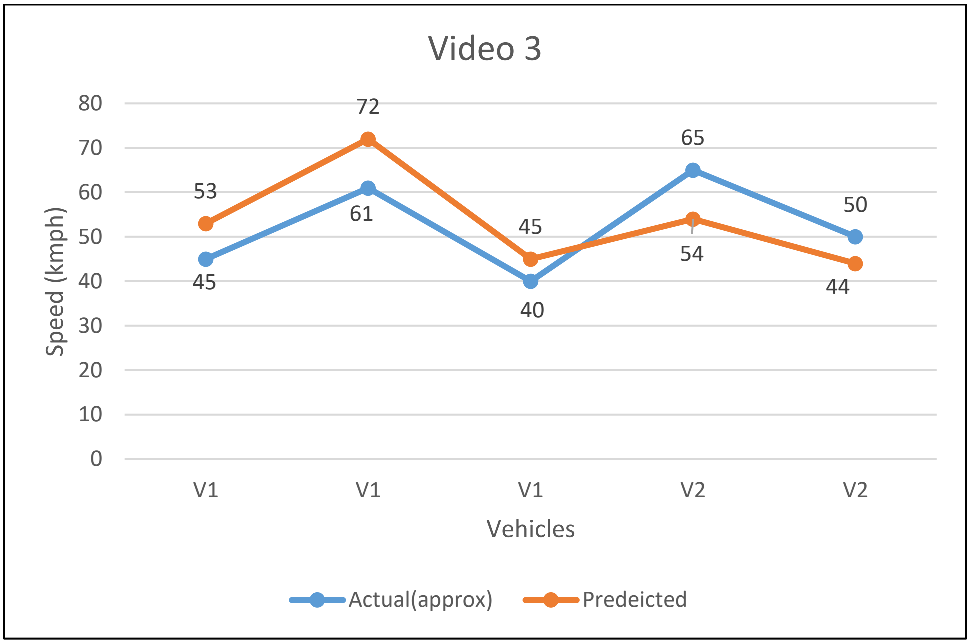


Figure 5.20 -Speed obtained by video 3

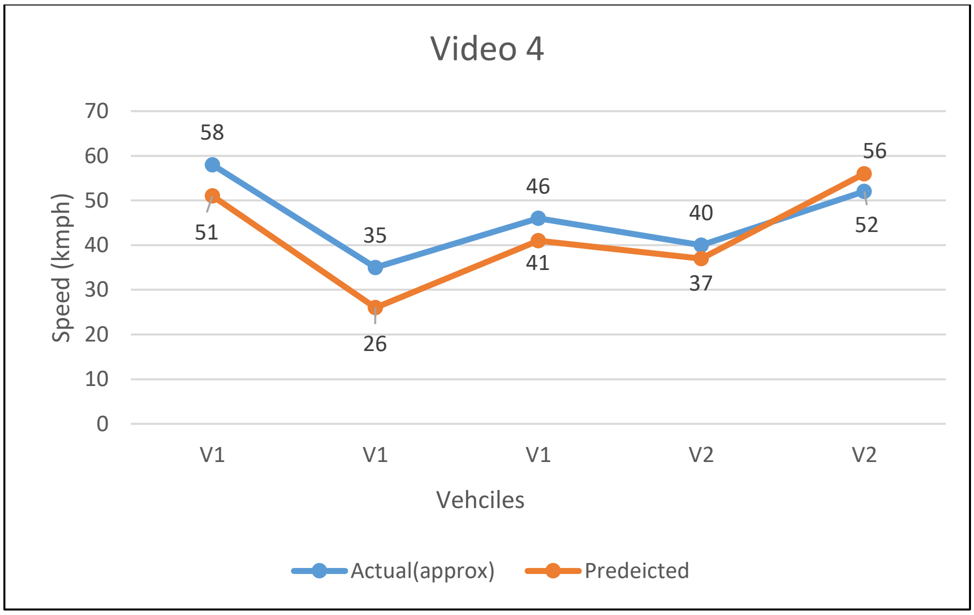


Figure 5.21 - Speed obtained by video 4

The average error in speed obtained by each of the video feeds are shown in Figure 5.22 below.

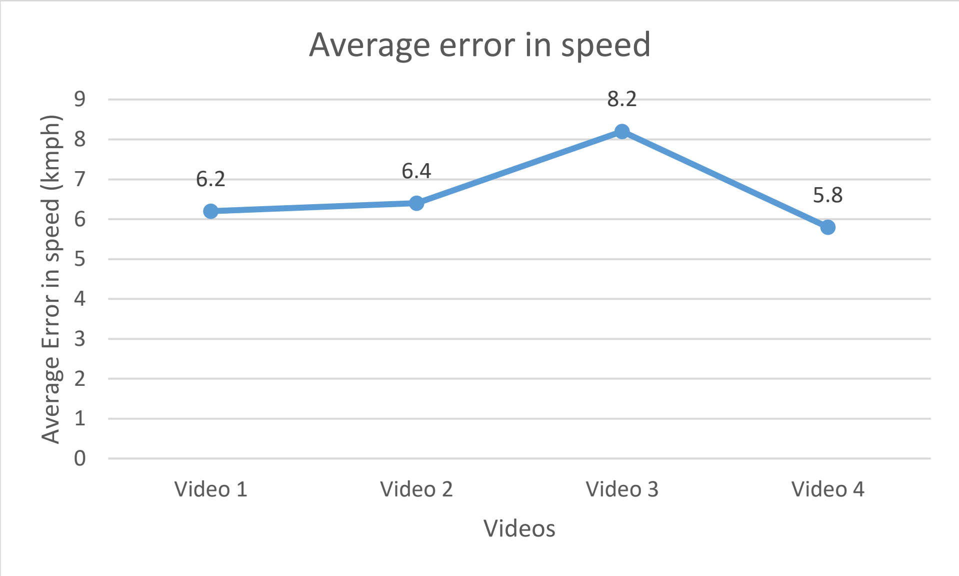


Figure 5.22 - Average error in speed of vehicles detected by video processing

By the above results it was clearly observed that the actual speed vehicles travelled and the speed detected using video image processing has a significant difference. There is an average error in speed is ± 6.65 kmph.

This is due to the angular effect of the location of the video camera, also the time taken to cross the two lines are considered from the video, where as the time taken for the vehicle to actually pass the two lines are different. These factors contributes to the error rate of the vehicle speed.

5.2 Traffic prediction using the current vehicle inflow.

For the purpose of evaluation two videos with duration of one hour from Maradana junction was obtained both on a week day and a weekend, to capture videos of heavy traffic as well as less traffic. The evaluation was done both for the Model and the prediction algorithm.

5.2.1 Evaluating the accuracy of the Model

First and foremost the accuracy of the ResNet-18 model was evaluated using the captured videos. Figure 5.23 shows the how the model is being validated by the obtained video feed. The standard error rate of the ResNet-18 model is roughly 30% which makes the accuracy of it 70%. In this research, the resNet-18 model has been trained with traffic images and the accuracy of the model was 85.78%, which makes the error rate of 14.22%.

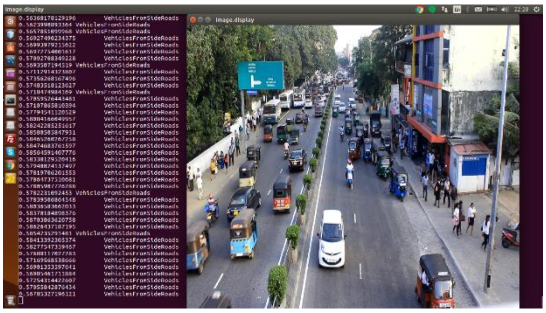
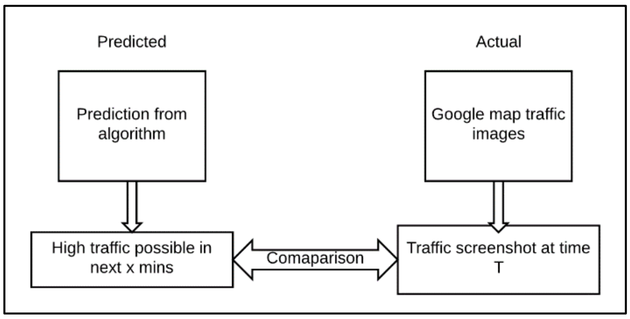


Figure 5.23- Classification of traffic

5.2.2 Evaluation of the prediction

The prediction was evaluated using the traffic information given in google maps. For the purpose of evaluation google traffic screenshots were taken at every 5 minutes, during the same time the video was recorded. The following diagram Figure 5.24 shows how the comparison was done.



Where, x<T<time of the video the prediction is made + x

Figure 5.24- Evaluation approach

As shown by Figure 5.24 the prediction given from the algorithm was compared with the google map traffic images taken for each time slot. The correct google map image was taken by looking at the time stamp of the video at the time of prediction.

For example, as show Figure 5.25, at time of 7.27am the prediction algorithm resulted in predicted traffic being developed during the next 10mins. The google image shows the traffic situation at 7.32am which is between the 10mins time.

Figure 5.25 - Evaluation using google maps

To evaluate the outputs defined in the algorithm the traffic outputs were compared against the google maps traffic colors. Table 5.3 shows the mapping between each output and the corresponding color.

Table 5.3 - Prediction category mapping

Traffic output category given by algorithm	Mapped googled color
1. Very high traffic during next 10mins	Marron
2. High traffic during next 10mins	Red
3. High traffic during next 5mins	Red
4. Medium traffic during next 10mins	Orange
5. Less traffic during next 10mins	Green

With the category mapping as shown in Table 5.3 the evaluation was carried out using the 2 video feeds. With these two videos a total of 50 prediction values were obtained. For each of the traffic records (taken through google maps) the corresponding number of records which the prediction was accurate is shown by Figure 5.26.

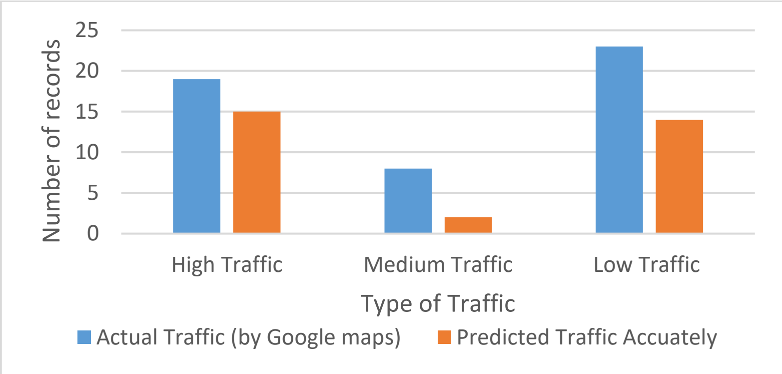


Figure 5.26 - Evaluation results of Traffic prediction

Unfortunately, the time slots chosen for video did not cover very high traffic (google color code Maroon) for evaluation. By looking into the graph if we consider the results of High, Medium and Low traffic individually, there is a lesser gap between the number of times there was actually high traffic vs the number of times the system was correctly able to predict it as high. A similar trend was observed when it comes to Low traffic. However there is a higher gap between the number of times there was actually medium traffic vs the number of times the system was correctly able to predict it as medium.

As of next the accuracy of the overall prediction was calculated and is Figure 5.27.

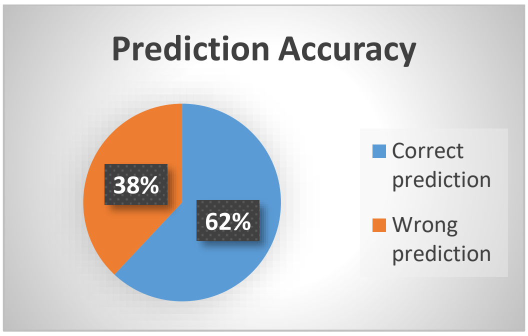


Figure 5.27 - Traffic prediction accuracy

By the results of Figure 5.27 we can see that the prediction is only 62% accurate and has an error rate of 38%. These incorrect detections are mainly due to the incorrect classification of the images through the model. As improvements the model can be further trained with a large number of images and also improvements to the algorithm can be made with continuous training of the system. Also approaches like *Long Short Term Memory* networks (LSTM), Remote Neural monitoring can be used to further improve this.

6 CONCLUSION AND FUTURE WORK

This research discusses three approaches to detect traffic conditions in urban areas of Colombo. One approach was to detect traffic by capturing vehicle speed using ultra sonic sensors. Another approach was to use video processing to capture vehicle speed and count. And machine learning was chosen as the third approach. In addition, a statistical method was used to predict traffic of urban traffic routes of Colombo.

In evaluating the work done using ultra sonic sensors it was found that, the distance between the sensors had to be placed 55.556% from the length of the road to obtain the most accurate speed. In addition, the proposed model cannot detect vehicles travelling in parallel lanes. Moreover with this model it is not possible differentiate a detected object as a vehicle or a person who is crossing the road. Due to these reasons, some vehicles were undetected and the vehicle speed was not very accurate. Hence, it can be concluded that using two ultra sonic sensors placed along the road side will not be suitable to detect vehicle speed in traffic scenarios that occur in urban areas of Colombo. As future work, modifications to the current model can be carried out to support proper detection of speed in the vehicles travelling in parallel lanes.

The proposed image processing method was evaluated using real world traffic videos to verify the suitability of this approach. There were several instances which produced incorrect vehicle count. In some situations, the detected vehicle count was more than the actual vehicle count and vice versa. For instance, a vehicle travelling in the middle of the two lanes was counted as two vehicles. This was due to the centroid of the vehicle falling in the detection range of both the lanes. Another example is that larger and lengthier vehicles were counted as more than one vehicle while smaller vehicles such as push bicycles were not counted at all. The reason behind this was the size and shape of the kernel(convolutional matrix) used in object detection. Further more, people crossing the road were also detected as vehicles. Due to the above reasons the proposed image processing method requires further improvements to gain accurate vehicle count. Therefore as future improvements working on a customized kernel shape and size that can capture people, long vehicles accurately could be carried out.

The speed of vehicles captured through video processing had an average error of ± 6.65 kmph. One reason for this is due to way the video camera is mounted. Also the time difference of a vehicle passing a distance in real life vs how it appears in the video leads to inaccurate speed. Moreover, abnormal speeds were observed due to vehicles changing lanes frequently. Hence it can be stated that the video processing approach used in this research is not suitable to detect vehicle speed and count in urban areas of Colombo.

For the evaluation of the machine learning approach video feeds from the real world were used. The ResNet-18 model used in machine learning has a general error rate of 30%. After training this model using the images needed for the research, the evaluation of the model resulted in a 14.22% error. This error also affects the results of the traffic prediction as the output given by the model was used in the traffic prediction algorithm. However, the accuracy of the prediction was 62%. As future improvements to increase the accuracy of the model, a larger number of images can be trained on this model. Considering the prediction, the usage of recurrent neural network (RNN) and Long short-term memory (LSTM) can be studied to provide better and accurate traffic predictions. These approaches can be used to predict time series of given time lags of unknown size and duration between important events, therefore can result in better traffic prediction.

Bibliography

[1] G. Padmavathi, D. Shanmugapriya and M. Kalaivani, "A Study on Vehicle Detection and Tracking Using Wireless Sensor Networks," *Wireless Sensor Network,* 2010.

[2] T. Kon, "Collision Warning and Avoidance System for Crest Vertical Curves," Blacksburg, Virginia, 1998.

[3] "Detection and Surveillance," in *Freeway Management and Operations Handbook*, p. Section 15.

[4] Enas Odat , Jeff S. Shamma and Christian Claudel, "Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasonic Sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, 2017.

[5] Soobin Jeon, Eil Kwon and Inbum Jung, "Traffic Measurement on Multiple Drive Lanes with Wireless Ultrasonic Sensors," *Sensors (Basel)*, 2014.

[6] A. Festag, A. Hessler, R. Baldessari, L. Le, W. Zhang and D. Westhoff, "Vehicle-to-Vehicle and Road-Side sensor communication for enhanced road safety," in *ITS World Congress and Exhibition*, New york , USA, 2008.

[7] Jun Liu, Hongqiang Lv , Jiuqiang Han and Bing Li, "An Ultrasonic Sensor System Based on a Two-Dimensional State Method for Highway Vehicle Violation Detection Applications," *Sensors*, 2015.

[8] Youngtae Jo, Jinsup Choi and Inbum Jung, "Traffic Information Acquisition System with Ultrasonic," *International Journal of Distributed Sensor Networks,* 2014.

[9] P. Michalopoulos, "Vehicle Detection Video Through IMage Processing: The Autoscope Stytem," *IEEE Transcations on Vehicular Technology,* vol. 40, no. 1, pp. 21-29, 1991.

[10] N. Chintalacheruvu and V. Muthukumar , "Video Based Vehicle Detection and Its Application in Intelligent Transportation Systems," *Journal of Transportation Technologies,* vol. 2, p. 2012.

[11] R. Rahmat and K. Rahmat, "VEHICLE DETECTION USING IMAGE PROCESSING FOR TRAFFIC CONTROL AND SURVEILLANCE SYSTEM," 2017.

[12] A. Lai, G. Fung and N. Yung, "Vehicle Type Classification from VisualBased Dimention Estimation," in *Proceedings fo the IEEE Transportation System Conference , Oakland* , CA, 2001.

[13] P. K. Bhaskar and S. P. Yong, "Image processing based vehicle detection and tracking method," in *International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, 2014.

[14] Naeem Abbas, Muhammad Tayyab and M.Tahir Qadr, "Real Time Traffic Density Count using Image Processing," *International Journal of Computer Applications* , vol. 83, 2013.

[15] Madhavi Arora and V. K. Banga , "Real Time Traffic Light Control System," in *2nd International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2012)*, Singapore, 2012.

[16] Kaiming He , Xiangyu Zhang , Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.