Practical Implementation of Image Processing based Vehicle Speed Monitoring System

A Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Jinna Hrudaya	**17881A04K2**
Rithika Rao. D	**17881A04M4**
Bukka Vinoothna	**17881A04P8**

SUPERVISOR
Dr. D. Krishna
Associate Professor

Department of Electronics and Communication Engineering
VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD
An Autonomous Institute, Affiliated to JNTUH

May, 2021

VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD
An Autonomous Institute, Affiliated to JNTUH

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project titled **Practical Implementation of Image Processing based Vehicle Speed Monitoring System** is carried out by

Jinna Hrudaya	**17881A04K2**
Rithika Rao. D	**17881A04M4**
Bukka Vinoothna	**17881A04P8**

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** during the year 2020-21.

Signature of the Supervisor
Dr. D. Krishna
Associate Professor

Signature of the HOD
Dr. G.A.E. Satish Kumar
Professor and Head, ECE

Kacharam (V), Shamshabad (M), Ranga Reddy (Dist.)–501218, Hyderabad, T.S.
Ph: 08413-253335, 253201, Fax: 08413-253482, www.vardhaman.org

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr. D. Krishna**, Associate Professor and Project Supervisor, Department of Electronics and Communication Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr. G.A.E. Satish Kumar**, the Head of the Department, Department of Electronics and Communication Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Electronics and Communication Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Jinna Hrudaya

Rithika Rao D

Bukka Vinoothna

Abstract

The well-grounded discernment of speed of moving vehicles is considered solution to traffic law implementation in most countries, and is seen by many as a key tool to decrease the number of road accidents and fatalities. Many automatic systems and many methods are engaged in different areas, they tend to be exorbitant and/or labour intensive, frequently employing old-fashioned technology due to the long progress time. Here we report a speed detection system that depends on simple everyday equipment - a laptop and a web camera. The proposed system relies on tracking the cars, which gives the track of the car using respective algorithm with the help of recorded video stream. In the thesis part, the proposed system deals with the technology such as Adaptive background subtraction based on the method Gaussian Mixture Model, and DBSCAN, the clustering algorithm for forming the clusters and the Kalman filter for tracking the selected automobile. Consequently, our system evaluates the actual speed of moving vehicles precisely. The output of the system shows promising results on videos obtained in a lot of scenes and with different templates.

Keywords: DBSCAN; Gaussian Mixture Model; Kalman Filter

Table of Contents

List of Figures

Abbreviations

Abbreviation	**Description**
GMM	Gaussian Mixture Model
RADAR	Radio Detection and Ranging
LADAR	Laser Detection and Ranging
EM	Expectation Maximisation
LQE	Linear Quadratic Equation
SIFT	Scale Invariant Feature Transformation
SURF	Speeded Up Robust Features
HOG	Histogram of Gradients
FOV	Field of View
KDE	Kernel Density Equation
ISDN	Integrated Services Digital Network
DBSCAN	Density Based Spatial Clustering of Application with Noise

CHAPTER 1

Introduction

We all know that over speed is the prime cause for road injuries. In this busy life schedule, everyone would go for driving at very speed rather than low speed to set foot on their respective locations punctually. Thus, it is mandatory to understand the demand of a monitoring system which would be used as a speed limit imposition system.

In the present system, to detect rash driving, police use a handheld radar gun and aims at the vehicle to record its speed. If the speed of the vehicle exceeds the allowable speed limit, the nearest police station is informed to stop the speeding vehicle. This process is more time consuming and as compared to the continuous increase of traffic this system cannot be trusted with the lives of people. Our proposed project aims to develop a wireless system that detects cars driving at speeds over a specified limit and inform concerned authorities immediately. This system does not need any human interception and a lot of time is saved effectively. The time required for a particular car for moving from one point to other is first calculated on the basis of the time required the speed of the car is determined. This data is then transmitted to the concerned police authorities at a remote location wirelessly.

A system which assists to restrict the speed of the vehicles and the person driving would be penalized under the law due to over speed and this is the best way of making everyone to drive at standard allotted speeds. In some places, traffic policemen monitor to appropriate functioning of moving vehicles on roads and at some places, traffic places RADAR system is used and this is a technology which is based on the Doppler Effect and with just a trigger it can discern the speed of the vehicles. Every system if it is a RADAR or any other it manually demands a human to take for monitoring the vehicles' speed and to notify if any vehicle breaks the law or over speeds. Figure 1.1 depicts how the accidents take place due to over speed.

Figure 1.1: Accidents due to over speed

Every system uses a modern and advanced technology to prioritize the automation over human handled machines. So, the speed monitoring system should also be made as automatic which could be done in many ways. This paper is an idea of one of such system. The project is evolved by keeping in view all the disadvantages mentioned above and is named as vehicle speed monitoring system based on image processing. This system mainly focuses on calculating the speed of approaching vehicle that over speeds. Figure 1.2 shows the statistical representation of the road accidents taking place.

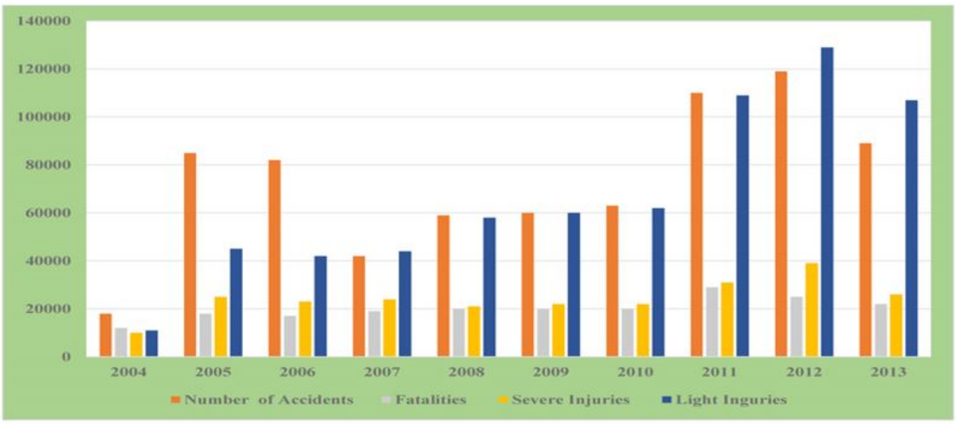

Figure 1.2: Statistical Representation of road accidents

Rash driving is the reason for some street mishaps everywhere on the world. The traffic populance has expanded extensively and there is no viable way to control or screen the speed of vehicles running on streets. This paper aims at monitoring and detecting the speed of vehicles. The main motto of the project is to find the owners of over speed vehicles guilty as they lead to major accidents. The method proposed in this research focuses on both hardware and software and is one of the efficient and easiest ways to find the velocity instead of using radar equipment. The applications of the proposed work are highway traffic surveillance control, dynamic traffic management and variable speed limits to control the over speed vehicles in order to avoid the accidents.

The proposed system to determine the speed using techniques such as Gaussian Mixture Model, DBSCAN, Kalman Filter techniques. Firstly, the input is taken in the form of video which is captured using camera pointing up towards the road. The technique referred in the proposed system is based on an adaptive background subtraction method known as the Gaussian mixture model. After the classification of each pixel using the background subtraction method, segments of foreground points are characterised by clusters produced using the DBSCAN (Density based spatial clustering of applications with noise) clustering technique, and these segments are denoted by Bounding boxes.

Lot of ways are being executed to examine and recognize the over speeding vehicle. But no self-activating system has been implemented so far that can accomplish the task of speed detection and vehicle identification in the absence of human assistance.

Some of those methods are RADAR and LADAR. The speed detection system RADAR (Radio Detection And Ranging) works on the principle of doppler effect. This is employed to know the frequency shift in the reflected wave. The greater the Doppler shift the more will be the speed. The devices for speed detection in RADAR are S-band, X-band etc working in their respective frequencies. The disadvantages of this system are:

- Time - Radar can take up to 2 seconds to lock on .
- Radar has wide beam spread (50 ft diameter over 200 ft range).
- Cannot track if deceleration is greater than one mph/second.
- Large targets close to radar can saturate receiver.
- Hand-held modulation can falsify readings.
- More interference sources.

On the other hand, LADAR system (Laser Detection Ranging), LADAR uses 3 semiconductor diodes to generate laser light. Uses light pulses to make 2 consecutive distance measurements, then divides by time. Lenses are used to collimate light to narrow beam. The disadvantages of this system are:

In order to overcome these disadvantages we proposed a simpler method based on Image Processing. The main objectives of the proposed system is To design an automatic and versatile system that addresses the issues of over speeding without use of human resources using image processing and to detect the speed of the over speeding vehicle and display their respective speeds.

CHAPTER 2

Literature Survey

In [1], the researchers have proposed a system to determine the speed using techniques such as Gaussian Mixture Model, DBSCAN, Kalman Filter techniques. Firstly, the input is taken in the form of video which is captured using camera pointing up towards the road. The technique referred in the proposed system is based on an adaptive background subtraction method known as the Gaussian mixture model. After the classification of each pixel using the background subtraction method, segments of foreground points are characterised by clusters produced using the DBSCAN (Density based spatial clustering of applications with noise) clustering technique, and these segments are denoted by Bounding boxes. GMM employs Expectation Maximization (EM) to improve the probability of observed data, however it would be expensive to code it for each pixel, thus EM is substituted with Online K - means approximation. The clustering concept based on density that is intended to identify clusters of any shape. The Lukas-Kanade algorithm is used in our suggested technique. Kalman filtering, also called as linear quadratic estimation (LQE), is an approach that consist of a series of measurements taken over time that contain statistical noise and other inaccuracies that provide values of unknown variables that are more accurate than those based on a single observation alone. Kalman filter using Optical flow measures each move of pixels per second, which also represents vehicle motion. The weight of the pixel is required for the procedure to determine speed. The proposed system can be developed to identify speed from vertical movement using adaptive pixel weights and to enhance DBSCAN segmentation to separate each object in a cluster of vehicles.

The proposed work in [2] mainly focuses on detecting the speed of the vehicles by using stationary cameras which are fit over a road. The following steps are included in order to measure the speed. Several background and foreground extraction techniques exist in object detection, such as the use of Gaussian distribution, adaptive median, morphological background estimation, and so on, but in terms of difficulty and efficiency, adaptive background subtraction is implemented, where the background is removed from the foreground. Shadow removal, convex hull, morphological procedures, sketching a bounding box around the identified objects, calculating its centroid, and maintaining track of the objects in each frame of the video are all part of the object tracking process. Speed calculation is the final process in which the ultimate speed in kilometers per hour is computed A perfectly functioning algorithm is created, and the results are displayed. Normal background subtraction involves the usage of a fixed picture known as the background picture. Each frame of the video would then be removed from this background picture, and the resultant picture is then thresholded to produce the final result as a binary picture. Native background subtraction seems to have its own plethora of challenges. If a stationary item which has been existing in the background image for a lengthy time goes out from the frame, it leaves a hole. Furthermore, because the backdrop picture is fixed, any variations in the intensity of light will be identified as spurious objects. The adaptive background subtraction technique is similar to the native background subtraction technique, other than that the background picture continuously updating itself in the event of any changes. Monitoring the identified items frame by frame is a difficult process since the system will not be able to derive continuous historical information about the image. Simultaneously, poor foreground recognition owing to shadows and noise makes tracking more challenging. While adaptive background subtraction addresses any changes in lighting, we also need to eliminate shadows to minimise false detection. The apparent way is edge detection, but we adopt a simpler method of comparing the pixel values in the current frame with the backdrop pixel, which has shown relevant to a wide range of shadows. The convex hull will be the first of multiple morphological procedures performed on the binary picture following shadow removal. Any two white pixels within the object are connected by a line in convex hull, and all points on the line are turned white. This is a hole filling approach that aids in the conversion of practically all linked objects to blobs. In a similar fashion, morphological procedures such as erosion, dilation, closure, opening, and hole filling are conducted, and the ultimate result of the picture with cars as correct blobs is accomplished with all noise eliminated. The very next step is to maintain track of the things that have been identified. Each identified vehicle (object) is issued a unique label, which is kept from the time the thing enters the scene until it exits. The bounding box is then drawn around each item, and the middle of the box is determined and recorded in an array as the item's centroid. These centroids serve as a basis for an object's position and aid in speed estimation. The speed of the observed blobs is computed and displayed on the screen.

The suggested work in [3] is primarily concerned with detecting the speed of the vehicles involves the following techniques. The current study seeks to develop an automatic vehicle counting system that can process videos recorded from fixed cameras installed over roads, such as CCTV cameras installed at traffic signals / crossings, and count the number of vehicles passing a spot in a specific time for further vehicle data collection. To solve the problem, a basic strategy was used, which included Gaussian mixture model-based item recognition, non-predictive regional tracking, and counting of tracked items based on basic criteria. The presented automatic vehicle counting system uses video data received from fixed traffic cameras to estimate the number of cars present in a scene by conducting causal mathematical operations on a collection of frames collected from the video. In each frame, the Gaussian mixture model distinguishes the item in motion from the background by monitoring identified items within a defined zone and then counting. A Gaussian Mixture Model (GMM) is a function that represents adaptive probability density as a weighted average of Gaussian component densities. GMMs are widely employed in a wide range of applications. GMMs are used in a variety of domains, including machine learning, astrophysics, computational biochemistry, and many more. GMM performs the task of differentiating the foreground and background from picture frames in this application by learning the backdrop of a scene. The goal of this study is to address the need for a strong vehicle recognition and system performance that may be used in a traffic monitoring system to address the present issues in this field. To begin, the Gaussian mixture model employs a common observable property change factor between the current picture and the reference picture, i.e. modelled background, for all pixels in a group of pixels to cope with changes in picture frames and automatic gain by the camera. A threshold value is generated to assess the resemblance of the objective description of the quality of a colour independently of its brightness between the GMM-learned background and the present observed picture, which is a pixel in the foreground acquired by the GMM. Blob detection is a method that allows a system to track the movement of items inside a frame. A blob is a collection of pixels that represents an item. This detecting technique locates the blob in subsequent picture frames. Before any blob detection, the blob region must be created, where pixels with comparable light or colour values are clustered together to discover the blob. In a real-world scenario, every surface has tiny changes, therefore if only one light or colour value is chosen, a blob may consist of only a few pixels. When attempting to arrange photos into usable components, it is possible that the resulting unit will be useless as a whole. Blob detection in this study makes advantage of contrast in a binary picture to compute a detected region, its centroid, and the area of the blob. The pixels identified as foreground are supplied by the GMM. A contour detection method is used to group these pixels together in the current frame. The contour detection technique divides the individual pixels into separate classes and then locates the contours that surround each class. Each class has been designated as a potential blob (CB). These CB are then sized, and tiny blobs are excluded from the algorithm to eliminate false detections. The locations of the CB in the current frame are compared using k-Means clustering, which identifies the center of clusters and groupings the input samples CB around the clusters to identify the cars in each region. When a moving vehicle reaches the base line, it is tallied. The frame is captured when the vehicle travels through that location. The blobs with the same label are evaluated in each location, and the vehicle count is increased. Blob analysis locates probable items and places a box around them. It determines the area of the blob, and the centroid of the object may be retrieved from the rectangle fit around each blob for tracking purposes. An additional constraint that the ratio of blob area to rectangle area around a blob should be larger than 0.4 guarantees that no unneeded items are recognised. To avoid excessive computing duplication and improve efficiency, tracking is performed exclusively inside a defined section of the frame known as the Count Box. Tracking is accomplished by looking for centroids in a small rectangular region surrounding centroids identified in the previous frame; if none are found, the item is added to a tracks array as a newly discovered item. The experimental outcome comprises of comparing the automatic car counting from videos to the researcher's manual counting.

The different stages are used in the proposed study in [4], which is largely concerned with detecting the speed of the vehicles. It mainly focuses on single and multi camera approach. Appearance-based methods and Motion-based procedures are the two models for vehicle detection. Appearance-based techniques detect vehicles based on features such as form, colour shading, and size. Models based on motion identify an item in relation to its surroundings. Appearance-based techniques make use of existing information to partition the foreground and static backdrop. The appearance-based technique might be feature-based or part-based. Physical aspects of the vehicle are used in feature-based models. The low level component technique makes use of low level properties such as shade, shadow, symmetry, and edge. String descriptors have mostly taken the place of low-level features. Several techniques to component descriptors have been developed, including Scale Invariant Feature Transformation (SIFT), Speeded Up Robust Features (SURF), Histogram of Gradients (HOG), and Haar like features. The vehicle is subdivided into smaller sections in a part-based model, and the spatial relationships between them are represented. Part-based models separate the vehicle into front, side, and rear sections to improve occlusion detection and camera FOV edge detection. Motion-based approaches separate moving vehicles from the backdrop based on motion. Periodic frame differencing and back ground removal are two motion-based approaches. The metric model updates the parameter estimates by applying a uni-model probability to each pixel. Non parametric background models employ pixel history to build a similarity measure of the data using previous samples of pixel values but do not view pixel value as a specific distribution. Non-parametric models include Kernel Density Estimation (KDE) and codebook. The identified foreground might match to a variety of things in the image. It is necessary to separate and identify the item of investigation. The number of classes is determined on the characteristics and categorization employed. Vehicle categorization entails extracting vehicle static data such as colour, number plate and logo. Visual cues such as edge, gradient, and corner are used in appearance-based approaches. This method can differentiate a wide range of vehicles with different time complexity and accuracy depending on the characteristic used. K means distinguishes five types of cars based on edge properties. Vehicle tracking is used to get the vehicle path by identifying motion dynamic properties and features in order to anticipate its position in consecutive frames. Vehicle tracking may be divided into three types: motion-based tracking, region-based tracking, and feature-based tracking. Multi-view vision, which employs two or more cameras, makes vehicle monitoring less time-consuming and more efficient. There are several algorithms, but none of them are completely suited to all applications. This paper contains a wealth of information on video-based vehicle detection, identification, and tracking, which will be useful to future researchers in the field.

The developed work in [5] is primarily concerned with detecting the speed of the vehicles consists of the following techniques. In spatial databases with noise, density-based approaches were presented to find the arbitrarily shaped cluster. Density-based clustering algorithms such as DBSCAN, DENCLUE, and OPTICS are widely employed. COBWEB and SOM are examples of model-based clustering techniques. These algorithms seek the greatest match between a given set of data and a mathematical model. Constraint-based clustering algorithms identify clusters that meet the user-specified choice or constraint. Algorithms suggested in each of these areas attempt to solve clustering issues involving enormous amounts of data in a huge database. The density-based clustering techniques are effective for discovering clusters in datasets of variable form and size. These methods often cluster as dense clusters of points in the data space divided by low density regions. DBSCAN would be the first density-based clustering method. It builds clusters based on a density-based connectivity analysis. The researchers investigated the various DBSCAN upgrades presented thus far in this work. These are offered as modifications and enhancements to the DBSCAN in order to perform successful clustering analysis on the underlying datasets. In this study, these are examined along with their advantages and disadvantages. Furthermore, a critical evaluation of these research works is presented in relation to the various parameters. DBSCAN looks for clusters by inspecting the surroundings of each item in the dataset. If the neighbourhood of an object p includes more than Min Pts, a new cluster is formed with p as the center object. It then gathers directly density-reachable items from these core objects iteratively, which may include the merger of a new density-reachable cluster. In this paper, they offered an overview of the different enhancements to the density-based clustering technique known as the DBSCAN. The goal of these adjustments is to improve DBSCAN's ability to provide effective clustering results from the supporting datasets. Furthermore, they have acknowledged the research contributions and identified certain flaws in various study studies. As a result, this work also portrays a critical assessment in which comparison and contrast have been used to demonstrate the similarities and differences between the works of various authors.

The researchers suggested in [6] a methodology for determining speed of the vehicle using the following strategies. In the proposed work, the system enables fully automated tracking and monitoring of the vehicle, which is beneficial to school buses, their owners, and children's safety, as well as providing a precise time of arrival of the vehicle at a certain place or station. As a result of the precision in time, children may spend more time studying, napping, or resting instead of waiting for a late bus. Wasting time waiting for a bus increases the student's comfortable and productive time management. The suggested system collects vehicle tracking data such as vehicle number (Unique ID), position, speed, Date, and Time and stores it in the Raspberry Pi database.

With the use of a temperature sensor and a gas leakage sensor, the system also offers students with a protective measure. The MySQL database system, which is a sophisticated feature of the Raspberry-Pi, is used to monitor the car using GPS and manage its information. The GSM/GPRS function is able in the database base monitoring and updating mechanism, which transmits the updated vehicle database to the server and allows users to access the database through a web page on their Smartphone. This displays the vehicle's current position in real time on the Smartphone. As a result, users will be able to use their Smartphone to continually monitor a moving car on demand and determine the expected distance and time for the vehicle to arrive at a specified place. The suggested system would be controlled by a Raspberry Pi installed inside the car. The GPS GPRS GSM SIM908 module would connect with the Raspberry Pi board through the USB port. The longitudes and latitudes of the present course obtained from GPS of the GPS GPRS GSM SIM908 module are verified to the longitudes and latitudes saved in the specific file format within the raspberry pi data. If the longitudes and latitudes don't really match those stored, an incorrect route detection alert message will be issued to the vehicle's owner's mobile phone. Additionally, the longitudes and latitudes of the present course acquired via GPS will be relayed to the server through GPRS, allowing the vehicle's present position to be tracked on a web page using a Smartphone. The designed service includes a login facility on a web page for the car's owner, students, and their parents in order to track the car. The DS18B20 temperature sensor and the MQ6 gas leakage sensor are also used to ensure the safety of the students. These sensors communicate with the Raspberry Pi. If somehow the temperature exceeds a certain threshold or if LPG gas leaks inside the car, an alarm message is delivered to the owner. Similarly, the system's built-in safety mechanism. Using sensors, the suggested system plays an essential role in real-time tracking and monitoring of the vehicle, as well as providing a safe and secure solution to the passenger. Whenever a car theft or accident happens, the proposed system sends the car's current position and speed to the car owner's cell phone. As a result, it is advantageous to monitor the car as soon as feasible. In some cases, such as when a student's safety is at stake, the proposed system includes a provision for sending an alarm message to the student's parent's mobile phone, which also plays a vital function.

The authors proposed an approach for assessing vehicle speed utilising the following procedures in [7]. The Mobile Bug Module sends a trigger pulse to the GSM Modem. It sends signals to the police station in ability to track calls. It is controlled by a micro controller through an RS-232 interface. Speed Sensors monitor the vehicle's speed and trigger the GSM Modem when it exceeds 40 kilometers an hour. The GSM Modem is set to convey signals once the average speed exceeds 40km/hr. The sensor detects phone calls and texts and delivers them as a pulse to the micro controller. The pulse will be sent to the GSM Modem as data. If the person driving the car receives a call or a message while driving, the LED illuminates and their unique ID is communicated to policemen via the GSM Modem, and the cops central controller has a GSM receiver, the output of which is supplied to another LED. The GPS module determines the vehicle's geographical location. This aids in determining our system's destination and acceleration. GSM Modem receives module data output such as global positioning system fixed data and geographic position-latitude. The planned work monitors the driver to see if he or she is receiving or making a call and stops them from diverting their attention from the road, so averting many accidents caused by the use of mobile phones while driving. A component of our system known as the mobile bug monitors calls and texts within a 1.5m radius using voltage variations in the signal. Thus, whenever a driver exceeds 40km/h, this gadget activates the GSM Modem and sends the message to the authorities. GSM Modem [7] delivers the message "THE DRIVER IS USING HIS MOBILE WHILE DRIVING" along with the "VEHICLE NUMBER PLATE" to the officers' control room using AT instructions +CMGS as AT commands +CMGS may be used to transmit SMS messages from a computer / PC AT+CMGS="91234567"CR¿ It is simple to send text messages. Ctrl+z¿ is a shortcut for cutting and pasting. The GSM modem is programmed to control that is linked to an RS-232 port. MAX-232 is used to connect RS-232 with a microcontroller. Every

GPS satellite sends out radio signals that allow GPS receivers to compute its (or the vehicle's) geolocation on the Earth and transfer the results into geodetic latitude, longitude, and velocity. To establish your vehicle's location, a receiver requires signals from at least three GPS satellites. GPS receivers, which are often used in autonomous vehicles, can only obtain information from GPS satellites. They are unable to interact with GPS or any other satellite. A GPS-based system can only compute its location and cannot transmit it to a central control room. To do this, they often employ GSM-GPRS Cellular network connections through an extra GSM modem/module. The circuit here known as the "Mobile Bug" is used to detect both arriving and departing calls/messages inside one radius of around 0.5-0.75m. The vehicle's speed sensor, which is situated on the transmission's output shaft (on certain older versions with mechanical speedometers, the speed sensor is positioned behind its speedometer), delivers electrical pulses to the microcontroller, which are created by a magnet spinning a sensor coil. So when driver uses his/her cellphone while driving, the policemen will be notified by a message stating that the driver is using his/her mobile phone while driving, as well as the location of their car and the registration plate data, which are already saved in the microcontroller's memory. As a result, lawbreakers can be quickly apprehended using this approach. As a result, this approach would serve as a model for policemen to provide an accurate and effective reaction when drivers report the usage of a mobile device while driving. It might also be retrofitted onto current cars, however this would need some modifications.

In [8], the researchers suggested a technique for measuring vehicle speed based on the following steps. This paper developed a vehicle video surveillance system based on differential GPS location and China Unicom's 3G connection technologies. The vehicle tracking system, which is focused on GPS location and 3G communication technology and is placed in the bus to analyze the interior and exterior of the automobile in real-time, is made up of a vehicle terminal host, a car camera, and a wireless network subsequently spread. The Monitoring Direction Center is designed to examine the data that comes directly ahead, as well as the network architecture. Enable video cameras at critical points of Vehicle Import and Export to capture video signals and transfer them to the vehicle endpoints host; the module is in charge of collecting data and compiling compression of video/audio, and finally storing the data in the vehicle endpoints' memory card. The host of the automotive endpoints can interface with the major location to post video data and warning signals in and out of the car using 3G wireless networks (China Unicom WCDMA). The GPS satellite is taken to generate location data and to interface with voice messages. The vehicle endpoint host can acquire GPS satellite location coordinates and transfer it to the central platform through 3G wireless networks (China Unicom WCDMA); the monitor of the central platform displays the vehicle location, speed, direction, angle, and other information; and it can perform car phone operations at the same time. They created a vehicle monitoring system based on GPS location and 3G communications based on the system architecture and core technologies mentioned above. In the system, they effectively employed China Unicom's public communication WCDMA network as the mobile communication connection and established a vehicle Private data network on it. In order to avoid interference from the car's radio and electromagnetic fields, we use a digital technique for video transmission via a wireless digital network monitoring terminal. Differential GPS technology, 3G wireless communication of WCDMA, flow management of data transfer, and sophisticated wireless connection were all used extensively in the creation of a vehicle tracking system based on GPS position and 3G communication.

The authors proposed a system for assessing vehicle speed based on the following methods in [9]. This technology continually calculates the speed and GPS coordinates, and these GPS coordinates aid in determining the region in which the vehicle has been located as well as the maximum speed permitted in that region. The determined speed and coordinates of the vehicle are continually saved on a memory card. If a vehicle's speed exceeds the posted limit, the driver is notified by a buzzer. If the motorist continues to violate the speed limit, an SMS with the car number plate and GPS coordinates of where he violated the average speed is sent to traffic regulators. As a result, a speeding ticket might be issued against the same car. The proposed methodology comprises of an Arduino Mega-based controller that monitors the positioning and operating speed using a GPRS+GPS Quadband Module (SIM908), GSM antenna, GPS antenna, and SIM card. When the vehicle is detected in any of the designated locations, the controller compares the velocity to the maximum permissible speed in that location. If over speeding is identified, an active buzzer employed in this system generates a buzzer sound to inform the motorist that he is violating the maximum speed. The Arduino Mega 2560 is the most popular ATmega2560-based microcontroller board. It has a crystal oscillator with a frequency of 16 MHz. 16 pins are employed for analogue inputs, while 54 pins can be employed for digital input and output. It is a low-power module designed to be deployed in four bands, 850/900/1800/1900MHz, integrating Global Positioning System (GPS) and Global System for Mobile Communications (GSM) technologies, allowing real-time tracking in GPS-enabled technologies with signal coverage. This antenna is necessary for message transmission and reception. It is also required for receiving phone calls. The GSM antenna has a frequency range of 850/1900 MHz. It has a VSWR range of 0 to 1.5 and a gain of 3.5 dBi. This can tolerate a maximum input power of 60W and has an input impedance of 50 ohms. To gain access to the SIM908 features, they employed the proper Attention (AT) instructions. When the SIM908 module is turned on, the GPS component is turned off. They give an order to turn it on. They reset the GPS component of the module in autonomous mode before receiving position information. This system, which is made using an Arduino Mega microcontroller and a sim908 module, may be replaced with a single module, such as the "Linkit One," which has GPS, GSM, and Wi-Fi capabilities, allowing the system to be smaller and more portable. However, this technique raises the module's cost. This system may be enhanced with a feature that allows GPS coordinates to be stored in the cloud. This is useful for tracking automobiles in real time. Real-time tracking is applied by law enforcement and the owners of transportation businesses that convey precious goods. This may easily be expanded so that the user may determine the position of the vehicle by contacting the sim card that has been inserted, and the system must be set to send the Location data to the owner's mobile phone. This really is useful for the driver if he is in a new place and forgets where he parked his car. In addition, if the automobile is stolen, it may be traced by continually obtaining GPS locations.

In [10], the investigators suggested a system for evaluating vehicle speed based on the following approaches. The technology is intended for vehicle testing and will aid research and development teams in the car industry in the performance testing of automobiles. Prototype automobiles must pass many tests, including indoor and outdoor testing, in accordance with automotive regulations. Such tests are critical for continuous vehicle improvement and design changes. The suggested technology will be employed for car testing outside. The designed system is primarily comprised of a microcontroller and a GPS + GPRS module. The controller receives vehicle characteristics and GPS coordinates from the GPS module. Using GSM/GPRS technology, the controller sends this data to the server. Along with the Arduino controller, the Sim808 module is designed for GPS and GSM/GPRS communication. Vehicle metrics that are often monitored include vehicle location, vehicle speed, engine compartment temperature, fuel level, and so on. Those characteristics are saved in a database on a web server, and a site is developed to show the data from the car parameters. Car position data is integrated with Geolocation at the same time to showcase vehicle actual location. Arduino is open microcontroller platform with many board versions based on the circuit board utilized. ATMEGA series 8 bit controller chips are commonly used. Certain variations use 32-bit ARM controllers. In this experiment, an Arduino MEGA 2560 controller board with an ATMEGA2560 (8 bit) controller chip is employed. A GPS solution provides the finest acquiring and monitoring capability in the industry. SIM808 is built with power-saving techniques in mind, with current usage as low as 1mA in sleep mode. SIM808 incorporates the TCP/IP protocol as well as enhanced TCP/IP AT instructions, which are quite helpful for data transmission. The technology would be used to observe or control the position and vehicle attributes of various test vehicles from a single location for research and development purposes, as well as to save data of those cars' design parameters also on server for subsequent analysis and record keeping. This project's usage of IoT and an open source platform makes it incredibly dynamic, efficient, and cost effective; as a result, it may be used to extend the monitoring of many vehicle characteristics.

The authors presented a procedure for analyzing speed of the vehicle in [11] based on various methods. The creation of a one-of-a-kind comprehensive method for determining the best solution for whole vehicle maintenance, including tracking fuel use, tracing non-economic activities, and harm to automobile components and efficiency. Recording and applying basic technologies, such as Microcontrollers, that can be connected with a wide range of cloud and virtual storage devices that are not only safe, but also flexible and give real-time monitoring. The technique must be user-friendly and simple to implement so that the average person does not see it as a bother but as a symbol of satisfaction to get it. The following describes how a vehicle monitoring system works. When the rider conceals their face and attempts to start the vehicle, they receive a failure answer since the Recognition module function has been prohibited by the rider. The rider is then approved by the Authentication module by using his/her fingerprint to start the engine, and the fuel status is communicated to the customer via the Customer communication module providing both voice and images, after which the rider may begin his journey. The Display module uses GPS to project the most cost-effective and efficient route. Using monitoring system technology, this study provides solutions to challenges encountered by vehicle passengers. The collected real-time data provides passengers with the necessary knowledge to improve the use of their car, lowering transportation costs. The section clearly also assist manufacturers in providing value-added services to their customers. According to the poll findings, installing a tracking system in a bike has a lot greater scope than installing one in a car. However, adopting in bikes introduces new challenges that must be well investigated.

In [12], the study defined a strategy for evaluating vehicle speed based on several approaches. They created and implemented a digital automobile monitoring system for public safety in this article. The system is made up of a central server, a vehicle unit, and a wireless communication network. This technology is intended to communicate video, audio, and other information acquired by car tracking terminals through 3G networks to the police station's tracking and control room. The suggested technology may be used by police agencies to broaden their perspective and increases the performance and accuracy of making decisions. The report describes the major technologies used in the creation of a digital vehicle monitoring system for public safety. The Visual C++ software platform is used to create the digital vehicle monitoring system for public safety. Oracle 10g relational database administration is used to establish background databases, as well as to store and manage various types of data. The suggested monitoring and command facility in this article is housed in a police station. Police stations can set up subcommand facilities that are linked to the monitoring and command facilities through DDN special lines. With swift advancement of information technology, wireless, and networking, the usage of a digital vehicle monitoring system has substantial public security significance. The design and functionality of a digital vehicle monitoring system based on 3G for public security are proposed in this study. This system creates a safe police vehicle network platform that permits real-time video, voice, and data capture from a vehicle and transmission to a monitoring and command centre. Officers in the control room have superior tactical awareness and can make better judgments faster because they have access to sophisticated multimedia apps. This technology takes full account of the privacy of the video collected, resulting in compelling evidence for conviction or acquittal. The technology suggested in this study can also be used for vehicle monitoring and tracking.

The study established a system for measuring vehicle speed in [13], which was based on many methodologies. The system is comprised of both local and central components. On-site equipment such as a Doppler radar, a strobe light, a camera, and a camera controller are mounted. The location is frequently chosen from the standpoint of a traffic accident situation. The Doppler radar unit, camera unit, and strobe flash unit are often located over each lane on a highly complex such as a barrier or tower. The camera processor is placed close to the attachment framework. ISDN (Integrated Services Digital Network) is a telecommunications network that connects local and central equipment such as computer, a hard copy, an image display, an optical drive, and so on. The Doppler radar monitors the speed of a moving automobile. When an automobile exceeds the predetermined velocity value, a camera with a fast shutter captures the front image of the vehicle. The images are automatically communicated over ISDN to the central hardware and stored on an optical disc. Police officers then examine the photographs on the picture display. They are also published on the internet using a hard copy machine. The captured photographs contain data such as the running velocity, date, time, speed limit, camera position, and so on. Even in adverse weather conditions, the photos are very clear. For speeding auto management, traditional speed checkers with still cameras have been employed. Furthermore, these technologies need the use of police personnel, such as film setup and processing. The Radar Speed Tracking System relieves police personnel of these duties. The system's benefits include reduced traffic accidents, reduced traffic police staffing, and reduced traffic noise. The systems have been installed by the National Police Agency of Japan since 1991 as part of the 5th Five Year Plan for Traffic Safety Global Wellness.

In [14], the research devised a methodology for assessing vehicle speed that was based on a variety of approaches. An IoT-based system is investigated in this work to monitor and evaluate vehicle condition and detect probable issues information examination from the on-board license information, IMU, and GPS via a Controller Area Network Bus (CAN-BUS) device. This information is integrated and presented to consumers using data fusion techniques. In additional, the sensor data is collected and processed through a server for defect identification, maintenance research, and operational planning. The research is divided into two sections. First is the hardware architecture for obtaining GPS and IMU data through CAN-BUS for evaluation, and the next is the server software that delivers sensor data from the in-car embedded system to the distant server and makes this information meaningful to the operator. CAN-BUS analysis is useful to acquire information on the both driving conditions (frequency of pedal use, gas consumption, and so on) and vehicle state (engine speed, speed, steering angle and so on). Most of this information is sent via a CAN receiver attached to the ARM Cortex A8 CPU. The IMU data comes from an accelerometer sensor attached to the processor through I2C, and the location data comes from a GPS module. All data sources that are present in the system and the knowledge that they may supply to the system The CAN-BUS data gives detailed information on the vehicle's state, including speed, steering angle, braking frequency, and the fraction of the accelerator pedal that is depressed. The technology is made up of three major components: hardware, server software, and analytical software. The hardware unit is an embedded system that includes numerous peripheral components such as GPS / GPRS, Wi-Fi, and Bluetooth, as well as an ARM Cortex A8 CPU that runs the Android operating system. The MCP2515 CAN receiver receives CANBUS data in this embedded system. The module that interacts through SPI satisfies the user program's demands via the Application Interface (API) developed for CAN-BUS communication with the relevant server database. The velocity data is transmitted to the applications via the android device and prepared API by the IMU sensor, which is linked to the CPU through I2C. The application module includes an Android-based client application that integrates CAN-BUS data (such gasoline, engine revolving brake pedal, speed, and steering angle) with GPS and IMU data, as well as server software that interfaces with this application. The functionality of the driver is decided using the acquired data and analysis tools on the web server. The database server, which is accessible via a management interface, section uses users to follow the car 24 hours a day and retrieve all car-related data via immediate sensor data. Furthermore, driver behavior and driving parameters may be deduced from sensor data gathered by the analysis system. The implementation of an IoT car monitoring and diagnostic system employing cloud apps is described in this study. The observed engine and vehicle data is transferred through CAN-BUS and integrated with GPS position and IMU sensor data before being transferred to a cloud server through a mobile communications link. As a consequence, vehicle data, position data, and drive information may be kept on the cloud database. Discovered knowledge can be shown in real time by the cloud server application.

The research in [15] developed a mechanism for measuring vehicle speed that was based on a range of methodologies. The primary goal of this research study is to build a sensor system architecture for all autos that may release data obtained via active signals. RFID sensors mounted on the vehicles' roofs collect statistics in real time. Then they rebuilt the system to automatically affect the speed of the cars based on the turn of events. They used the fuzzy logic algorithm to regulate the motion. The gadget consists of two different units: a zone transmitter unit and a receiver unit. The receiver unit has a display screen as well as a speed control. The transmitter unit operates at 8MHZ in the 260-470MHz radio frequency range. Because this is permitted spectrum, they are not subject to FCC charges or regulations. The transmitter is cordless and may be used up to 100 feet away from the cars. While the data from the transmitter is relayed to the receiver mounted in the zones' cars. The embedded mechanism in the cars then instantly warns the driver to reduce the vehicle speed and enter the zone speed established by the operator. The device offers the driver a certain amount of time to reduce the vehicle's speed. However, the car's receiving device will immediately lower the car's speed in accordance with the zone and remain the speed until the vehicle is outside of that zone. After then, the driver can manually raise the speed and arrive at his or her location. During this whole paper, they quickly covered the issue that today's individuals drive very recklessly, accidents occur regularly, and we all lose significant lives as a result of making foolish mistakes while driving. As a result, in order to avoid such accidents, this initiative informs drivers to reduce their speed for the proper location. It can aid in the prevention of accidents that occur in slow-moving zones. It has the potential to save many lives. The given circuit regulates the motor speed. The PWM technique controls the speed of the dc motor, and the R module measures velocity. A circuit is required for the circuit. The speed of the pulse train is determined by the input speed. This circuit is useful for driving dc motors at the desired speed.

Researchers in [16] investigated a methodology for monitoring vehicle speed that was based on a variety of approaches. Vehicle overspeed monitoring stations are located along the motorway and may detect cars travelling in just single direction. Whenever the vehicle passes and pushes the two piezo-electricity sensors that are placed on the highway at a specific distance, the extracted features of the piezoelectricity sensors are recorded by the RCM3000 module's interrupt management software. The speed may then be calculated in real - time basis. If the speed exceeds the limit, two photographs of the vehicle will be inspected. The first close shot image is used to detect the car number plate, while the second establishing shot image is used to capture the interstate panorama for the peccancy action. Each vehicle overspeed monitoring system has a unique IP address, and the data package, which includes the overspeed vehicle's speed, peccancy time, peccancy location serial number, compressed picture of the establishing shot, and other information, is delivered to the remote monitoring station over the GPRS network and the GGSN gateway. The two sensors S1 and S2 are installed on a highway based on a predetermined distance L. The output signal of a sensor, which is created as a vehicle wheel passes the sensor, must be filtered and regulated by the regulating circuit since it is combined with a variety of disturbing signals. The Rabbit 3000's interrupt pin will then appropriately detect it. Simultaneously, the velocity of the wheel, i.e. the speed of the vehicle, may be calculated in order to determine the time interval between two successive pulses emitted by the same sensor. When the peccancy of overspeed is determined by evaluating it to the upper limit speed, the current speed, serial number of the monitoring station, and peccancy time are noted. The two CCD cameras are installed well above highway and are linked to the supervisory main board via coaxial wires. When the Rabbit 3000's CPU senses an overspeed occurrence, it instructs the cameras to take two vehicle images. The first close-up photograph is used to identify the car number plate, while the second image is used to capture the highway panorama for the peccancy action. When the brightness is low at night yet the headlights of the cars are dazzling, the number plate in the image is difficult to discern. To reduce disruption, an additional lamp-house is installed above the street and employed to take photos at night. The monitoring station's core is the control main board, which employs the Rabbit 3000 as its CPU. TLC5510 is an 8-bit A/D converter controller manufactured by TI Corporation in the United States. Because CCD camera output signals are analogue, the TLC5510 is needed to transform the analogue picture to digital image signals that the RCM3000 module can detect. TLC5510 has a conversion speed of 20Mbps and can fulfil picture capturing requirements. Speed is a critical element in the highway safety management system, as well as a fundamental technical specification extensively used in sectors such as bridge vibration condition monitoring, railway junction tracking, railway station automated management, and so on. In this study, a wireless monitoring system of vehicle overspeed based on RCM3000 enables remote speed monitoring by exploiting the GPRS network's benefits such as broad coverage, high reliability, cheap cost, and so on. This new measurement is both foreground and practical. The methodology also extends to other observation areas including the sea, weather, geology, and so on.

In [17] study examined a mechanism for measuring vehicle speed that was based on a number of methodologies. A shadow-region-based statistical nonparametric (SNP) methodology is presented in this research to build a unique model for shadow detection of all pixels in an image frame. To efficiently and scientifically create a model for shadow detection, we do not examine each pixel in image sequences; instead, they developed the model by studying a shadow region in a single picture frame. A colour ratio between illuminated pixels and shadow pixels is used as an index in our design to create a unique model for each shadow pixel. They further proved that under the Lambertian condition, the ratio may be deemed constant in all picture sequences. When compared to currently known approaches, the model creation approach needs far less work and hence a significantly lower computing burden. To increase the efficacy of shadow suppression even more, a spatial analysis postprocessing stage is introduced to validate the real shadow pixels. The identified shadow region's geometry and boundary information are utilised to validate the real shadow pixels. The representations of outcomes in this presentation only apply to scenarios where cars are clearly isolated from one another. When automobiles are very near to one other and their shadows are huge, their shadows frequently overlap other cars in the image frame. Under these conditions, the geometric features of genuine shadows are useless for shadow detection. For example, if a vehicle's cast shadow is unidentifiable and the shade is near to other cars, the shadows cannot be appropriately segregated using size differentiation. Additionally, if a real shadow is surrounded by automobiles, the ratio will be less than 50 percent. It is simple to demonstrate that boundary filtering will misidentify the genuine shadow as an object. To overcome the shadowing issue, edge and colour information from cars and shadows may be used to validate shadows and vehicular traffic. The challenge of how to establish appropriate transfer tracking as cars or their shadows begin to intersect each other remains unanswered. In this study, an SNP approach is used to build a colour ratio model for traffic imagery shadow detection. In terms of shadow suppression rate and computing time, our strategy surpasses two well-known techniques. The suggested model has been implemented effectively to an ITMS. Even in the shadows, traffic characteristics such as vehicle speed and traffic flow may be collected with high precision. The absolute error rate for estimating vehicle speed is less than 5.3 percent and within 3.7 percent for estimating turn ratio. More attention should be placed in future studies on improving the robustness of shadow recognition and vehicle detection under occlusion and diverse lighting situations. On the one hand, the Gaussian ratio model developed under a certain lighting environment may fail in a significantly different lighting environment. Building a library of ratio models for varied lighting circumstances will be useful in traffic management technologies. Shadow pixels that are near to moving vehicles or overlap other vehicles, on the other hand, may be misclassified as moving-vehicle pixels. In traffic photography, pixels inside one shadow region may contain identical colour information. The colour dispersion may be used to locate uniform subareas, which may then be used to validate the real shadow region. Vehicle shadowing has an impact on image system reliability, hence methods for distinguishing individual cars should be devised. Color data for individual monitored cars can be a helpful tool in resolving this issue.

The study in [18] looked at a system for assessing vehicle speed that was based on a variety of approaches. In this research, an image processing strategy for constructing a video-based highway vehicle tracking system is given. The suggested technique is divided into two stages: modelling and monitoring. A 3-D model of the backdrop road picture is created during the modelling step. Each automobile in the scene is separated and tracked across several frames during the tracking step, and estimations of vehicle speeds in separate interstate lanes are determined. The findings of experiments on frame sequences acquired from Santa Ana region highways will be discussed. The proposed system employs an activation approach to construct a road model in order to compare the frames with a backdrop picture and to generate a connection between the identified vehicle positions on each frame and three-dimensional space. The direct mapping improves the computational efficiency of the run-time approach. Images of moving autos are separated in each frame and tracked over consecutive frames in the subsequent stages of monitoring. The operation of producing the background from the frames is done on a regular basis to account for variations in the lighting and overall conditions of the roadway. This option adapts the vehicle detection method to the scene conditions. The background picture is discovered by inspecting many frames in the picture sequence. Areas of the road are covered by automobiles in any given frame. As time passes, the automobiles will shift, revealing the covered road. Monitoring automobiles in a series of pictures and obtaining their spatial coordinates results in a three-dimensional geometrical representation of the route. By allowing direct 2-D image mapping, this approach will significantly cut computing costs during runtime. The tracking algorithm detects and maps the cars into the real 3-D scene using the road model and backdrop picture. Once each precise tracking phase on the entire sequence is achieved. The automobiles appear as big solid blobs in the difference image, but noise and camera jitter cause erroneous non-zero difference values that often cover only a few pixels. At the low level processing stage, these excess points are deleted. To link the blobs in each frame, connected component techniques are employed. The position, area, and density estimate of each blob are determined. Noise is defined as blobs that are tiny or have a low density. The tracking algorithm uses the position of the blobs in each frame to follow each vehicle through the series. The tracking method examines each frame in a three-dimensional space generated by stacking many frames and finds nearby blobs in neighbouring frames. The first appearance of automobiles in the sequence is handled by establishing a threshold in the search for immediate neighbours. The method for tracking the speed of automobiles in separate lanes of a motorway has been presented. The procedure is divided into two stages: road modelling and tracking. Proper assessment of traffic speed in various motorway lanes enables for the timely implementation of appropriate reaction techniques to potential congestions. Experiment findings on frame sequences acquired from Santa Ana region highways show that the suggested method has a high performance reliability.

The investigation in [19] focused at a system for evaluating vehicle speed that used a range of methods. In this study, they created and built an adaptive video-based vehicle recognition, categorization, counting, and speed-measurement tool for real-time traffic data gathering employing the Java programming language and OpenCV. It will be used for traffic flow monitoring, planning, and regulating in order to manage transportation networks as part of the implementation of intelligent transportation management systems in smart cities. The suggested system can identify, identify, count, and measure the speed of cars travelling down a certain road. It can collect traffic data in csv/xml format from live and recorded video and transfer it to a central data-server. For removing background from the item, they used the MOG2 background subtraction method, which separates foreground objects from the background in a succession of picture frames. As a plug-and-play system, the suggested system can identify, label, and count cars of all types and sizes. They tested the suggested system in Dhaka, Bangladesh's metropolis, at six distinct locations under various traffic and environmental circumstances. In Dhaka, the overall average reliability for identifying all types of cars is greater than 80 percent. The suggested system has two modes of operation:

recorded video mode and real-time mode. It turns video into a series of picture frames, then removes the backdrop and detects moving objects. There are two sorts of video sources: recorded video and real-time video. Originally, an area-threshold for vehicle detection is specified. To count and quantify the speed of identified cars, a count line and speed line are established. During this step, the planned system is installed and configured. At this step, they position cameras on lampposts or pillars of roadways at an angle to provide a clear view of the road. Then create a count line and a speed line, and also establish the distance threshold between the count line and the speed line, which is the true distance in meter. Other standards were established, such as the minimum vehicle size, picture threshold, and so on. The camera captures a stream of data from the road and delivers it to the system for further processing. Background subtraction is a widely used technique for recognising moving objects in video captured by static cameras. For object extraction from video, the suggested system employs the background removal approach. A minimum area can be established as an area threshold in the proposed system. If the size of a moving item exceeds the given minimum area-threshold, it is only regarded a vehicle; otherwise, it is disregarded. As a result, no person will be detected in the system. A speed line is also included in the proposed system to measure the speed of moving vehicles. It is better to establish both the count line and the speed line while establishing the system. The suggested system may create traffic data and analyze video and classifying vehicle type, length, speed, and time and date. It may create traffic data in either csv or xml format. It is a plug-and-play system. The current outcomes show that the suggested system can identify, characterize, count, and measure the speed of moving vehicles with greater than 80 percent accuracy.

In [20] authors examined a system for measuring vehicle speed that employed a variety of approaches. This study provides their efforts to automatically detect traffic congestion levels from video streaming. Considering a stationary camera view and a pre-defined road section to observe, their system can calculate traffic speed and overcrowding levels (i.e., free flow, sluggish moving, or congested) in real time. The technology is extremely beneficial as a strategy for enhancing further automated systems that can employ this information. The methodology in their system is configured to discover the best forty features, each of which is at least 30 pixels away from other features. Corner characteristics that have not previously been traced will be added to the tracing list. Following the identification of corner features, the following step is to trace the movement of features on the tracking list. The traffic speed is calculated by dividing the average distance of feature dislocation by the time required to transfer the features into their new placements. Because they already knew the length of the projected picture, they can properly identify the average distance of the feature movement. Additionally, the time it would take to transfer features from their starting place to their current area may be calculated using the FPS (frame per pixel) value and the number of video frames missed since the last frame processing. The efficiency of traffic speed is calculated by evaluating the system's estimated speed to the actual speed. They only examined one-way traffic for this reason. The real vehicle speed is determined by monitoring the time it takes a single vehicle to pass through two pre-defined limits with specified distances between them. When the back vehicle is ready to leave the start and finish borders, the start and finish times are recorded. The average of all genuine vehicle speeds is the genuine traffic speed. They have presented the design of a system for monitoring traffic congestion levels in real time in this work. The video frame processing is split into two steps. While traffic density can be computed, the first stage estimates it by a process of edge recognition, morphological closure, and selective flood fill. The image that was processed in the first step is then utilized in the second step. In this stage, the point characteristics of cars (represented by blobs) are recovered using the Shi-Tomasi corner sensor and then traced using the Lucas-Kanade tracker to assess traffic speed. The efficiency of the system is revealed by the evaluation findings based on two films obtained from two separate locations with differing levels of traffic congestion. It has been able to anticipate traffic speeds and congestion levels in real time with great accuracy. Despite its efficacy, there is still much opportunity for development. To begin with, it may not be adequate for detecting traffic situations on multi-lane roads since each lane may have a distinct traffic scenario in some circumstances. Because our methodology does not take this into account, it occasionally overestimates the actual circumstances. The prevalence of motorbikes, which might potentially lead to overgeneralization, is the next concern.

The authors in [21] investigated a method for monitoring vehicle speed that used a number of methodologies. In this study, they proposed the design and implementation of a low-cost and dependable Internet of things framework comprised of an array of RFID sensors for real-time car monitoring as it travels from one point to another on the high-speed highway. When compared to image processing-based systems, the vehicle's unique detecting capacity via RFID sensor network makes it a better alternative. In this research, real-time stamps are extracted from an RFID sensor network array, and the speed of the vehicle is estimated in real-time using Euler's techniques. An Arduino platform with an Ethernet connection may be utilised as a core controller in this case, and the resulting data may be accessed through the internet through cloud computing. They advocated using RFID technology to detect the vehicle since it is a non-contact automated identification approach that can detect traffic tools and get associated data via radio frequency signal, thus the RFID system's overall operation is not influenced by night or bad weather. Vehicles are outfitted with passive RFID tags in the planned RFID system. The interrogator, which consists of an antenna and a transceiver and decoder (the RFID reader), generates a signal that activates the RFID tag, allowing it to read the ID contained in it and use it to identify that exact vehicle. Each node on the road has an RFID reader, and nodes link to the internet through a WAN link such as WIFI or Ethernet. The node might be any tiny embedded module equipped with an IP stack capable of sending the timestamp to the cloud. When the automobile gets close to an RFID reader-enabled node, a timestamp is created and communicated to the internet cloud. The nodes are assumed to be at identical distances from one another in this case, therefore we can construct real-time distance versus time graphs of the specific vehicle using these timestamps. When they have the real-time distance versus time plotting, they can extract several kinematics characteristics from it, such as velocity. So, they found the velocity from the distance versus time plotting and also estimating the same utilizing modified Euler's algorithm to estimate the vehicles acceleration at the next query point, and by doing so, they monitored the motion of the vehicle. If the car exceeds the speed limit, the driver can be notified via online apps, and the highway traffic administration can obtain the same data in real-time. India is now ranked first in the world in terms of traffic deaths, hence there is an urgent need to limit the number of road traffic-related deaths across the country. Vehicle velocity monitoring in real time is critical for avoiding deadly accidents on highways. When a motorist violates the limitations, alerts might be issued to the driver.

The researchers of [22] studied a technique for measuring vehicle speed that included many approaches. With focal-plane analogue preprocessing, the proposed system and method take use of the image sensor's unique properties. Sparse asynchronous data output with high temporal precision and low latency, great dynamic range, and low power consumption are among these properties. The system can monitor vehicle velocities ranging from 20 to 300 km/h on up to four lanes at the same time, day and night, and under various air conditions, with a precision of 1 km/h. The results of vehicle speed readings acquired during a system test installation on a four-lane highway are given and discussed. The speed estimate's accuracy has been assessed using calibrated light-barrier speed data. The output tracking error has a standard deviation of 2.3 km/h and a mean that is close to zero. An embedded traffic data sensor defined as a particular, bio-inspired Silicon-Retina imager with focal-plane analogue preprocessing is provided in this research. The got connected sparse sequential data output with low computational cost and low latency, as well as a wide dynamic range and a power efficiency. Unlike many other non-video-based traffic data collecting systems, the sensor provides a considerable degree of freedom in terms of organized competitive and can service several lanes. Unlike typical CCD or CMOS imagers, which record image irradiance and create constant data volume at a set frame rate regardless of scene activity, the SR sensor has an array of autonomous, self-signaling pixels that adapt in real-time to relative changes in light intensity by posting their location on a delayed arbitrated bus. Static sceneries yield no output because pixels that are not stimulated by a change in light are not activated. There is really no time quantization at this moment since there is no pixel readout clock. The detector is fully separate of scene lighting, directly captures object reflectance, and significantly decreases redundancy while keeping exact time information. Since output bandwidth is dynamically allotted to dynamic sections of the picture, rapidly moving vehicles may be detected even in fluctuating illumination circumstances. The notion of vehicle speed estimates and methods for an embedded traffic data sensor have been introduced. The implemented data processing operates on an AE data stream given by an SR temporal derivative image sensor with a temporal resolution of 10 milliseconds. The information processing technique benefits from the SR imager's capacity to identify relative intensity changes as well as its fast asynchronous transmission. The described technique has been built on a small embedded device and tested for incoming and leaving traffic directions ranging from 20 km/h to 120 km/h.

In [23] researchers investigated a strategy for monitoring vehicle speed that includes a variety of methodologies. An method for moving vehicle localization in digital picture sequences is introduced and investigated at the beginning. It might be used in the future as a step in traffic monitoring, speed verification, accident detection, looking for stolen automobiles, number plate identification, and so on. It is based on the selection of portions that differ from the backdrop in successive frames. It necessitates previous preparation of the backdrop image, which is not always practicable, particularly when monitoring is done continually. As a result, the process of creating the backdrop picture is based on averaging frames for an assumed time while also doing localization. It consists of a few steps: subtracting a frame from the backdrop (used to choose new items on the screen), binarization, dilatation, filling gaps, and building the rectangle surrounding the item. The purpose of these procedures is to distinguish isolated cars from the backdrop. The strategy is based on what is arguably the simplest and quickest way - selecting areas that are distinct from the backdrop in successive frames. This approach necessitates the creation of the backdrop picture in advance. However, it is not always easy to achieve in real life situations. Particularly when picture processing takes more than a few moments. There really are two issues to address in this case. To begin with, the conditions (e.g., weather, lighting) during object localization might be significantly different from the ones at the time the background image was taken. They are always evolving. It is evident that a few hours after collecting the backdrop image, rain, snow, and other elements may substantially alter the actual landscape. The main factor is the difficulty in capturing a good backdrop image free of automobiles and pedestrians in congested streets, highways, and so on. Such circumstances are sometimes only met at night. As a result, the original approach of producing the backdrop picture is employed in this study. It is based on averaging frames across an expected time period. The procedure can be done as many times as the user desires, such as once a day, every hour, every quarter, and so on. After the backdrop picture has been prepared, the real localization process may begin. The procedure of producing the backdrop might potentially be carried out concurrently with the localization. The following processes are used in this paper: removal of a frame from the backdrop (done to choose new items on the scene), binarization, dilatation, filling the gaps, and creation of the rectangle encompassing the item. Just several tests were run on a Pentium D dual-core processor with 1GB RAM. The data were not statistically examined because the observer's perception was the most relevant here. Only he can determine whether or not the localization is operating properly. As a consequence, the findings were visually graded based on images supplied by the algorithm that identified possible automobiles. Figure 6 depicts two examples. During testing, the average 'frames per second' (fps) rate was 6 or 7. It is certainly too tiny for real-time use, although it should be noted that those findings were acquired in Matlab. The execution of the idea in another language will undoubtedly increase this pace; 12 frames per second will suffice. Nonetheless, 6 frames per second may be used effectively in many real-world applications of the localization problem. To be more specific, twelve distinct previously acquired greyscale picture sequences were employed. As it turned out, if a moving vehicle was close to the camera, it was appropriately localized. In that scenario, the size of the object sufficed, and the subsequent processing was more successful. Many more issues developed when automobiles were situated at a distance. To begin with, the modest size of the items made differentiation from the background quite difficult. Furthermore, the issue of opacity was more significant. It's clear because when automobiles go farther away from the camera, their forms shrink and there are more of them. To overcome this issue, it is suggested that the examined region be limited, for instance, to the lower part of an image, where things near to the camera are located.

Study explores an approach for controlling vehicle speed that involves a range of approaches in [24]. The very first purpose of this report is to assess how precise time data from such systems are, and the second goal is to see if the reliability of one site varies over time. They also explored whether calibration coefficients can be applied to enhance the precision of the weight data collected. The final goal is to evaluate the weight data accuracy classes and subsequent areas of application in accordance with COST 323. The processes require analyzing dynamic weight data from a piezo-based TEC to static weights. This allows for an evaluation of the system's exactness, and it is done on two different dates to see if it changes over time. Calibration coefficients are often used to try to increase accuracy. Finally, accuracy classifications are computed in accordance with COST 323. Their study indicate that there was a systematic error between both the dynamic and static weights, which grew as vehicle weight rose. Furthermore, the system's effectiveness altered with time. Varying mean static weights throughout the two data collections, created by various vehicle samples, were one major contributor to this. Furthermore, calibration coefficients enhanced the consistency but not the precision of dynamic weight data. To evaluate the TEC system's accuracy, researchers compared dynamic weight data with equivalent static weights, which are supposed to represent the genuine weight. Each TEC system registration contains a time stamp that corresponds to the registered loads, but the number plate is not logged in this system. As previously stated, this is due to legal considerations. They compensated for the missing number plate registration by placing a private video camera adjacent to the TEC site and capturing every car that passes by. Following this, they recorded the attributes and time of each vehicle that passes by the TEC site. In this study, they investigated the accuracy of weight data from a TEC equipped with piezoelectric sensors. They also investigated if the accuracy varied over time and how it would be improved by using calibration parameters. According to their observations, there is a systematic error between Both the dynamic and static weights are available in classical piezo-based designs. WIM systems, as well as piezo-based TEC. The structured paraphrase dynamic weight data were discovered to be precise enough for various applications. Additionally, they discovered that the system's accuracy varies with time. The present moment different mean static was one element that contributed to this vehicle weights, resulting in unrepresentative vehicle samples. TEC offers interesting weight data, but additional study is needed to see how precise these data may be. Continuing research is gathering data from 20 new piezo-based TEC locations, and preliminary findings reveal the same trend as in this work. In addition, they will attempt to detect errors automatically and generate correction factors based on information such as the average weight of the front axle and the average weight of personal vehicles.

The researchers in [25] investigated a method for managing vehicle speed that incorporates a variety of tactics. As a result, this study presents a tracking framework based on roadside lidar to detect and track cars with the goal of estimating vehicle speed accurately. Within this system, on-road cars are initially recognised from observable point clouds, followed by a centroid-based tracking flow to provide initial vehicle transformations. The tracking flow employs a tracker that employs the unscented Kalman Filter and the joint probabilistic data association filter. Furthermore, to increase the accuracy of projected vehicle speeds, vehicle tracking is adjusted using an image matching algorithm. To provide for real-world settings, the efficiency of the proposed technique was assessed using lidar data received from two separate panoramic

3-D lidar sensors, a Robo Sense RS-LiDAR-32 and a Velodyne VLP-16, at a traffic signal and a road intersection, accordingly. Verification using standard data provided by a test vehicle outfitted with highly accurate systems reveals that more than 94 percent of cars could be spotted and tracked, with a typical speed accuracy of 0.22 m/s. Vehicle detection, tracking, refining, and speed verification are all part of the suggested technique. Vehicles are identified in three steps, then tracked by UKF and JPDAF, which use the centroid of the cluster as the vehicle position, leading in vehicle speed biases owing to the inconsistencies of the scanned clusters. As a result, a tracking refining module is being created to address this issue. The key method in this module is image comparison, which converts vehicle clusters to 2-D pictures. Ultimately, the anticipated speeds are compared to a reference from a test vehicle outfitted with an independent controller. Because detection algorithm has been reduced to a vehicle and nonvehicle classification issue after static point reduction, it is achieved in the suggested scheme by a simple rule-based predictor and two machine learning-based classifiers, SVM and RF. Although the modest number of samples, the training dataset is suitable for a classifier designed to differentiate cars from other moving objects. On the one hand, "other moving objects" in urban contexts mostly relate to walkers, cyclists, motorcyclists, and a few other false positives. Because their characteristics differ from those of cars, they are easily distinguishable from automobiles. As a result, the precision and recall are reasonably good, and this stage should not be regarded as a bottleneck in the workflow. The rule-based technique, on the other hand, produced higher recall and was thus used in subsequent studies. Due to a lack of training data, early experiments of deep learning algorithms did not yield sufficient results. Nonetheless, given the growth of deep learning and the growing availability of open-source benchmark data, more attempts to incorporate deep learning methodologies will be investigated. The research also demonstrated high-precision 3-D lidar data may be predicted to have a speed accuracy of c. 0.2 m/s, which might benefit more thorough and precise traffic flow or activity recognition. Because of the faster refresh frequency, the precision of the final motion control is similarly great, allowing precise acceleration and deceleration measurements to be acquired. But, one major disadvantage is the inherent absence of RGB information, which makes vehicle identification more difficult than when using camera phone data. The paper describes an integrated vehicle tracking framework that makes use of roadside lidar data. During the first case, vehicle clusters were identified from raw point clouds using a three-step methodology. Then, for each vehicle, a centroid-based tracking approach was used to detect clusters. After then, a modification module was employed to enhance the correctness of vehicle speeds from 0.41 to 0.22 m/s, exceeding the precision recorded in recent literature. The analysis shows that lidar sensors can identify and track cars at a variety of speeds in typical urban areas, indicating that lidar sensors may be used for precise speed control.

In [26] authors studied a system for controlling vehicle speed that employs a number of methods. The video-based traffic monitoring system is demonstrated. The system's goal is to provide a high-level description of the traffic scene, including vehicle position, speed, and class. Techniques are provided for identifying moving vehicles, distinguishing cars from their shadows, tracking, and categorization. If indeed the shadow is not separated from the cars, it is especially difficult to classify them under sunny lighting conditions. Our innovative categorization algorithm runs in real-time on decreased hardware. The shadow may be isolated from the vehicle, and understanding about the shadow's form may be effectively utilized. The shadow methodology itself involves the use of high understanding about the morphology of the scene as well as global data. In this research, a video-based traffic monitoring system with a shadow managing method is provided. It is demonstrated that traffic monitoring is doable even with low-cost gear and under challenging lighting circumstances. It is also demonstrated that a high-level description of the seen scene may be constructed using the information acquired from all moving objects. The predicted outcomes of low-level image algorithms may be calculated using this description. Depending on the scenario, the difference between the actual and predicted outcomes might be designed to improve low-level processing settings or to choose between distinct alternative methods.

The techniques of the real-time traffic monitoring system are given in this study. The shadow analysis technique has produced positive results in tests. The high-level techniques (tracking, classification, and communication) are run on an Intel i486-based PC. The low-level functions (detection and shadow separation technique) are built on a DSP (the Motorola DSP96002). The high-level techniques are written in C, while the low-level time-critical techniques are implemented in the DSP's assembler. The picture has a resolution of 256x256 pixels and 256 grey levels. The frame rate is 5 Hz. The techniques were evaluated under regular traffic and lighting circumstances for many hours on different picture sequences. The classification accuracy of vehicles was greater than 99 percent. With no prior information of the observed scene, the shadow detection rate was 95 percent during the setup phase. In this example, the shadow was appropriately recognized. The shadow was dismissed after several minutes in 90 percent of the remaining 5 percent examples since the shadow separation algorithm could not identify shadows in this direction. The information about prospective shadows throughout the day is kept and applied for future shadow analysis. If a shadow occurs in these instances, which is usual for the tracking system, the shadow's likely orientation is well-known. As a result, the detection performance rises to 98 percent. In 90 percent of these situations, if the direction was accurately determined, the shadow could be isolated from the cars. The categorization response rate was greater than 95 percent after limiting the size of the vehicle over time.

The developers of [27] explored a system for managing vehicle speed that uses a variety of strategies. The Generic Obstacle and Lane Detection System (GOLD) is described in this study as a stereo vision-based hardware and software architecture that will be deployed on speeding vehicles to improve road safety. Based on customised computational hardware, it detects both generic barriers (without symmetry or form limitations) and lane orientation in a supportive environment (with painted lane markers) at a rate of 10 Hz.The viewpoint effect is eliminated from both left and right stereo pictures using a geometrical transform provided by a specified hardware module. The computing output is presented on both an on-board computer and a power button to provide visible information to the driver. The evaluation was performed on the mobile laboratory (MOBLAB) prototype land vehicle, which again was driven for more than 3000 kilometers at speeds of up to 80 km/h through extra-urban highways and highways, demonstrating its durability with regard to shadows and fluctuating light conditions, diverse road textures, and driver behavior. This paper presents a system for lane and distance measurement that meets the stringent real-time restrictions imposed by the automobile industry. The entire system was put through its paces using photos captured by the stereo vision system deployed aboard the innovative vehicle MOBLAB. It must have been driven for almost 3000 km via extra-urban highways and highways, under various traffic and lighting circumstances, at speeds of up to 80 km/h. GOLD is currently being tested on ARGO as well. However, because autonomous vehicle employs stereo vision, the accuracy of the results is inextricably linked to the alignment of the vision system. Nonetheless, because the ultimate goal of obstacle detection is to determine the open area in front of the vehicle rather than a comprehensive 3-D representation of the environment, coordinate system becomes less crucial. ARGO is now testing an update to the GOLD system that may use be same and do knowledge of data fusion between the two capabilities of lane detection and obstacle identification.

In [28] researchers investigated a method for regulating vehicle speed that employs a number of tactics. In this research, they gave a thorough examination of the impact of these two variables on lane detection accuracy and performance. They used a variety of cutting-edge image classifiers on the BIT-Vehicle and LabelMe data sets. The data collection is downscaled into several sizes to produce a variety of spatial resolutions. Furthermore, they investigated the influence of colour by converting each colour version to a gray-scale version. Finally, employing over 46 000 individual tests, they made a reasonable conclusion about the influence of these two properties on the classification accuracy and performance of image classification techniques. The experiments revealed that the colour and spatial resolutions of the vehicle pictures have no serious influence on the identification results achieved by most state-of-the-art image classification systems. Many image classification algorithms, however, need a link between spatial resolution and processing time. Their discoveries had the potential to save not just money but also time for vehicle categorization systems. Several ways have been created that use specialized hardware components instead of, or in addition to, cameras. In order to gather vehicle features, a range sensor was employed. The feed forward neural network was used to do the categorization. The Global Positioning System (GPS) was developed to measure vehicle trajectories as a distinguishing feature to categorize vehicles. Wireless accelerometer and magnetometer sensors were employed to determine vehicle axles, resulting in a high recognition rate. However these approaches achieve excellent recognition rate, they need the use of specialized gear and settings. For vehicle categorization, many vision-based algorithms based purely on normal digital photos have been developed. Some of them are meant to identify and determine car emblems, while others are meant to detect and identify geometrical properties of cars. Alternative strategies, on the other hand, focus on visual cues, which are thought to be a strong mechanism for creating a tidy vision-based vehicle categorization system. For vehicle categorization, photos with minimal spatial resolution yield almost the same identification rates as photos with high spatial resolution, according to this study. That is, higher spatial resolution photos contain too many extraneous features for vehicle classification, whereas low resolution photos remove these extraneous elements. As a result, a camera with a modest spatial resolution is sufficient for the vehicle categorization task. True-color cameras are unnecessary; a monochrome camera will sufficient, particular if the system's primary purpose is generic vehicle type categorization. For most picture classification frameworks, colour information does not help recognition rate.

Researchers tested a system for limiting vehicle speed that utilises a variety of approaches in [29]. A unique object-based technique is devised to automatically remove moving vehicles and determine their speeds from two successive digital aerial photos. To categorize the items in the picture, several characteristics of grey values and sizes are checked. By eliminating large things such as roadways, automobiles and their related shadows may be distinguished.

To determine the speed, the cars and shadows from the two photos are first identified. The related automobiles in these photographs are connected based on their order, size, and proximity to a threshold. Furthermore, the movement speed may be calculated and used the distance between the matching vehicles and the time lag between the two photos. Their test results is a promising outcome for identifying the speeds of moving cars. The suggested approach will be used in the future for a pair of Quick Bird panchromatic and multi-spectral photos with a coarser spatial resolution. In this study, a new approach for both vehicle extraction and speed detection is devised. Based on their grey levels, pixels in a picture are initially formed into objects. Following that, the objects are categorized and identified using a variety of grey value and size indices. The retrieved cars and shadows are saved to the traffic data. The car's velocity is then determined by comparing the results of vehicle extraction from two successive photos. On digital aerial photos, the proposed method is tested. Two photos covering the same region with a time lag are required to determine vehicle speed. To begin, an overlap area is taken from two successive photos to generate two photos covering the same region. Geometric distortions between two pictures arise as a result of the camera's perspective projection. As a result, registration with the pairs of photos was performed using 8 ground control points. After registration, the pixel sizes of the two photos in a pair differ. Ultimately, picture mosaicing is used to assemble images in the same pixel sizes. Using a pair of digital aerial photos with a brief time lag, a novel automated methodology for vehicle extraction and speed detection was developed. Environmental factors such as road lines, nearby trees, and road sign boards were discovered to have a substantial effect on vehicle extraction accuracy. The majority of automobiles were effectively retrieved as a consequence of the case study for two regions in Tokyo. Vehicle speeds were also retrieved with high accuracy from two consecutive photos. The performance of vehicle matching, on the other hand, is dependent on the precision of vehicle extraction.

In [30], authors investigated a system for restricting vehicle speed that employs a number of tactics. In this paper, they offered a simple approach for predicting the relative location of the vehicle in the next frame based on the current frame picture. The predicted ratio between such a vehicle and the left-right lanes may be calculated using the vehicle's speed and the frame rate of a camera. Experimental results reveal that the suggested method produces less than 5.28 percent error for various automobiles and cameras. Lane detection is a very well known problem for which a traditional approach can be implemented. When lanes are detected using a variety of image processing algorithms. It is possible to generate a virtual longitudinal line indicating the middle of the vehicle (or the middle of the image). The lengths from the continuous line to the left-right lanes may then be used to compute the lateral position ratio. The camera's frame rate and vehicle speed can be used to predict how far the vehicle will move longitudinally in the next picture frame from the present image. A style of management for predicting where the vehicle will be laterally in the following picture frame is proposed in this paper. Using vehicle speed and camera frame information, the proposed system can estimate the speed-adaptive lateral ratio between left and right lanes. When the ratio is included in the control system of self-driving automobiles, it will aid in lateral control for lane maintaining. The results of the experiments confirmed the effectiveness and validity of the presented proposal. Furthermore, because the processes are simple and straightforward, the suggested notion is feasible for implementation in a hardware system.

CHAPTER 3

Proposed System

In the proposed methodology, the practical implementation of image-processing based vehicle speed monitoring system makes use of video stream obtained from static cameras, performing casual video recording forming a set of frames to estimate the accurate speed of the targeted moving vehicle present in the scene. In each frame from the video stream, Gaussian mixture model segments the object in motion from the backdrop by tracing the detected objects inside a particular region and then calibration is carried out. Figure 3.1 shows the proposed block diagram.

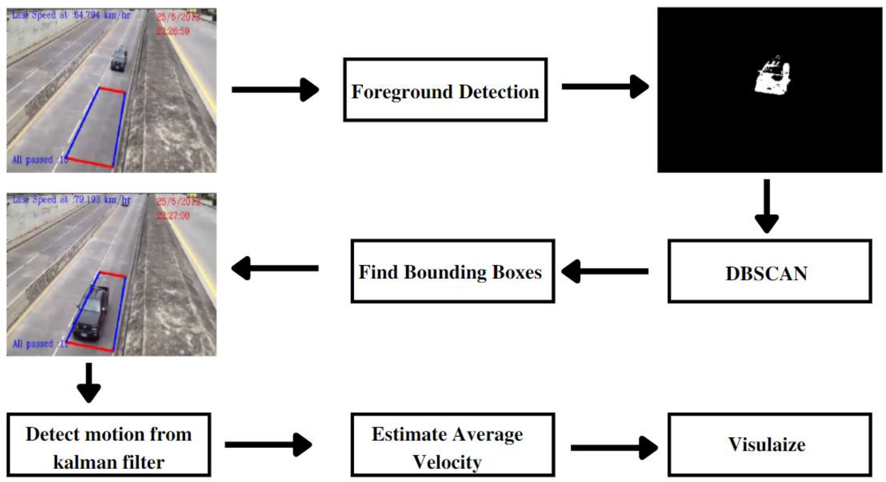

Figure 3.1: Block Diagram

The input to the monitoring system is a recorded video stream using camera with 10 frames per second. And the screen size of the recorded video is 1280x1024. The input video stream consists of two masks namely mask1 and mask 2. The distance between these two masks is considered as 10 metres. The final speed of the targeted automobile is calibrated by the mentioned formula in the report taking the time instances at which the automobile reaches the masks that is at the respective frames. For software four videos are taken to calculate speed and compare them. In case oh hardware the input is recorded over a camera.

3.1 Foreground Detection

"Background subtraction" is a technique for finding moving objects in a video sequence for example, cars driving on a freeway. The idea is that subtracting the current image from a time averaged background image will leave only non stationary objects. It is, however, a crude approximation to the task of classifying each pixel of the current image; it fails with slow-moving objects and does not distinguish shadows from moving objects. The basic idea of this research is that we can classify each pixel using a model of how that pixel looks when it is part of different classes. We learn a mixture-of-Gaussian classification model for each pixel using an unsupervised technique- an efficient, incremental version of EM. Unlike the standard image-averaging approach, this automatically updates the mixture component for each class according to likelihood of membership; hence slow-moving objects are handled perfectly. Our approach also identifies and eliminates shadows much more effectively than other techniques such as thresholding. Application of this method as part of the Roadwatch traffic surveillance project is expected to result in significant improvements in vehicle identification and tracking. Figure 3.2 shows the output received after foreground detection.

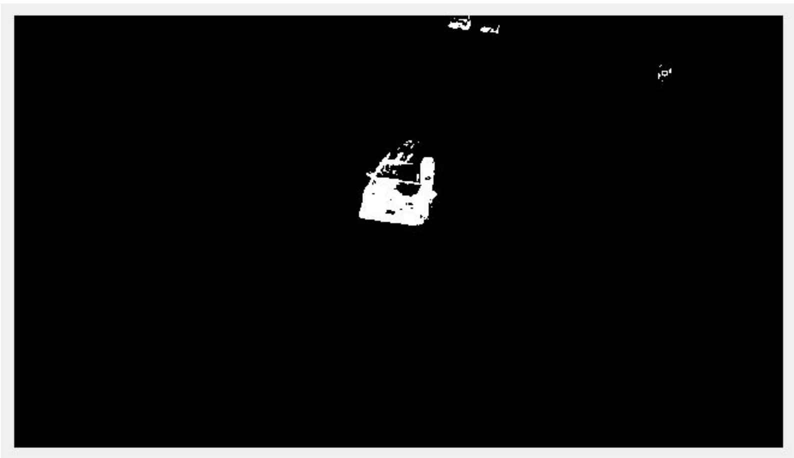

Figure 3.2: Output received after foreground detection

This is a method for real-time segmentation of moving regions in image sequences involving "background subtraction", or thresholding the error between an estimate of the image without moving objects and the current image. The numerous approaches to this problem differ in the type of background model used and the procedure used to update the model. This paper discusses modeling each pixel as a mixture of Gaussian and using on-line k-means approximation to update the model. The Gaussian, distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively is considered part of the background model. This results in a stable, real-time outdoor tracker which reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes.

Using the background threshold T and classifying the current pixel using the matching test. This scheme is improved in five different ways:

- By using a different measure for the matching test: In this case, the ordering and labelling phases are conserved and only the matching test is changed to be more exact statistically. Indeed, we have to check every new pixel against the K existing distribution to classify it in background pixel or foreground one. This test gives a binary mask and was chosen in an approximation to the true Maximum A Prior (MAP) solution to permit a real-time implementation. This approximation causes false positive detections due the use of the intervals. To solve this problem, likelihood maximization is used instead of the approximation.

- By using a Pixel Persistence Map (PPM): Pixel Persistence Map (PPM) which is a map of the same dimension as the frames containing at each location (x,y) the weight of the Gaussian matching the current colour of the pixel (x,y). Small PPM values indicate foreground pixels, while large indicate background ones. So, the foreground detection is made using a decision test with a threshold on the PPM. The disadvantage is that when there is camera jitter (CJ) or movement in the background (MB), the PPM needs to be bounted by a very low threshold in order not to consider flickering pixels as foreground but this threshold tends to discard true foreground pixels. To solve this problem, the threshold on the PPM in a spatiotemporal fashion using a tracking feedback. This scheme presents the advantage that flickering pixels are avoided far from the targets, while the targets themselves are not affected. The drawback of this strategy is the delayed detection of new very small targets.

- By using a foreground model: These approaches use one model for the foreground and one for the background. The background model is the Mixture of Gaussian per pixel, as the background will be different per pixel. On the other hands, foreground objects are not static and move over all pixels. So, the foreground model is spatially shared and only one model is used for describing all foreground objects depicted in all pixels. Model the foreground by a uniform distribution. Using a global Mixture of Gaussian for the foreground. Once the foreground model is determined, we have to compare the probabilities that the pixels belong to the background or the foreground. Respectively noted P(x / B) and P(x / F). where the CF and CB are the costs of false positive detection and false negative detection, and P(F) and P(B) are the a prior foreground and background probabilities.

- By using the most dominant background model: The ordering and labelling phases are conserved and the matching test is the same but instead of using a percentage of the Gaussian only the most dominant is considered to represent the background. Furthermore, instead of using the corresponding m as the reference value,using the most recent pixel value m represented by this Gaussian. The idea is to avoid any artificial value as the representation. All the previous foreground detection used the pixel as element of comparison. To enhance the robustness, other feature sizes are used.

3.2 DBSCAN

DBSCAN is the first density-based clustering algorithm. It was proposed by Ester et al. in 1996, and it was designed to cluster data of arbitrary shapes in the presence of noise. Clustering is an important unsupervised learning tool and widely applied in computer vision and image processing tasks, e.g., image segmentation. Many clustering algorithms have been proposed from different perspectives, and some of the important works in this field include k-means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), EM (Expectation Maximization) and NCuts (Normalized Cuts). While a large amount of clustering algorithms were proposed from various perspectives, most of existing algorithms require some a prior knowledge of the data to be clustered in one form or another, and their performance often depends heavily on user-specified parameters. Typically, many algorithms, e.g., k-means and spectral clustering, require as input the number of clusters, which is not easy to determine in many cases.

While some other algorithms determine the number of clusters automatically, they require other parameters as input. Examples of this kind include DBSCAN where a neighbourhood radius and a minimum cluster size need to be determined beforehand, and AP where the preference values of all data need to be specified. In either cases, the clustering results are influenced by the parameters selection, and a careful tuning process is required to generate satisfactory clustering results.

DBSCAN requires two parameters as input, it is able to generate clusters of arbitrary shapes in a region-growing manner. Based on these properties, we propose to use DBSCAN to combined these two algorithms. After extracting a cluster, we extend the cluster with DBSCAN, where the required parameters are determined from the original cluster adaptively. Repeating this process in the remaining unclustered data, we are able to overcome the over-segmentation tendency and generate clusters of arbitrary shapes.

- Minimal requirements of the domain knowledge to determine the values of its input parameters, which is very important problem especially for large data sets.

- Discovery of arbitrary shaped clusters.

- Good efficiency on large data sets.

The density-based clustering algorithms are useful to discover clusters from the datasets with arbitrary shape and of large size. These algorithms typically cluster as dense regions of points in the data space that are separated by regions of low density. DBSCAN [9] is the first density-based clustering technique. It grows clusters according to a density-based connectivity analysis. The key idea of DBSCAN is that for each object of a cluster the neighbourhood of a given radius (Eps) has to contain at least a minimum number of objects (MinPts), which means that the cardinality of the neighbourhood has to exceed some threshold. Here Eps and MinPts are the user's specified parameters which mean the radius of the €- neighbourhood and minimum number of points in the €- neighbourhood of a core point respectively. If this condition is not satisfied then this point is considered as non-core point.

DBSCAN searches for the clusters by checking the € - neighbourhood of each object in the dataset. If the €- neighbourhood of an object p contains more than MinPts, a new cluster with p as a core object is created. It then iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a new density reachable cluster. The process terminates when no new object can be added to any cluster. The pseudo code of DBSCAN algorithm is shown in figure 3.3.

```
Algorithm: DBSCAN (D, Eps, MinPts)
// All objects in D are unclassified.
Begin
FOR ALL objects o in D DO:
If o is unclassified
            Call function expand_ cluster to construct a
cluster wrt. Eps and MinPts containing o.
End

FUNCTION expand_ cluster (o, D, Eps, MinPts)
Begin
        Retrieve the Eps-neighborhood (o) of o;
IF | N_Eps (o) | < MinPts       //i.e. o is not a core object
            Mark o as noise point and RETURN;
ELSE            // i.e. o is a core object
            Select a new cluster- id and mark all objects
in N_Eps (o) with
This current cluster-id
    Push all objects from N_Eps (o)| (o) onto the
 Stack seeds;
    WHILE NOT seeds.empty () DO
CurrentObject: = seeds.top ();
Retrieve the Eps-neighborhood
N_Eps (CurrentObject)   of CurrentObject;
     IF| N_Eps (CurrentObject) | ≥ MinPts.
            Select all objects in N_Eps (currentObject) not
yet classified or are marked as noise,
            Push the unclassified objects onto seeds and
mark all of these objects with current
 Current-id;
Seeds. Pop ();
RETURN
End
```

Figure 3.3: Algorithm of DBSCAN

3.3 Bounding Boxes

A bounding box is an imaginary rectangle that serves as a point of reference for object detection and creates a collision box for that object. These are rectangles over images, outlining the object of interest within each image by defining its X and Y coordinates. This makes it easier for machine learning algorithms to find what they're looking for, determine collision paths, and conserves valuable computing resources. Bounding boxes are one of the most popular image annotation techniques in deep learning. Compared to other image processing methods, this method can reduce costs and increase annotation efficiency.

There are numerous approaches that have tackled the object localization problem. Broadly speaking, they could be classified based on the used localization forms, e.g. bounding box shapes or tightly enclosing shapes. For bounding box localization, the sliding window classifier approach usually is the preferred method.

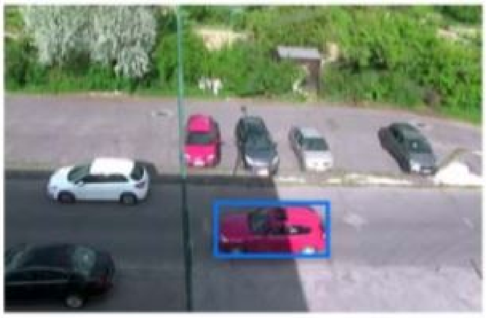

Figure 3.4: Bounding Box over the object detected

Figure 3.4 shows the bounding box which is circumcised over a the object. Sliding window classifier approach: In the sliding window classifier approach, a window of suitable size, say, is chosen to perform a search over the target image. First, a classifier is trained on a collection of training samples spanning the object of interest for detection as one class and random objects as the other class. Formally, samples belonging to the object of interest for detection are referred to as positive examples, while random object samples of no interest are referred to as negative examples. For a single object detection task, the idea is to train a binary classifier, which determines if the presented object is positive or negative. The trained classifier can then be used to inspect a target image by sampling it, starting from the top-left corner. It is noteworthy that the input dimension of the trained classifier is generally a fraction of the size or dimension of the target image; hence, sampling of target images can be achieved.

3.4 Object Tracking using Kalman Filter

Kalman Filter was proposed by R. E. Kalman in1960. It is an important task to reliably detect and track multiple moving objects for video surveillance and monitoring. However, when occlusion occurs in nonlinear motion scenarios, many existing methods often fail to continuously track multiple moving objects of interest. We have proposed an effective approach for detection and tracking of multiple moving objects with occlusion. Moving targets are initially detected using a simple yet efficient block matching technique, providing rough location information for multiple object tracking. More accurate location information is then estimated for each moving object by a nonlinear tracking algorithm. Considering the ambiguity caused by the occlusion among multiple moving objects, we apply Kalman filtering technique for reliable object detection and tracking.

We exploit a simple yet practical idea to help track multiple objects with occlusion. Although, the objects overlap with each other, different objects move with different directions and/or speeds, which do not change much during several frames. Considering the continuity of the movement, we can predict the next position of each object based on its estimated velocity when they depart. By comparing to the previously detected objects, we can match the objects after overlapping based on their predicted velocities.

Kalman filtering is widely used to track moving objects, with which we can estimate the velocity and even acceleration of an object with the measurement of its locations. However, the accuracy of KF is dependent on the assumption of linear motion for any object to be tracked. If an object takes some abrupt turns, the nonlinear movement cannot be well handled by the Kalman filter framework due to the linear movement assumption of the design of Kalman filter. Figure 3.5 shows the path annoted using Kalman Filter.

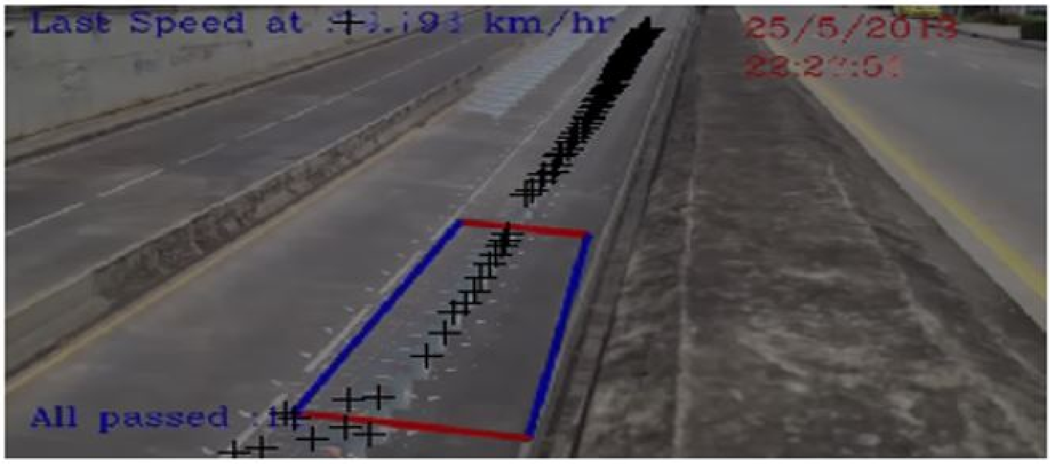

Figure 3.5: Path annoted using Kalman Filter

The idea of state space representation and incorporated it into statistical estimation theory for the development of this filtering technique. Based on the statistical characteristics of the system noise and measurement noise, the measurement variables are used as input signal, and the estimation variables that we need to know are the output of the filter.

3.5 Estimation of Average Velocity

Recent technological developments in computer vision and image processing have revealed that video cameras are an efficient means of collecting and analysing traffic data. Video based surveillance systems are more sophisticated and robust because the information that is associated with image sequences presented in a video allow us to identify and classify vehicles in the most effective manner. It consists of a stationary video camera that is placed a fixed location for capturing images.

For calculating the moving vehicle speed, the primary task is to segregate the moving vehicle from the video stream correctly. There are numerous approaches for detecting the moving object from the video stream like temporal differencing algorithm, optical flow algorithm, background subtraction algorithm. In the proposed methodology, we used adaptive background subtraction using Gaussian Mixture Model (GMM) for the object detection or the fore ground detection. For an accurate measurement of the velocity of the object, we have considered a distance of 10 metres between the masks 1 and 2. The speed calculating equation comes into the picture only if the automobile is in between the mentioned masks.

The final mask and the initial mask are considered in accordance with the motion of the automobile at various instances of time. Frame number when automobile at mask 1 considered as the initial frame. Frame number when the automobile is at mask 2 is considered as the final frame and the frame numbers are to be noted down for further calculations.

The following formula has been employed to estimate the average velocity in the proposed system:

$$speed = (distance * k)/(finalframe - initialframe) \tag{3.1}$$

Where,

distance = 10 meters, which is constant

k = constant value which is calculated initially by taking the known speed

final frame - initial frame = difference between the frame at mask 2 and mask 1.

After calculating the speeds, it is visualized whether the car is travelling at high speed or low speed.

3.6 Hardware Setup

The same procedure is performed in hardware setup to show the practical working of the proposed system.

The hardware setup shown in the image where laptop and phone are connected through IP address of WI-FI using an app called IP Webcam. The phone camera is used to record the movement of the vehicle. The speed of the vehicle is calculated and displayed using the algorithm developed in the MATLAB software. An android mobile is connected to the laptop via a Wi-Fi IP address using an application called IP WEBCAM. Make sure that the laptop with which we are running the MATLAB Code and the mobile phone with which an application called IP WEBCAM IS installed should have the same IP ADDRESS. The mobile phone which we have installed IP WEBCAM acts as the mobile camera ha the IP camera. So that the video gets tracked and records and sent to the code. When we click the run button on the Matlab software ,the video gets recorded from the mobile camera which is operated by the remote of the toy car. And the recorded video is sent and the car gets tracked and later we calculated the speed of the toy car. Figure 3.6 shows the hardware setup of the proposed system.

Figure 3.6: Hardware Setup

The code written for hardware reads the video from the connected mobile which records the movement of the vehicle.After tracking the video from frame 1 to frame 2, the speed is calculated in the same as in the software and displays the required result.

CHAPTER 4

Algorithm Details

The software used is MATLAB. It is a programming tool that is used to develop algorithms, products, and many more which help to transform this world. It has many applications like image processing, communications, Signal Processing, etc.

1. It supports Graphic User Interface.
2. Has many inbuilt Libraries.
3. Easy to access data from the databases.
4. It consists of a Tool Box where a set of functions are designed.

4.1 Functionalities Implemented on GUI Interface to access the video

Video Reader is a command that we used to read the files from the database. User Interface Get File (uigetfile) is used to pick a Matlab code i.e., when we click on the browse button the video gets selected from the database using the below command.

[Filename, pathname] = uigetfile ('*.*', 'Pick a MATLAB code file');

If none is selected that means filename and pathname is 0 then we display a message in the command window has "User pressed cancel".

Else then we concatenate both the filename and pathname using "strcat" and then read the video using VideoReader (filename) and then calculates the first frame of the video and displays it on axis.

SPEED=0;

SPEED=num2str (SPEED);

We converted the speed in numerical to string to display it on editor window.

When we press play video button on GUI a function call is made to play video Callback where it has 3 parameters hObject, eventdata, handles.

“hObject” is used to handle the play video, event data reserved is to be defined in a future version of MATLAB and handles structure with handles and user data (see GUIDATA).

The function is displayed below:

Function playvideoCallback (hObject, eventdata, handles).

We used the while loop to record all the frames of a video. Once all the frames are read, there is a pop up with a message in the dialogue box of GUI has Process Completed.

For video tracking, videotrackingCallback function is written and the parameters are similar to play video button.

Frame = []; A video frame

Detected Location = []; the detected location

Tracked Location = []; the tracked location

Label = ”; Label for the ball

Utilities = []; Utilities used to process the video

We defined empty numeric arrays initially and further it gets updated.

4.2 Implementation of Video Tracking

We have three functions which we have implemented in video tracking.

Utilities=show Detections (filename, handles, label);

ShowTrajectory (utilities, handles);

TrackSingleObject (param, utilities, filename, handles)

Show detections function is used to calculate the speed and for this and we have written another function detects object. In this we are converting the image into a grey scale image by rgb2gray. We found foreground Mask for the grey image.

utilities.foregroundMask = step (utilities.foregroundDetector, gray Image);

Later we are read the mask1.png using imread method and converted the image to binary and white im2bw method and then complementing the image to do mathematical calculations and this process we are repeating even for mask2.png.

We did foreground mask for the first frame and performed bit and operation for the mask1 and mask2 to check whether the car has reached first frame.

If sum (sum (the result of bit and) operation is ¿0 then it means the car has reached the first mask and we display a message has "car is detected" and then counted the frames. We have set initially str=0 and when str¿=1 first frame gets detected and final count gives the difference of frames. And when final count¿1 then it calculates the speed

4.3 To Annote the tracked Path

We are doing this using Kalman Filter. This is to configure the Kalman filter for tracking. We have used a computer vision toolbox we have continued till the end of the video file. It detects object we have used a function "object new". Kalman filter is done to track single objects. It is used for tracking. The 4 steps that we have implemented Kalman filter in our project is

1. Initialize the location where the object has started.
2. To track the object i.e. vehicle.
3. And store the path for each and every moment and record the path.
4. Annotate the path and visualize the path.

The Kalman filter has got many wide ranges of applications in many scenarios. The Kalman filter is especially used to track objects and concentrates on three important features:

1. To predict the future location of an object.
2. To overcome the noise when it detects the inaccurate one.
3. And to Track the multiple objects.

4.3.1 Challenges of Object Tracking

In general, when we track an object without a Kalman filter we face some challenges. In the video which we have in our database, we can observe that a vehicle that is coming from a long distance on the road. showDetections ();

To separate moving objects from background vision. Foreground Detector is used so that white pixels can be observed on the vehicle. If we go with the background subtraction only a small portion of the vehicle gets observed because of low the contrast of the vehicle and the road it is moving on. And if we try to detect the object the noise gets added to it. To observe the complete object trajectory we concatenate all the video frames into a single object. ShowTrajectory (); Two issues can be observed:

1. we are unable to calculate the vehicle's location
2. The box to annotate with the vehicle is getting missed.

To overcome the challenges we have used Kalman Filter.

4.3.2 Single Object Tracking using kalman Filter

- Using the —trackSingleObject— function helps you to create —vision.KalmanFilter— by using —configureKalmanFilter—.
- To eliminate the noise in tracking systems we use —predict— and —correct— methods
- Use —predict— method by itself to estimate vehicle's location when it is occluded by the box.
- To select the Kalman filter parameters we have use the —configureKalmanFilter— function.

The nested helper functions are included in —trackSingleObject— function. To transfer the data we have use following functions and its gets updated.

Frame = [];

Detected Location = [];

Tracked Location = [];

Label = ";

Utilities = [];

To track a single object we have written the function trackSingleObject (param). To detect the moving objects read video and to display the results we created utilities.

Utilities = create Utilities (param);

For the first time when the vehicle gets observed we have to initialize a track by Kalman Filter. f the object gets detected we have to call predict function and correct functions. If the Object is detected.

Predict (kalmanFilter);

TrackedLocation = correct (kalmanFilter, detectedLocation);

Kalman filter addresses two different cases:

1. Kalman Filter predicts the current video frame state and by using that it detects the newly detected object location.

2. When the vehicle is unable to detect it uses the previous state to

Vehicle's trajectory by overlaying all video frames of the vehicle can be observed using.

param = getDefaultParameters ();

This tracked location will be annotated by using annotateTrackedObject (); function this will give you the annotation on the vehicle which we detected. InsertObjectAnnotation function is just to get with red color annotation is shown. InsertObjectAnnotation (combinedImage, shape, region, label, 'Color', 'red');

So here we make use of the Kalman filter to estimate the value of K initially. And this keeps on updating from the previous values.

4.4 Hardware Implementation

We have used the IP WEBCAM app on our android phone which is connected by the laptop. And both the devices should have the same Wi-Fi. Components used:

1. Car
2. Mobile
3. Laptop with software installed

We have written code for GUI with the name Real-time vehicle Speed Detection. Figure. For this, we have used a tripod to hold the mobile phone. When we click on the tracked button the video which is captured from the mobile phone gets captured and the sent one is gets loaded. Then we initialize the parameter as stated below:

idx = 0;

count=0;

setstr=0;

precount=1;

It calculates the number of Frames using the code below:

NumFrames =utilities.videoReader.NumFrames ;

frmcnt=1;

SPEED='0'

Then it will run in while loop until the car has reached the mask1. As we have seen when the toy car reached the first mask the loop starts and the frame count begins. The final count value gets updated. If the final count value is greater than 0 then speed gets calculated as shown by the below equation. The distance we have taken is 10 meters that is the distance between starting position and the end position. Here the final count is used to calculate the frame count that keeps on count from starting point to endpoint. And the constant value k is multiplied that is the speed of the previous vehicle.

if finalcount is greater than 0

disp(finalcount);

$$SPEED = (10 * 86.6)/finalcount; \tag{4.1}$$

disp(SPEED);

If the final count is greater than 0, then the loop gets entered and calculates the speed. Here 86.6 is considered as a constant K value.

CHAPTER 5

Experiments and Results

Our project motivation is to calculate the vehicle speed both in hardware and software using Image processing Techniques. For software we have used MATLAB installed on our PC. For hardware, we have done using General Processing system where we will be using a toy car to calculate the speed of the toy car. The car is operated by a remote controller. And the camera which we have used is the mobile camera. And the camera is placed on a holder. Further details are clearly explained in further sections.

5.1 Software Results

Here we have explained in detail the step-by-step procedure that we have followed in order to detect the speed of the vehicles.

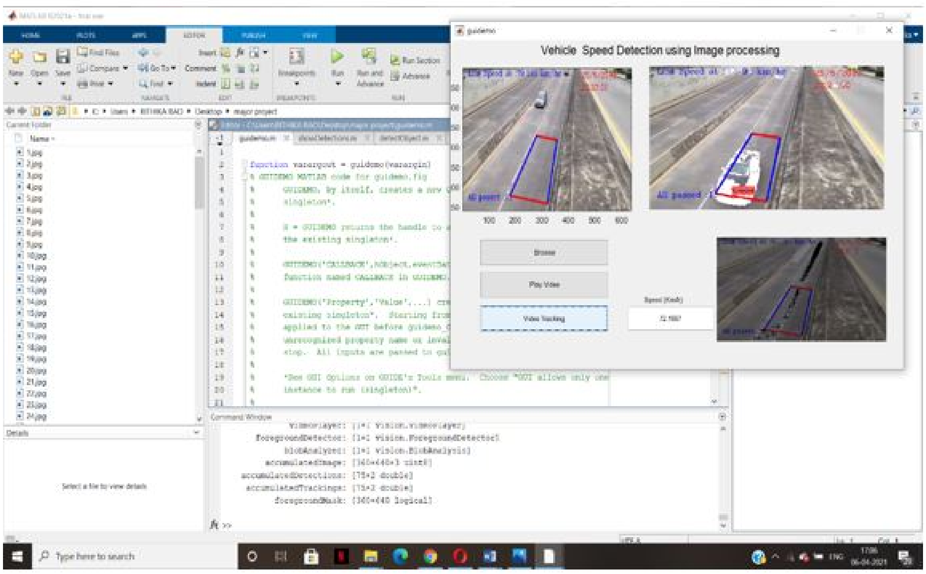

Figure 5.1: GUI Interface

For software we have built the required code which involves the techniques to detect and track the speed of the vehicle in motion. We have written various MATLAB functions in order to perform the calculation of speed. The speed which is calculated is observed on the Graphic User Interface(GUI) interface of MATLAB. Figure 5.1 shows the GUI interface we have implemented in MATLAB software .

The Graphic User Interface(GUI) which we have used has 3 options. One is to browse the video which we have in our database. After clicking the browse button the video gets selected. The second option which we have is the Play video button when we click on it the video gets played and a popup message gets displayed as “Process Completed”.The third button which we have is Video Tracking it gets tracked and displays the speed in km/hr. There are three windows in the GUI. One shows that the video gets selected and the other is to track the video and the last one is to get the path which the car has followed

Firstly we have collected the recorded video from the database that we found on the internet for software. The video which we collected from the database has vehicles moving at different speeds also with time and date the vehicle has arrived. The first step after getting the frames from the video is Background subtraction where we remove background which means the background is fixed at the back and it is compared with the current frame. That is it fixes one threshold value and it compares the frame and background is subtracted at the backside. There the pixels of the foreground are clustered and these clustered images are represented by DBSCAN(Density-based Spatial Clustering of Applications with Noise)And this clustering is shown by bounding box. And then to calculate the speed of the vehicle the unknown values are calculated using Kalman Filter and then speed is calculated for that vehicle.

The speed of the vehicle is tracked when the car reaches the first mask then we need to begin the frame count till the end of the mask. Here we have considered the masks where simply it is determined has two points one is the starting point and the other is the endpoint. The difference between the first mask and the second mask gives the frame count which is shown in Figure 5.2. We consider the bounding box which is rectangular in shape with some length where we consider it as the distance of two masks. When the car reaches the last mask, calculation of speed is performed and speed is gets updated.

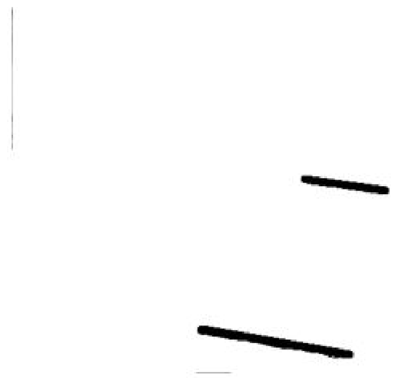

Figure 5.2: Mask Image

The mask image shown in Figure 5.2 indicates the starting and ending point. When the car passes the starting point, it indicates as the first frame, and the ending point is indicated as the second frame. So when the car reaches the first frame the frame count begins and it is calculated till the last frame. And the frame count has used to calculate the speed of the vehicle. The calculated speed is displayed when the car is passed through these two frames. The distance between these frames is considered to be 10 meters.

Figure 5.3: The car hasn't reached the mask

When we run the code that is when we click on the browse button the user can select the video from the database and it is shown in the first window of GUI. And when we click on the Play the video button until the actual speed calculation gets started. In Figure 5.3 the car hasn't reached the starting point and we can observe the last speed which is displayed on the image is 64.794 km/hr and all passed is 10.while in Figure 5.4 the car passed through starting point and reached the endpoint and we can observe that the last speed which is displayed is at 79.193km/hr and all passed the count has increased to 1 i.e the value before is 10 and now it gets updated and displays have 11. We can also observe from figures 5.3 and 5.4 that the time which is shown is also getting updated. And this is the point where the speed calculated is displayed.

Figure 5.4: The car has reached the mask

The speed and All passed gets updated. When the car has passed through the bounding area we are able to see the speed that is calculated and the all passed count also increased to 11.

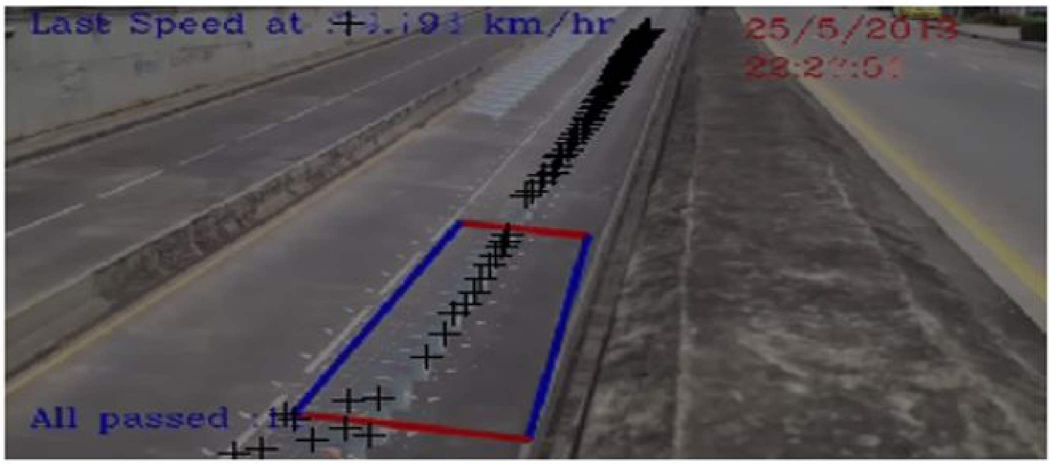

Figure 5.5: Tracked path

The speed is then calculated. To annotate the tracked object we have used Kalman Filter shown in Figure 5.5. The annotation of the tracked object is shown with black color cross lines where it is shown on the road. The path that is covered by the vehicle is determined by using the Kalman Filter. Where the Kalman filter is used to determine unknown values from series of measurements and calculations.

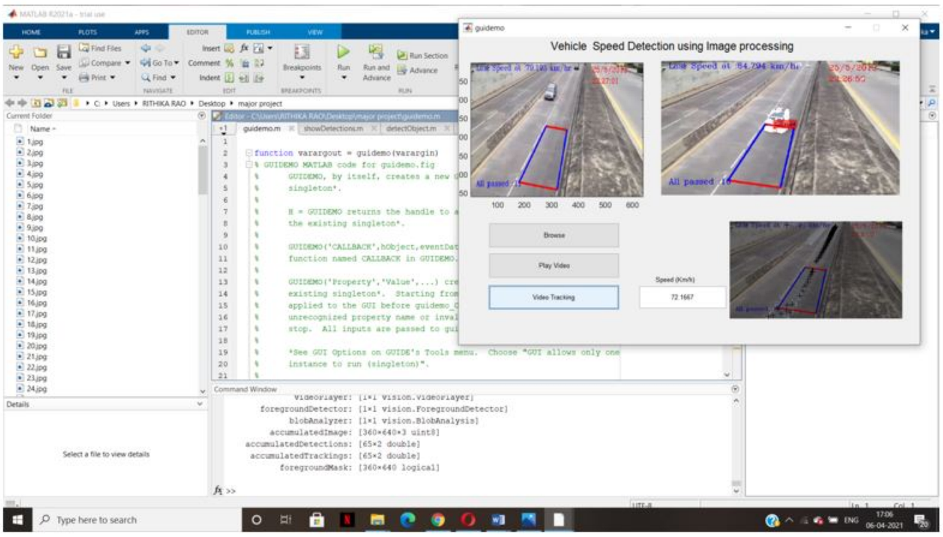

Figure 5.6: Final displayed result

After performing all the operations the speed gets calculated using the formula discussed above. The speed gets calculated and in the second window we can observe that the car is detected and labeled in red color. In the last window, the path gets tracked and shown in black color. The speed displayed is 72.947km/hr which is shown in figure 5.6.

5.2 Hardware Results

The setup that we have done for the hardware implementation is very simple. We have used the code which we have used for software. The additional thing that we have added is the GUI to track the speed of the toy car and added one more functionality. The name given for GUI is Real-Time Vehicle Speed Detection. It has a button Tracking at the below and the speed is calculated and displayed in another box which is initially shown with 0. After recording the movement of the car using a camera the speed gets displayed. Figure 5.7 shows the GUI to detect the speed of the toy car.

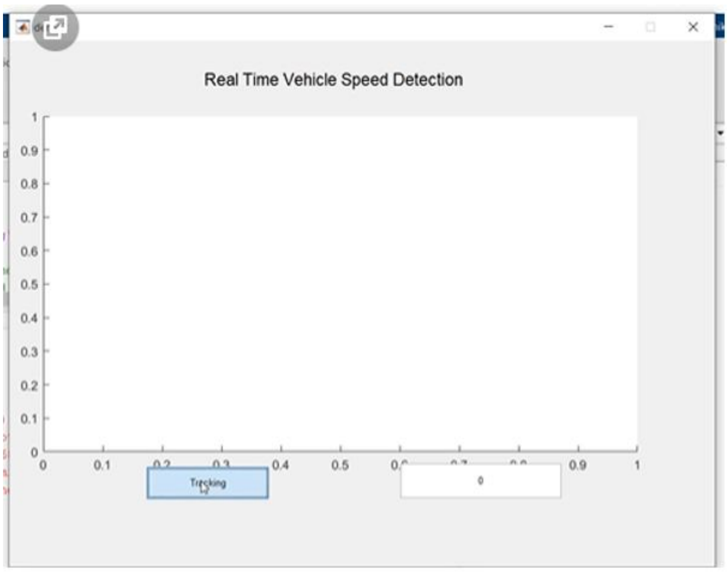

Figure 5.7: GUI Interface of hardware Implementation

Figure 5.8 shows the toy car that we have used to perform the speed calculation. The car which we have used is operated by using a remote. So that the speed which we want to calculate with different range of speeds ie low, medium, high are easily operated with this toy car. So we have selected this for our reference.

Figure 5.8: The object taken for reference in hardware implementation

The hardware setup consists of the android mobile phone is used to record the video and the target i.e. the car is tracked and the speed is calculated using the code developed in the MATLAB software. And later we have calculated the speeds as LOW, MEDIUM, HIGH. We have calculated the speeds with different operations which are operated by a toy car. the car which is operated and gets tracked and by performing background subtraction the background is removed and only foreground points are detected and they are clustered using the DBSCAN Technique. So here we consider a mask where it acts has starting and ending position. Figure 5.9 shows the mask image used in hardware implementation.

Figure 5.9: Mask Image used in hardware implementation

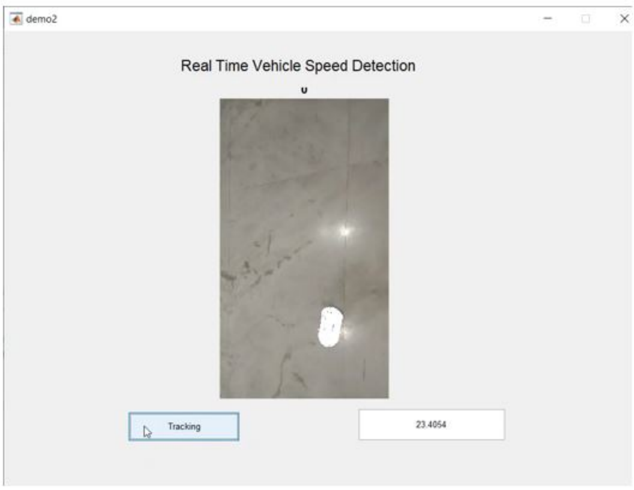

Figure 5.10: Car travelling at low speed

The distance is calculated as the number of frames between the first mask and the second mask which is shown in the above figure. And k we have taken the value as 86.6 and in the denominator we have taken the frames that are calculated in reaching the first, mask to the second mask. In the Figure 5.10 the car is operated at low speed where the frame count is less and the operated with slowly and the object gets tacked and low level of speed is shown in 23.493 km/hr.

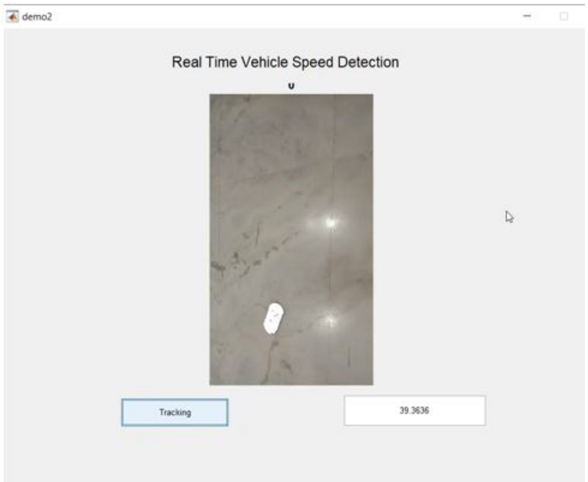

Figure 5.11: Car travelling at medium speed

In the below Figure 5.11 the car is operated at a medium speed where the frame count is mid-range and the operated with medium level of speed and the object gets tacked and middle level of speed is shown as 39.3636 km/hr.

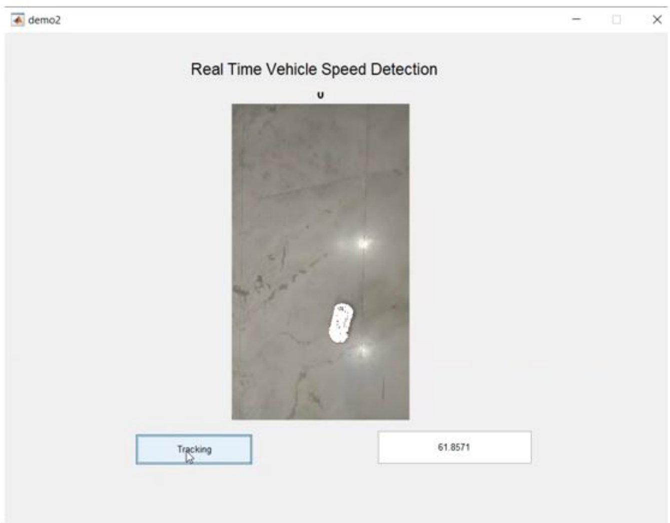

Figure 5.12: Car travelling at high speed

In the below Figure 5.12 the car is operated at high speed where the frame count is high range and the operated with high level of speed and the object gets tacked and high range of speed is shown as 61.87km/hr.

Hence that the car is traveling at a lower speed and later the car traveling at medium and higher speeds respectively. With the calculated speeds of the vehicle, we can conclude whether the vehicle is traveling at a higher speed or lower one in order to avoid the major accidents taking place due to over speed.

CHAPTER 6

Conclusions and Future Scope

6.1 Conclusion

We developed a vehicle speed measurement system concept in which we upgrade the optical low approach with Kalman filter tracking to overcome the problem of overlapping with static foreground objects and also increase speed identification. In order to provide a more exact object representation, foreground detection using a Gaussian mixture model was merged with DBSCAN clustering. Based on our findings, we can infer that the combination of optical flow and Kalman filter techniques produces pretty decent results even when the picture quality supplied by the commercial camera is low.

6.2 Future Scope

Efficient hardware results can be obtained by caliberating the camera with inclusion of all it's properties. Since we have considered only mask image for caliberation, some inefficient results are obtained when car misses the frame while travelling from initial point to final. Since we used a mobile phone for recording the video. If the mobile is slow and touches only two frames and reaches from endpoint, leads to inefficient result.

Lot of constraints get affected like a shadow, lightning position. To avoid this we can use calibrated camera and Embedded Hp should be installed in laptops so that it executes the code on time without any delay.

REFERENCES

[1] Jozef Gerát, Dominik Sopiak, Miloš Oravec, and Jarmila Pavlovicová. "Vehicle speed detection from camera stream using image processing methods". In: *2017 International Symposium ELMAR*. IEEE. 2017, pp. 201–204.

[2] Pranith Kumar Thadagoppula and Vikas Upadhyaya. "Speed detection using image processing". In: *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. IEEE. 2016, pp. 11–16.

[3] Prem Kumar Bhaskar and Suet-Peng Yong. "Image processing based vehicle detection and tracking method". In: *2014 International Conference on Computer and Information Sciences (ICCOINS)*. IEEE. 2014, pp. 1–5.

[4] BS Shobha and R Deepu. "A review on video based vehicle detection, recognition and tracking". In: *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*. IEEE. 2018, pp. 183–186.

[5] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and S. Sarasvady. "DBSCAN: Past, present and future". In: *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*. 2014, pp. 232–238. DOI: 10.1109/ICADIWT.2014.6814687.

[6] PA Shinde and YB Mane. *IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*. 2015.

[7] Nilesh Ananthanarayanan. "Intelligent vehicle monitoring system using wireless communication". In: *2013 International Conference on Advances in Technology and Engineering (ICATE)*. IEEE. 2013, pp. 1–5.

[8] Ling Li and Ting Wang. "Design and Implement of Vehicle Monitoring System Using 3G Technology". In: *2012 International Conference on Industrial Control and Electronics Engineering*. IEEE. 2012, pp. 832–835.

[9] Ravi Kishore Kodali and M Sairam. "Over speed monitoring system". In: *2016 2nd international conference on contemporary computing and informatics (ic3i)*. IEEE. 2016, pp. 752–757.

[10] Mayuresh Desai and Arati Phadke. "Internet of Things based vehicle monitoring system". In: *2017 Fourteenth International Conference on Wireless and Optical Communications Networks (WOCN)*. IEEE. 2017, pp. 1–3.

[11] R Sasvanth Narayan, V Loganathan, Prateek Lakkar, and B Suganthan. "IoT Cloud based Optimization Of Vehicle Using Monitoring Systems". In: *2019 International Conference on Communication and Electronics Systems (ICCES)*. IEEE. 2019, pp. 1108–1112.

[12] Hua Ye and Gang Ding. "A digital vehicle monitoring system based on 3G for public security". In: *2010 International Conference on Computer and Information Application*. IEEE. 2010, pp. 146–148.

[13] Yukino Sato. "Radar speed monitoring system". In: *Proceedings of VNIS'94-1994 Vehicle Navigation and Information Systems Conference*. IEEE. 1994, pp. 89–93.

[14] Ercüment Türk and Moharram Challenger. "An android-based IoT system for vehicle monitoring and diagnostic". In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2018, pp. 1–4.

[15] Eshan Pathak, Tushar Sharma, Shubhra Dixit, et al. "RF Based Smart Zone Vehicle Speed Monitoring and Control System". In: *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE. 2018, pp. 664–668.

[16] Ma Zengqiang, Gao Guosheng, Song Wanmin, and Yan Yan. "Wireless monitoring system of vehicle overspeed on freeway based on GPRS". In: *2008 27th Chinese Control Conference*. IEEE. 2008, pp. 550–553.

[17] Kai-Tai Song and Jen-Chao Tai. "Image-based traffic monitoring with shadow suppression". In: *Proceedings of the IEEE* 95.2 (2007), pp. 413–426.

[18] B. Gloyer, H.K. Aghajan, Kai-Yeung Siu, and T. Kailath. "Vehicle detection and tracking for freeway traffic monitoring". In: *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*. Vol. 2. 1994, 970–974 vol.2. DOI: 10.1109/ACSSC.1994.471604.

[19] Amit Ghosh, Md Shahinuzzaman Sabuj, Hamudi Hasan Sonet, Swakkhar Shatabda, and Dewan Md Farid. "An adaptive video-based vehicle detection, classification, counting, and speed-measurement system for real-time traffic data collection". In: *2019 IEEE Region 10 Symposium (TENSYMP)*. IEEE. 2019, pp. 541–546.

[20] Okaswara Perkasa and Dwi H Widyantoro. "Video-based system development for automatic traffic monitoring". In: *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*. IEEE. 2014, pp. 240–244.

[21] Rahul B Pendor and PP Tasgaonkar. "An IoT framework for intelligent vehicle monitoring system". In: *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE. 2016, pp. 1694–1696.

[22] Martin Litzenberger, Bernhard Kohn, Ahmed Nabil Belbachir, Nikolaus Donath, Gerhard Gritsch, Heinrich Garn, Christoph Posch, and Stephan Schraml. "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor". In: *2006 IEEE intelligent transportation systems conference*. IEEE. 2006, pp. 653–658.

[23] Dariusz Frejlichowski. "Automatic localisation of moving vehicles in image sequences using morphological operations". In: *2008 1st International Conference on Information Technology*. IEEE. 2008, pp. 1–4.

[24] Timothy Pedersen and Torbjorn Haugen. "Vehicle weight estimation based on piezoelectric sensors used at traffic enforcement cameras experiences from the norwegian system". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 1–5.

[25] Jiaxing Zhang, Wen Xiao, Benjamin Coifman, and Jon P. Mills. "Vehicle Tracking and Speed Estimation From Roadside Lidar". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 5597–5608. DOI: 10.1109/JSTARS.2020.3024921.

[26] Michael Kilger. "A shadow handler in a video-based real-time traffic monitoring system". In: *Proceedings IEEE Workshop on applications of computer vision*. IEEE Computer Society. 1992, pp. 11–12.

[27] Massimo Bertozzi and Alberto Broggi. "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection". In: *IEEE transactions on image processing* 7.1 (1998), pp. 62–81.

[28] Khaled F Hussain, Mahmoud Afifi, and Ghada Moussa. "A comprehensive study of the effect of spatial resolution and color of digital images on vehicle classification". In: *IEEE Transactions on Intelligent Transportation Systems* 20.3 (2018), pp. 1181–1190.

[29] Fumio Yamazaki, Wen Liu, and T Thuy Vu. "Vehicle extraction and speed detection from digital aerial images". In: *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*. Vol. 3. IEEE. 2008, pp. III–1334.

[30] Seongrae Kim, Junhee Lee, and Youngmin Kim. "Speed-adaptive ratio-based lane detection algorithm for self-driving vehicles". In: *2016 International SoC Design Conference (ISOCC)*. IEEE. 2016, pp. 269–270.