A FRAMEWORK FOR INTEGRATION OF WIRELESS SENSORS NETWORK AND OBJECT DETECTION SYSTEM TO MONITOR CARELESS DRIVING: THE CASE OF ADDIS ABABA

A Thesis Presented

by

Birhanu Mesfin Alemu

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements for the Degree of Master of Science

in

Computer Science

February, 2021

ACCEPTANCE

A Framework for Integration of Wireless Sensors Network and Object Detection System to Monitor Careless Driving: The Case of Addis Ababa City

By

Birhanu Mesfin Alemu

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

____

Internal Examiner

Temtim Assefa, PhD ______________ April 25, 2021

External Examiner

____

Dean, Faculty of Informatics

26 January 2021

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Birhanu Mesfin Alemu

April 25, 2021

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as an advisor.

Dr. Asrat Mulatu

March 6, 2021

Addis Ababa

Ethiopia

February 2021

Table of Contents

Acknowledgments

First I would like to thank God and Next, I want to acknowledge the following people without whom I would not have been able to complete this research work and without whom I would not have made it through my master's degree! Mrs. Ayelech Gugsa you always and everywhere promote best mothers, Dear Brother and Friend Yonatan Daniel, you are a happening at a serious moment to me, my new boss and Friend Tesfaye Golja I mean you too and finally I would like to appreciate my adviser Dr. Asrat Mulatu for his big patience and contribution for the success of this work.

I wish to express my cordial thanks to St. Mary's student support service office as a whole that I got the kind and accountable for why they are there. Finally, I would like to express my respect to Dr. Getahun for his encouragement and respect for his students as a whole.

List of Acronyms

PC: Personal Computer

LED: Light Emitting diode

ETA: Ethiopia Transport Authority

EWCA: Ethiopian Wildlife Conservation Authority

DVSS: Doppler's Vehicle speed Sensor

WSN: Wireless Sensor Network

WSN: Wireless Sensor Node

ODS: Object Detection System

TCP: Transport Control Protocol

IP: Internet Protocol

ADS: Autonomous driving system

RTA: Road traffic Accident

GUI: Graphical user Interface

DC: Direct current

LCD: Liquid Crystal display

2D: Two dimensional

CVIS: Cooperative Vehicle Infrastructure System

MSFF: multi-scale feature fusion

KMPH: kilometer per hour

NGO: Non-governmental organization

JIS: Japanese Industrial Standards

GPS: Global Positioning system

ADC: Analogue to Digital Converter

PDF: probability density function

DLT: Direct linear transformation

MoG: Mixture of Gaussian

GUI: Graphical User Interface

2NF: Second Normal Form

BCNF: Boyce-Codd Normal Form

COCO: Common Object is Context

List of Tables

List of Figures

Abstract

Road traffic accidents are a global problem affecting all sectors of society. An accident is an error that occurs in the driver-vehicle-roadway system. According to the literature reviewed different things in the driver-vehicle-roadway system contribute to traffic accident among which Careless driving is a very common reason especially in developing countries like Ethiopia. As a result of car accidents, the death of human and property loss has been part of our news menu every day. Every morning we see dead animals (beginning from small birds to the bigger Mammals like hyena and other) on the street. Previous research works on Wireless Sensor Network proposed different solutions to reduce accidents with the mechanism of prevention, warning, and reporting, but they are not enough to bring a strong solution. As the survey made shows the cases of most car accidents are a result of violating traffic rules such as over-speeding, abrupt lane change, and traffic light violation.

This research demonstrates that the Ethiopian traffic management system has been using very old systems which has very limited capacity. In this proposed work, monitoring driving behavior with the help of wireless sensor technology is the target. So the proposed research work focused on developing a framework for integrating wireless sensor network and object detection system which used python socket programming, Doppler vehicle speed sensor, surveillance camera, light-emitting diodes, and proximity sensor nodes to develop the system. It is shown that the functionality of the proposed framework help in reducing abrupt lane-changing behavior, traffic light violation, and over speeding.

Keywords: Wireless sensor network, object detection system, Intelligent Transportation Systems, MonitorCarDriving

Chapter One

Introduction

1.1 Background

The pervasiveness of the transport solution to transport problems can have a major influence on people's lives, other living, and non-living thing's safety [1]. To provide safe, rapid, comfortable, convenient, economical, and environmentally compatible movement of people and goods transport engineering is operating. It is working through the use of technological and scientific principles to the planning, functional design, operational, and management of facilities for any mode of transport. Traffic engineering, a branch of transport engineering, deals with the planning, geometric design, and traffic operation of roads, streets, and highways, their network, terminals, abutting land, and the relationship with other modes of transport [2].

Accessibility and mobility are the key functions of transportation systems. Accessibility is the ability to reach desired goods, services, activities, and destinations. Mobility is the movement of people and goods [3]. Restrictions to accessibility and mobility, which can result from traffic congestion, have a profound impact on the national economy, quality of life, and the nation's safety and security [4][5].

Intelligent Transportation Systems (ITS) and transportation analysis tools allow us to understand disruptions in transportation systems, predict effects, and therefore, mitigate the impacts of such events [6]. Road traffic injuries are a global problem affecting all sectors of society. To date, road safety has received insufficient attention at the national and regional levels. This has resulted in part from: a lack of information on the magnitude of the problem and its preventability; a fatalistic approach to road crashes; and a lack of the political responsibility and multidisciplinary collaboration needed to tackle it effectively. However, much can be done to reduce the problem of road crashes. Indeed, many high-income countries have been able to reduce their road traffic injury burden by up to 50 percent over the last few decades [7].

Even though the total number of reported accidents decreased in the last few years, safety is one of the challenging issues in the transportation industry in Ethiopia [8]. The need to evaluate the impact of traffic signals (in the traffic network) on safety is the reason for developing a suitable methodology in this thesis to answer several questions on this topic.

The traffic accident is the result of the multiplicity of factors and it is often the interaction of more than one variable that leads to the occurrence of an accident. Accidents occur as a result of the interaction of many different factors among which road & traffic characteristics [9].

Most investigations have revealed that 70% to 80% of all traffic accidents are due to human error. The term human error however is often controversial. It doesn’t satisfactorily describe that a large number of injury and deaths that occur on the road as the result of driving errors while abilities to do so are impaired by alcohol or drugs, lack of experience, lack at the distribution of attention [10].

Car accidents are accidental collisions between automobiles. Car accidents can damage one or more autos, people, or structures. Car accidents are also called traffic accidents, auto accidents, road accidents, and motor vehicle accidents because which thousands of deaths and hundreds of thousands of disabilities each year. Worldwide, car accidents kill an estimated one million people each year. Africa has one of the highest road traffic death rates in the world, with little difference in rates between those countries categorized as low-income. Since Ethiopia millennium the number of road traffic accidents in Ethiopia was alarming. In the past eleven years, the number of road traffic accidents was estimated at around 291577. From those traffic accidents, 36796, 54731, 58987, and 141063 road traffic accidents were fatality, serious injuries, light injuries, and property damage respectively. Traffic accidents increased periodically at an alarming rate and it was a serious problem throughout the globe particularly in developing countries like Ethiopia. The car accident has been one of the most causes of death in Ethiopia and the world as a whole [11].

It further expressed that road traffic accidents can be defined as “An accident that occurred on a way or street open to public traffic; resulted in one or more persons being killed or injured, and at least one moving vehicle was involved too.

The accident is defined as an error in the driver-vehicle-roadway system and it must be recognized that different types of accidents are caused due to differences at any given location namely, rear-end, side-swipes, head-on, night-time, bad-weather, etc. Bad weather accidents can result due to a road pavement or due to inadequate signs for inclement weather [12]. Different causes contribute to a traffic accident where Driver' behavior has the biggest contributions. A driver who fails to use proper care, pay adequate attention to the road, or drive safely is considered as carelessness. This can cause serious harm to others sharing the road. Careless driving can also be defined as operating an automobile or street vehicle without providing due care, attention, or reasonable consideration for others using the road.

Fatigue driving & drunk and driving are the main contributions to road crashes, Up to 20% of all traffic accidents are believed to be due to drowse while driving and 40% of traffic accidents due to drink and drive. Falling asleep while driving causes at least 100,000 crashes annually; 40,000 lead to nonfatal injuries, over 1500 result in fatal injuries [13].

Examples of careless driving include texting while driving, driving at an unsafe speed, failing to maintain a safe distance behind other automobiles, Causing an accident involving another vehicle, pedestrian, or cyclist, Running a stop sign or red light, Failing to check mirrors when changing lanes or driving in reverse, The general failure to provide proper care.

Most of the present-day vehicles are embedded with an intelligent system of sensors, a Global Positioning System (GPS), and transceivers for transmission and reception of signals [14]. The vehicle closer to the accident area receives emergency messages and reacts according to the situation by either slowing down or performing lane change. Vehicles far from the accident zone have a higher probability of reception of emergency messages & might take decisions [15].

Developing countries face a high ratio of car accidents due to misbehaving drivers. Patterns involved in dangerous crashes could be detected if we develop accurate prediction models capable of automatic classification of the type of injury severity of various traffic accidents.

These behavioral and roadway accident patterns can be useful to develop traffic safety control policies. I believe that to obtain the greatest possible accident reduction effects with limited budgetary resources, measures must be based on scientific and objective surveys of the causes of accidents and the severity of injuries.

1.2 Motivation

As a result of misbehaving and careless driving road accident has been an increasing phenomenon. Issues like Unmonitored roads and traffic are seeking the high attention of authorities related to transportation. Even though many systems have been developed to minimize accidents, still further work is necessary to improve the situation. The application of wireless sensor networks (WSN) in road traffic monitoring systems can reduce road traffic accidents through wireless sensing & communication. Research works surveyed that a huge number of human life and other living creatures are being destroyed because of careless vehicle driving behaviors and faults. The day-to-day mass media posts and also the WHO reports showing that road traffic accidents need strong monitoring work to reduce injuries as a result of car accidents [9],[10],[11].

Therefore, to reduce accidents through monitoring careless driving behavior, this research work proposed a new system called monitor careless driving (MonitorCarDriving) to monitor how vehicles are driven through a wireless sensor network using TCP/IP communication from a sensor placed on the roadside to a remotely placed server computer which is under the administration of traffic police office. Python socket programming is used to create communication between server and client Computers. The system collects driving data in a server computer and transfers it to the client computer.

An automated detailed report is generated and saved in the client computer for vehicles driven against the pre-set traffic laws. All details of careless and ignorant driving can be generated by the server computer.

1.3 Statement of the Problem

Researchers have been coming with many different ideas of research works on how to notify, prevent and detect car accidents, but, most of these research works focus on the cases that are external to the driver which we call either a geographic violation or other cases related to the capacity of the vehicle technology contributing to the accident [14]. On the other eyes, these factors are commonly related to mechanical and technical issues that are beyond the driver's control and also considered to be external to the driver [16]. The role of the driver in road accident prevention is vital. The reaction time of the driver for acceleration or deceleration adjustment is strongly influenced by disarrangement and traffic interruption probability [15]. It is learned from literature that the dynamism in the traffic flow has an impact on driver behavior.

Therefore, this research paper is aimed at identifying the main manifestations of how drivers act carelessly learning from the traffic law of the Ethiopian transport authority as well as from drivers chosen from different sectors. Then, the study addressed the following research questions?

1.4 Research Questions

A. what driver-related factors contributed to a specific accident?

B. how to notify the driver of an insensitive environment to drive carefully?

C. How to identify careless driving?

C. How to monitor careless driving in a sensitive environment?

D. How to record driving behavior as careless from pre-taught traffic laws?

1.5 Objectives

The General Objective of this research work is to study the existing framework of car traffic monitoring systems and develop a Framework for Integrating wireless sensor networks and object detection systems to Monitor Careless Driving.

1.5.1 Specific Objective

- To review related works in the area of a car accident detection, Prevention, and recognition.
- To evaluate the performance of the developed system with different datasets.
- To examine the current traffic monitoring tools of ETA
- To develop a framework for Wireless Sensor Network and Object detection systems with graphical modeling.
- To validate, draw conclusions, facts, and results collected from Perspectives.

1.6 Significance of the research

Ethiopia has a very old traffic control and monitoring system which is not efficient and effective in traffic law enforcement activity. The major contribution of this research is to propose a framework for the Integration of WSN and object detection system for Traffic law enforcement. The finding of this study benefits traffic control and management work by automating the means in the areas which have high exposure to a traffic accident. The research work also reduced the burdens of the traffic police and indirectly creates consciousness within the drivers. This benefits the traffic management office to focus on automated traffic control and monitoring systems to modernize and simplify the job. On the other hand, the study result creates awareness for those who are engaged in driving jobs to drive carefully considering that they are being monitored. Theoretically, the novel idea of developing a system for monitoring careless driving will open the door for researchers to extend the study in similar area. Finally, the significance of the study can be understood in terms of saving the precious life of humans and other animals from the car traffic accident.

1.7 Scope and Limitations

The scope of this study is to develop a framework for integrating wireless sensor networks & object detection systems to monitor careless driving by learning traffic standards. Here, different design methods are observed, its strength and drawbacks are well discussed.

The study took the three biggest traffic laws which are speed limit law, lane-keeping, and the law of traffic lights mainly and then used proximity sensors and speed sensors to collect related data and based on which the software MonitorCarDriving is applied.

Accordingly, the proposed solution can monitor and judge how the car is being driven relative to areas sensitive for the accident (school areas, market areas, park areas, and inside the city) and relative to the law of traffic which is indicated by different signs and symptoms.

The limitation of this research is that it deals only with external character views like lane keeping, following traffic lights, and drivers with proper speed according to the laws set by ETA. Issues expressed under internal characters meaning mechanical & electrical issues of the car are left for future research study.

1.8 Methodology

The research followed a design science research method. The research procedure included the following phases:

1.8.1 Literature Review

To identify the research problem and to give a solution to the identified specific problem relevant and up-to-date literature about the state of the art of wireless sensor networks and object detection systems were explored. In addition to this traffic policies and ways of implementation, car accident cases, prevention methods, and reports are accessed. During the Literature review, things were given categories so that they were discussed very clearly.

1.8.2 Algorithm Development

Under this phase, it is planned to give the effort to design and develop an algorithm and a method for monitoring Careless driving.

The machine was feed traffic policies and to learn the behavior of the driver and to evaluate if he/she is either misbehaving. Software like Visio, Python socket programming is used.

1.8.3 Evaluation Methods

When the algorithm development phase gets complete then it is evaluated to ensure whether it can correctly do what it is intended to do. Also, the performance of this algorithm is compared to that of related works that are explored in the Literature review.

Finally, the evaluation step also helps improvement of the performance of an algorithm developed to achieve the supposed efficiency. Observation & measuring of how well the artifact supports a solution is done.

The evaluation is aimed at comparing the objectives of the solution to the actual results derived in the design and development.

During this several concerned bodies were interviewed. The researcher focused on what tools and means the office of traffic management has been using, made a group discussion, in-depth interview, and document analysis to get sufficient information needed.

1.9 Organization of the Thesis

This thesis is organized into six chapters in the following ways.

- In chapter one, an overview of the study such as background information, motivation, statements of the problem, objectives, scope, and limitation, and significance of the study were described.
- In chapter two literature review and related works are described.
- Chapter three described the research methodology.
- Chapter four described the design of the proposed solution. Contained a detailed description of the proposed hardware and software requirements, selected sensor technology and Installation, system architecture, and model.
- Chapter five presented steps of MonitorCarDriving software development and algorithms (codes) and result evaluation of the proposed solution.
- Chapter six is all about the conclusion, Research Contribution, and Recommendation of the future remaining work.

Chapter Two

Literature Review and Related Works

2.1 The Concept of Wireless Sensor Networks

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor. These nodes are a tiny device that has the capability of sensing, communication, and computation.

Usually, data are transferred to the server to monitor the environment or the required phenomena.

Wireless networks monitor environmental or physical conditions, such as sound, temperature, pressure, vibration motion, or pollutants, and cooperatively pass their data through the network to a sink or main location where the data can be observed and analyzed.

A base station or sink acts as an interface between the network and its users. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically a wireless sensor network contains sensor nodes. The sensor nodes can communicate among themselves using radio signals. A sensor node, also called mote, is an electronic device that consists of a processor along with a storage unit, a transceiver module, a single sensor or multiple sensors, along with an analog-to-digital converter (ADC), and a power source, which normally is a battery [22].

It may optionally include a positioning unit and/or a mobilization unit. A sensor node uses its sensor(s) to measure the fluctuation of current conditions in its adjacent environment. These measurements are converted, via the ADC unit, into relative electric signals which are processed via the node's processor. Via its transceiver, the node can wirelessly transmit the data produced by its processor to other nodes or/and a selected sink point, referred to as the Base Station. As illustrated in Figure 2, the Base Station, by using the data transmitted to itself, can both perform supervisory control over the WSN it belongs to and transmits the related information to human users or/and other networks [23].

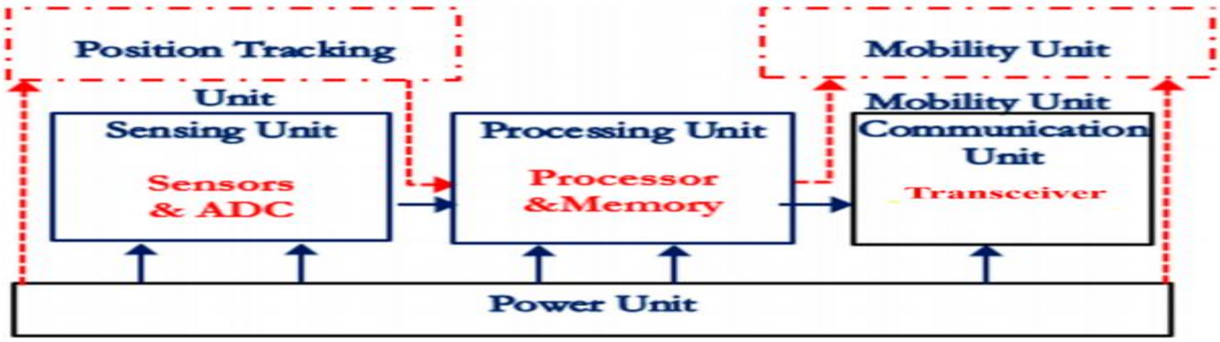

Fig.2.1 Typical Architecture of Sensor Nodes Used in WSN [23]

Like a lot of technological developments and advances, the concept of a wireless network with some autonomous elements or devices monitoring and analyzing the surrounding environment to collect, process, and analyze data, was driven mainly by military and defense organizations [24]. The development of this technology was done in four main stages with military and defense organizations as the main boosters or supporters.

Table 2.1 Table of summary of features of each generation [24].

	1st Gen. (80's & 90's)	2nd Gen. Early 2000's	3rd Gen. T0day (2010+)
Size	Shoe Size	Book or Smaller	Coin Size or Smaller, Nano size
Weight	Kilograms	Grams	Milligram or Negligible
Node Architecture	Separated Sensing, Processing and Communications	Integrated sensing, Processing and Communication	Fully Integrated Sensing, Processing and Communication
Protocol	Proprietary	Mainly Proprietary, Wi-Fi, ZigBee, WiMax, Others	Standard 802.15.4, Wi-Fi, ZigBee, WiMax, Others
Topology	Point-to-Point, Star, Multihop	Client-Server, P2P	Fully P2P
Power supply	Large Batteries or line feed	AA Batteries	AAA Batteries, Solar/ Wind Generated, Others
Life span	Hours or Days	Days to Weeks	Month to Years

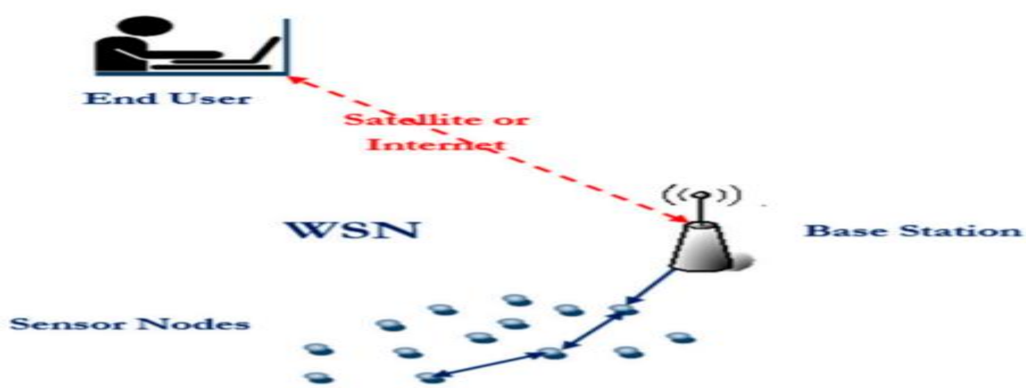

Fig.2.2 Typical Architecture of WSN [22]

2.2 Wireless Sensor Nodes

The wireless sensor node [25] is a vital part of the wireless sensor network which is located in a strategic location. Multiple applications are conducted using a wireless sensor node. The wireless sensor node consists of a controller, transceiver, external memory, and other sensors interfaced with analog to digital converters. The sensor nodes can sense the object and communicate within a certain distance for transferring and processing data by its inbuilt transceiver. The basic architecture of a sensor is schematically presented in Figure 7.

There are different types of sensor nodes used in different applications such as Mica, Mica2, Micaz, BTNode2, BTNode3, iMote, ZebraNet, TelosB, etc. Figure 2.6 shows a commercial TelosB sensor node.

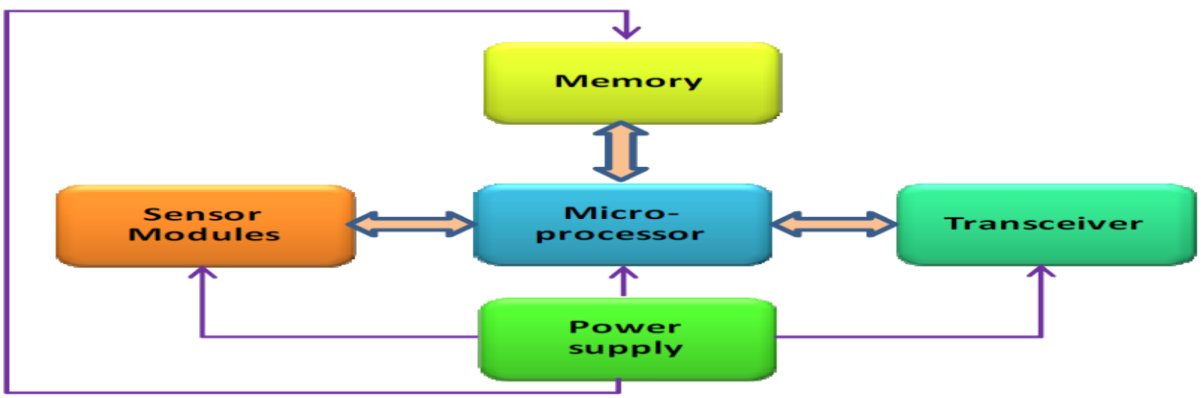

Fig.2.3 Architecture of Sensor Node (adapted from [25])

2.2.1 Sensors and Sensing Principles

A sensor is an object, which can perform the sensing task. A sensor is a hardware device that can sense, monitor, and measure physical conditions [22].

A sensor can perform on-board processing and communication. It also can store information. There are two types of sensors: passive sensor and active sensor [23].

Passive sensors are self-powered where the measurement is performed without notation of direction. The temperature sensor is an example of a passive sensor. An active sensor is a group of sensors that can generate shockwaves by explosions. Radar and seismic sensors are examples of active sensors. From a technical point of view, a sensor can translate physical parameters into the form of signals for analyzing and measuring.

A sensor acts as a transducer [27], which can convert the physical phenomena, into electrical or some other form of energy.

Furthermore, the resulting signals are processed during the signal conditioning stage. The analog-to-digital converter transforms the analog signal into a digital signal. Finally, an actuator can control the flow of signals directly in the physical world by further processing, converting, and conditioning. The following picture shows the sensing steps in the sensor.

Therefore, in this study there is a kind of sensor node chosen is

Fig.2.4 Sensing steps in the sensor (Adapted from [25]).

2.2.2 Proximity sensors

"Proximity Sensor" includes all sensors that perform non-contact detection in comparison to sensors, such as limit switches, that detect objects by physically contacting them. Proximity Sensors convert information on the movement or presence of an object into an electrical signal. There are three kinds of detection systems that perform this conversion: systems that use the eddy currents that are generated in metallic sensing objects by electromagnetic induction, systems that detect changes in electrical capacity when approaching the sensing object, and systems that use magnets and reed switches.

The Japanese Industrial Standards (JIS) define proximity sensors in JIS C 8201-5-2 (Low-voltage switchgear and control gear, Part 5: Control circuit devices and switching elements, Section 2: Proximity switches), which conforms to the IEC 60947-5-2 definition of non-contact position detection switches.

JIS gives the generic name "proximity switch" to all sensors that provide non-contact detection of target objects that are close by or within the general vicinity of the sensor, and classifies them as inductive, capacitive, ultrasonic, photoelectric, magnetic, and the like[21].

This Technical Explanation defines all inductive sensors that are used for detecting metallic objects, capacitive sensors that are used for detecting metallic or non-metallic objects, and sensors that utilize magnetic fields as Proximity Sensors.

There are different kinds of proximity sensors among which the following are some:-

- Inductive: This type of sensor is used to detect nearby metallic objects. The sensor creates an electromagnetic field around itself or on a sensing surface.
- Capacitive: This type of sensor is used for the detection of metallic objects and nonmetallic objects.
- Photoelectric: This type of sensor is used to detect objects. A light source and receiver are the main components of such sensors.
- Magnetic: This type of sensor uses an electrical switch that is operated based on the presence of permanent magnets in a sensing area.

Proximity sensors are used to collect information about how close the vehicle is to the object that has been detected by object detecting sensors. This means that Proximity sensors are sensitized by object detecting sensors and as long as no object has been detected this sensor is not active. It would be recommendable to come up with sensors with dual functionality like being able to sense proximity and do detection of an object with one sensor node to reduce the use of different types of sensors in a case like this study.

Most specifically capacitive type of proximity sensors is selected for this work due to their ability to sense metallic (sensing closeness of the body part of the vehicle) and non-metallic which can be either the wheel or other part of the car.

2.2.3 Speed Sensors

For speed estimation, I used Doppler's Vehicle Speed Sensor (DVSS) which is a fully self-contained, roadside mounted, vehicle speed measurement sensor. This non-intrusive, high-performance speed sensor shatters existing sensor performance and cost points. In addition to the low unit cost, the sensor is extremely robust and performs without maintenance for years.

The sensor is battery-powered, solar-charged, and mounts quickly on existing poles or overpasses. The Doppler Vehicle Speed Sensor-102 uses a 24 GHz Doppler microwave transceiver system coupled to a Digital Signal Processor to measure and calculate the vehicle's speed. The DVSS-102 is capable of determining the average or composite vehicle speed for a multiple-lane freeway or highway. Speed information is backhauled to MonitorCarDriving data server over a 3G cellular data link.

Fig.2.5 Doppler Vehicle Speed Sensor (DVSS-101)

Specifications

Coverage

- Range up to 1800 ft
- Installs on existing infrastructure – no new poles; mounts on virtually anything
- 20 minute installation time; 15-minute replacement time configurable coverage areas to suit specific installation requirements

Communication

- Real-Time Traffic Information reporting
- 3G wireless modem data-backhaul
- Adaptive traffic speed reporting (Variable reporting schedules based on congestion level)
- Bi-directional

Specifications are subject to change.

Measurement

- Bi-directional traffic data collection
- Configurable data acquisition/sample rate
- Average speed within +/- 1mph
- 1/10th mph for single vehicle in field of view
- HOV lane speed measurements

Mechanical

- Enclosure
- Anodized aluminum tube
- Bright White Powder Coat
- Dimensions

2.2.4 Object detection

The purpose of the object detection sensor is to sense if any moving object like a human being or animal is crossing the road or at the zebra area (allowed crossing area) vertically.

2.2.5 Light Emitting Diodes

Light-emitting diodes (LEDs) in this case come up only with two lights. LED lights function in the same way as a surveillance camera but are before the camera and lights ON as warning when the first speed and distance are sensed with unpermitted measures and it lights red light on when last which almost considered as almost stop area where the red lights ON.

2.2.6 Surveillance Cameras

After the Object is Detected and the speed of the vehicle is known at a limited distance between the object and vehicle most specifically at a yellow light area the camera became ready to capture and if the expected measure is not taken by the driver because of any reason and the car enters a red light area which is specified by the reading of speed and proximity sensor the camera keeps recording the cars' overall movement including its Car number.

2.3 Application of Wireless Sensor Networks

According to the paper in [24], due to the potential in monitoring and alert possibilities of WSNs, the proliferation, and expansion of this type of network is currently increasing and spreading through several areas of study and real-life applications.

Due to the unlimited potential of WSN applications, the whole field spectrum is very difficult to define.

Some of the most prominent fields where WSN are being applied today are:

Area monitoring: Both for military and consumer use in intrusion or breach detection in the WSN deployment area allowing for better security of infrastructures or areas, WSN applications can be classified into two categories: monitoring and tracking. Monitoring applications include indoor/outdoor environmental monitoring, health, and wellness monitoring, power monitoring, inventory location monitoring, factory, and process automation, and seismic and structural monitoring. Tracking applications include tracking objects, animals, humans, and vehicles. While there are many different applications, below describe a few example applications that have been deployed and tested in the real environment [12], [14], and [22].

Natural Disaster prevention and detection: Sensor nodes are deployed in a study area and collect data from the surrounding environment, analyzing certain physical values that allow authorities to detect a possible disaster before it occurs or right after it happens, reducing response time and life and property loss. There are ongoing and successfully implemented projects on the detection of fire, floods, landslides, and other natural disasters that may occur, Wireless Sensor Networks are considered to be among the most rapidly evolving technological domains thanks to the numerous benefits that their usage provides.

As a result, from their first appearance until the present day, Wireless Sensor Networks have had a continuously growing range of applications [44].

Healthcare: Medical and healthcare applications are still developing. Patient-related applications consist of either implanted or wearable devices that are in charge of the overall monitoring of patients and their health status by collecting relevant data. WSNs can also be deployed in a hospital allowing for better management of staff, patients, and resources [43].

Environmental monitoring: In [41], the rapid development and miniaturization of sensor devices, and the recent advances in wireless communication and networking technologies, are allowing scientists and engineers to develop networks of small sensors that can be used to continuously monitor the health and stability of the environment we live in.

Industrial monitoring: Industries can implement WSNs for machine health or other production-related monitoring, detecting machine faults, controlling production, and many other factors that are relevant to the industry itself [45].

Smart Home/Office: Sensor nodes deployed on a room or set of rooms can provide information about physical elements such as temperature, luminosity, noise, gas presence, and/or other variables that impact the living or working conditions [42].

2.4 Related Works

The study in [38] presented a dual roadside seismic sensor for the detection of moving vehicles on the roadway by installing them on a road shoulder. The signals of seismic sensors are split into fixed intervals while recording. Various kinds of vehicle characterization information, including vehicle speed, axle spacing, detection of both vehicle axles, and moving direction, can also be extracted from the collected seismic signals as demonstrated. In comparison to existing sensors, this new design of dual seismic sensors is cost-effective, easy to install, and effective in gathering information for traffic management applications. The study error resulted in showing that both vehicle speed and axle spacing detected were less than 20%. In [39] the authors proposed an efficient calibration procedure with accurate results, based on recorded vehicle movement in perspective view.

First, they projected road image using the Direct Linear Transformation (DLT) method, then the vehicle position is detected using the Background Subtraction and tracked using a Mixture of Gaussian (MoG) to determine the vehicle speed. Finally, they developed a prototype of Automated Traffic Flow Monitoring based on Python programming. In the experiment results, the accuracy of vehicle position detection is evaluated based on the Euclidean distance.

The average difference between the results of position detection with ground-truth is 12.07 pixels with a camera angle of 40 degrees. The percentage of speed measurement accuracy using the DLT projection method is 96.14%.

The aim of [40] was to present a video-based vehicle speed measurement system based on a mathematical model using a movement pattern vector as an input variable. The system uses the intrusion line technique to measure the movement pattern vector with low computational complexity. Further, the mathematical model introduced to generate the pdf (probability density function) of a vehicle's speed that improves the speed estimate. As a result, the presented model provides a reliable framework with which to optically measure the speeds of passing vehicles with high accuracy. As a proof of concept, the proposed method was tested on a busy highway under realistic circumstances. The results were validated by a GPS (Global Positioning System)-equipped car and the traffic regulations at the measurement site. The experimental results were promising, with an average error of 1.77 % in challenging scenarios.

Md Mominul Ahsan [25] developed a prototype vehicle speed monitoring system using the accelerometer-based wireless sensor. The basic concept of the system was developing an experimental system to generate random speed data, which can represent vehicle speed on the road, and developing software to monitor and manage the speed data wirelessly.

The functionality of the system has was simulated in a laboratory environment by setting different speed limits for monitoring single or multiple vehicle speed scenarios through appropriate algorithm and code development. After the study, a kind of conclusion made says that sensor-based vehicle speed monitoring system has great potential for monitoring vehicle speed wirelessly.

2.4.1 Framework and Application of Wireless Sensor Network for Car Accident Detection, Prevention and Reporting Systems

The researcher on [25] developed a prototype for the vehicle speed monitoring system using an accelerometer-based wireless sensor. The basic concept of the system is based on the following methodology: developing an experimental system to generate random speed data, which can represent vehicle speed on the road, and developing software to monitor and manage the speed data wirelessly.

A wireless sensor attached with a mechanical wheel measures the acceleration vibration of the system, which is equivalent to wheel speed, and transmits the data wirelessly to a computer. Software (Speed Manage) has been developed using Java Socket programming codes which converts the vibration data to equivalent speed data and presents these in a Graphical User Interface (GUI). If the detected speed is greater than a set speed limit, the data is automatically saved in a central database in the form of an electronic report for taking any further action.

On the other hand, the authors of [26] came up with a framework for automated driving system testable cases and scenarios. The goal of the research was to develop an example of a preliminary test framework for ADS that are in development and may come to market in the near to mid future. The following steps were conducted to support the development of the sample test framework.

Technologies of interest in this work included light-duty automated driving functions that fell within Level 3 (L3) to Level 5 (L5) of the SAE1 levels of driving automation (SAE International, 2016).

The functions were identified based on prototype vehicles and conceptual systems.

In the mountain roads, there are always tight curves and the roads are also narrow. In these kinds of situations, the driver of a vehicle cannot see vehicles coming from the opposite side.

Thousands of people lose their lives each year because of this problem. The Authors of [27] proposed a sensor-based accident prevention system to reduce accidents in the curve roads which is done by alerting the driver by a means of LED light that glows when the vehicle comes from the other side of the curve.

The vehicle is detected with the help of an ultrasonic sensor which is interfaced to the microcontroller Arduino UNO. The design of this system mainly consists of two parts; they are hardware design and software design. Hardware design consists of sensors like the ultrasonic sensor, a microcontroller, and LED. Ultrasonic sensor uses +5V DC supply. Its range is from 2 cm to 100 cm.

Microcontroller Software design is done for sensing the vehicle or obstacle and to operate the LED by using Arduino 1.0.5 IDE tool which is open-source software. Programming can be done by using embedded C or C++. The operating system that we used is windows 8. The LED light here we used is of green color and uses a maximum +5V DC supply.

The authors of [28] presented a novel Internet of Things-based accident detection and reporting system for a smart city environment. The proposed approach aims to take advantage of the advanced specifications of smartphones to design and develop a low-cost solution for enhanced transportation systems that are deployable in legacy vehicles.

In this context, a customized Android application is developed to gather information regarding speed, gravitational force, pressure, sound, and location.

Speed is a factor that is used to help improve the identification of accidents. It arises because of clear differences in environmental conditions (e.g., noise, deceleration rate) that arise in low-speed collisions, versus higher-speed collisions).

The information acquired is further processed to detect road incidents. Furthermore, a navigation system is also developed to report the incident to the nearest hospital.

The proposed approach is validated through simulations and comparison with a real data set of road accidents acquired from the Road Safety Open Repository and shows promising results in terms of accuracy.

The Three Researchers of [28] introduced a new system in automobile technology which is about how to keep a 10-meter distance between one vehicle and another vehicle so that the vehicle doesn't crash or cause any traffic problem. The system aims to prevent accidents mainly due to not knowing the following distance (i.e., 10m) between one vehicle and another vehicle.

The proposed system comprises an idea of having safety while reversing a vehicle, detects any object within the following distance, and displays the distance between one vehicle and another vehicle to the driver using LCD. They have used ultrasonic sensors to detect any vehicle on both the front and backside of our vehicle. This system is also used in a large crane which is mainly operated in the harbor area. If the car reaches 10 meters, the green color light glows. At an 8-meter distance, yellow color light glows showing the warning. When it reaches a 5-meter distance red color light glows and which indicated that the driver is not following the indicators of the roadside.

The distance is also indicated to the vehicle driver. By this proposed system, safety is maintained in crowded areas and in-vehicle reversing process.

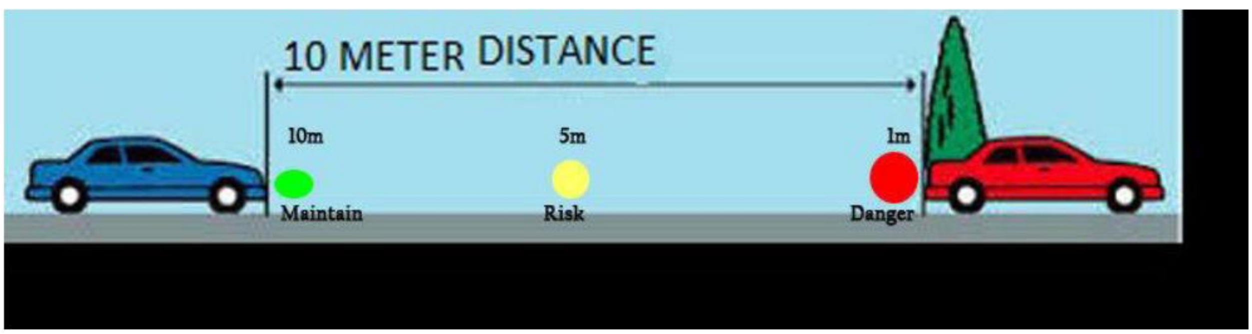

Fig.2.6 Proposed model for vehicle to vehicle proximity detection [29]

Traffic collisions, in particular, high-speed car accidents often result in huge damages, long traffic queues, and loss of human lives. In this work, we present an intelligent modular autonomous system that monitors traffic on highways and alerts drivers of sudden stops, in poor visual conditions [29].

The system is composed of several identical modules, to be placed in the middle of a highway's lane, that sense the lights and communicate their presence and velocity to their neighbor modules via RF.

With such information, the nearby modules estimate the velocity of the passing cars. When the module ahead detects a car passing at a much slower speed than what was previously estimated, it alerts the other modules, so they produce a visual indication for the oncoming drivers, preventing accidents.

They proposed a system that is to be installed on the road and is composed of several modules that work together to perform real-time traffic monitoring and detection of hazardous situations. Each module incorporates very bright LEDs that are activated when a hazardous situation is identified, thus warning drivers approaching the location. One of the key features that differentiate the proposed systems from others is the fact that it is pro-active in detecting accidents by exchanging messages between modules.

Each module measures the time elapsed between the communication of a preceding module and the detection of the vehicle by the sensor. Therefore, knowing the distance between each device it is possible to determine a vehicle's speed, and if it reduced drastically the velocity, or even if it stopped.

2.4.2 Application of Wireless Sensor Network and Object Detection System for Car Accident Prediction, Detection, Prevention and Reporting Systems

The authors of the paper referred to in [30] proposed an automatic car accident detection method based on Cooperative Vehicle Infrastructure Systems (CVIS) and machine vision.

First of all, a novel image dataset CAD-CVIS is established to improve the accuracy of accident detection based on intelligent roadside devices in CVIS. Especially, CAD-CVIS consists of various kinds of accident types, weather conditions, and accident location, which can improve the self-adaptability of accident detection methods among different traffic situations.

Secondly, they developed a deep neural network model YOLO-CA based on CAD-CVIS and deep learning algorithms to detect an accident.

In the model, we utilize Multi-Scale Feature Fusion (MSFF) and loss function with dynamic weights to enhance the performance of detecting small objects. Finally, our experiment study evaluates the performance of YOLO-CA for detecting car accidents, and the results show that our proposed method can detect car accidents in 0.0461 seconds (21.6FPS) with 90.02% average precision (AP).

When YOLO-CA is compared with other object detection models, the results demonstrate the comprehensive performance improvement on the accuracy and real-time over other models.

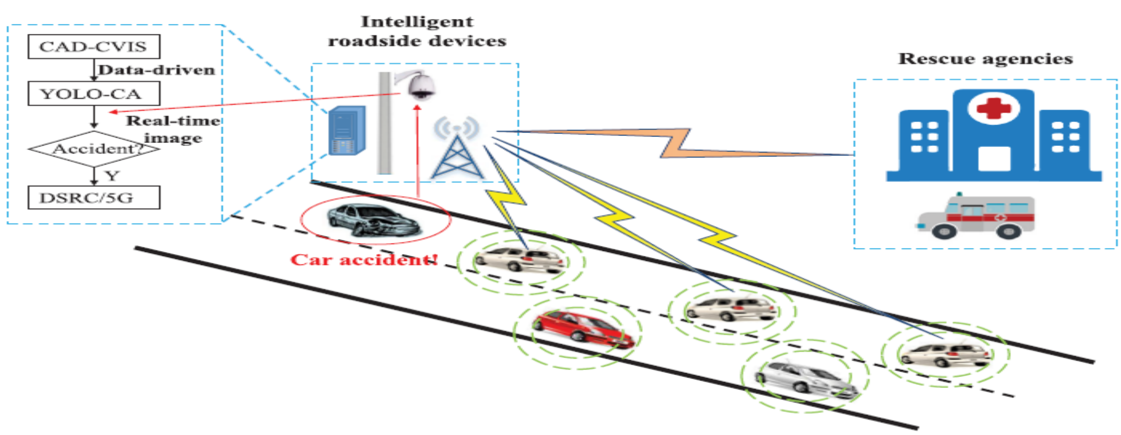

Fig.2.7 The application scenarios of the automatic car accident detection system based on CVIS [30].

The other research work came up with a simple and low-cost system for automatic animal detection on highways for preventing animal and object vehicle collision using image processing and computer vision techniques are presented. A method for finding the distance of the animal in real-world units by the camera-mounted vehicle is also presented.

The presented system is trained on more than 2200 images consisting of positive and negatives images and tested on various video clips of animals on highways with varying vehicle speed. As per the two-second rule, the developed system was able to alert the driver when the vehicle speed is up to 35000 mph. Though the developed system focused on automatic animal detection in context to Indian highways it is supposed to work in other countries also.

The developed system can easily be implemented in any car and can be extended to detect other animals also after proper training and testing to prevent collision of animals with vehicles on the road [31].

Object Detection in wireless sensor networks (WSN) plays an important role in the surveillance application. In existing, a targeting object framework called Face Tracking is used for object detection. A polygon region called face, that is used for is constructing and detecting objects. The nodes that are grouped inside a face can communicate only among them.

Edge recognition calculation is utilized to discover an edge discovery calculation is utilized; in which two hubs are associated has the best inclusion territory. An Optimal selection algorithm is used to select the nodes which can track the target with less energy usage. The target may not be tracked exactly if the target moves out of the coverage area or if the energy of the tracking node becomes low.

To overcome this problem, a tracking scheme, called t-Tracking is proposed to achieve the quality of tracking. Distributed tracking algorithm sends queries about the energy level and coverage area, to all the nodes in the face in which the target has to be tracked next. Based on the reply from all nodes, a node with the best energy level and coverage area node is selected for target tracking. Since the best node is selected, the target can be tracked with accuracy.

Paper [32] proposed a vision-based crash detection framework for a mixed traffic flow environment considering low-visibility conditions. Retinex algorithm was introduced to enhance the image quality of low-visibility conditions, such as night, foggy, and rainy days.

A deep learning model (i.e., Yolo v3) was trained to detect objects in a mixed traffic flow environment and a decision tree model was developed for crash detection, considering various crash scenarios between motorized and non-motorized traffic. e proposed method achieved a hit rate of 92.5% and a false alarm rate of 7.5%. Interesting findings include that the model outperformed empirical rule-based detection models; the image enhancement method can largely improve crash detection performance under low-visibility conditions; and the accuracy of object detection (e.g., bounding boxes prediction) can impact crash detection performance, especially for minor motor-vehicle crashes.

2.5 The Factors that Contribute to Road Traffic Accident

To develop a framework for integration of wireless sensor network and object detection system to monitor Careless driving behavior it's very important to know road traffic rules which are about understanding when it is right or ethical driving behavior and when it becomes illegal (unethical) driving.

According to the authors on [17], driving behavior is aggressive if it is deliberate, likely to increase the risk of collision and is motivated by impatience, annoyance, hostility and/or an attempt to save time". This definition is best suited to the Jordanian case where the attempt to buy time is a reality for most Jordanian drivers. Most drivers included in the survey reported that they are often late for work and they are late for appointments most of the time.

They gave themselves the unjust priority of being before the others at all times because they thought they were the only ones late and the others all had time at their disposal.

However, in most countries speeding or excessive vehicle speed on the road is considered to be the single biggest factor for road accidents contributing to fatal injuries or even death and financial costs to society. According to the article [18], Factors such as inexperience, lack of skill, and risk-taking behaviors have been associated with the collisions of young drivers. In contrast, visual, cognitive, and mobility impairment have been associated with the collisions of older drivers. They investigated the main causes of road accidents by drawing on multiple sources: expert views of police officers, lay views of the driving public, and official road accident records.

The results show that both expert views of police officers and lay views of the driving public closely approximated the typical factors associated with the collisions of young and older drivers, as determined from official accident records.

The results also reveal potential underreporting of factors in existing accident records, identifying possible inadequacies in law enforcement practices for investigating driver distraction, drug and alcohol impairment, and uncorrected or defective eyesight. And also this investigation highlighted a need for accident report forms to be continuously reviewed and updated to ensure that contributing factor lists reflect the full range of factors that contribute to road accidents.

Finally, the views held by police officers and the public on accident causation influenced their memory recall of factors involved in hypothetical scenarios. Delay in completing accident report forms should be minimized, possibly by use of mobile reporting devices at the accident scene.

The study [19] has shown that the magnitude of RTA was found to be 23.17%. According to the drivers' perceived cause of the accident, 22 (38.60%) of the accident was due to a violation of traffic rules and regulations. The majority of the victims were pedestrians, 19 (33.33%). Drivers who were driving a governmental vehicle were 4.16 times more likely to have RTA compared with those who drive private vehicles.

Drivers who used alcohol were 2.29 times more likely to have RTA compared with those drivers who did not consume alcohol. The Magnitude of reported road traffic accidents was high. Violation of traffic laws, lack of vehicle maintenance, and lack of general safety awareness on pedestrians were the dominant reported causes of RTAs.

Research paper [20] is aimed to evaluate Road Traffic accidents (RTA) in Addis Ababa–Adama Expressway which is the first modern expressway in Ethiopia. The study investigated several characteristics of accidents. Among others, it included accident type and severity, time of accident occurrence, and types and level of traffic.

Advanced statistical tests were conducted to identify the contribution of accident instruments. Out of 1,137 RTA datasets, 75% occurred during the daytime and about 24% of accidents occurred at night. The highest fatality accident occurred in January. In total, 84 fatalities, 191 serious injuries, and 438 light injuries were registered. The highest number of serious injuries (about 38%) occurred during the weekend. A total of 1,348 vehicles were involved in road accidents. About 33.5% of accidents occurred due to rear-end collision while about 31% occurred due to vehicle rollover. The remaining accidents occurred due to a collision with guardrail and curbstone about 39% of accidents occurred due to unethical driving behavior while about 19% of accidents occurred due to overspending. Also, driver fatigue, flat tires, poor vehicle brake performance, and steering problems, the presence of animals were contributing factors to the reported accidents.

On the other study [8] that traffic accident kills above 450 people annually in Addis Ababa and the study forecast that Addis Ababa is going to lose above 600 million birrs in next five years if the challenge is not combating. For this problem, speed takes the first role, and the next is a pedestrian fault.

Road damage, drug user drivers, and their ethics have also a big impact on road safety and flow in Addis Ababa. The road carnage has a severe impact on human, social, and economic development. The study finding could stimulate discussion and inform policymakers in traffic safety policy formation.

Education about road use for passengers and drivers, enforcement, differentiating pedestrian road from car road by traffic management are ultimately recommended [21]. Under this subtopic, different road Traffic accident cases are studied through different topics, and in which almost all research authors clearly expressed that driver Behavior has more contribution to RTA which directly emphasizes that studying driving behavior is very important to work reduce Road Traffic accidents.

2.6 Research Gap Analysis

One of the major problems of facilitating the accident reduction which occurs as a result of careless driving is that there has been no means of monitoring every step of driving status. There are so many research works done on this area using wireless sensor network technology.

All or at least nearly all traffic laws are not inclusively studied rather some category of researches focuses on only monitoring and controlling over speed while the other category-focused on monitoring traffic light violation.

Even though the researches that are done on wireless sensor networks to overcome or reduce car accidents are targeting reporting the accident, Predicting the accident, and Warning the driver that the accident may happen almost all focused on unintentional cases of the accident[27][33][34].

Therefore, the research works done on the application of wireless sensor network works on the things that are supposed to case accident like drink and drive focusing on alcohol sensing, bad road conditions where the road is covered by hills and not visible for the driver so that warning systems are made and thirdly, accident reporting systems are made and installed by car just to alarm during the occurrence of an accident which is by sensing the vibration, weight and the like.

The proposed framework for integration of wireless sensor network and object detection system to monitor careless driving is a research idea that has focused on monitoring careless driving behavior. Therefore this research idea included three functions which are, monitoring how the vehicle was being driven, do warnings and lastly, it records reports of what happened and why for further analysis under the traffic management office.

Chapter Three

Research Methodology

3.1 Introduction

The purpose of the research is to discover answers to questions through the application of the scientific procedure. It is particularly useful to clarify understanding of a problem and all methods are valuable if used appropriately in light of the research problem and objectives. This chapter discusses every detail of the research processes, the tools used to achieve the required outcome, means of implementation, and ways of evaluation. Since the question is a design problem where design science research is appropriate and so is conducted [35].

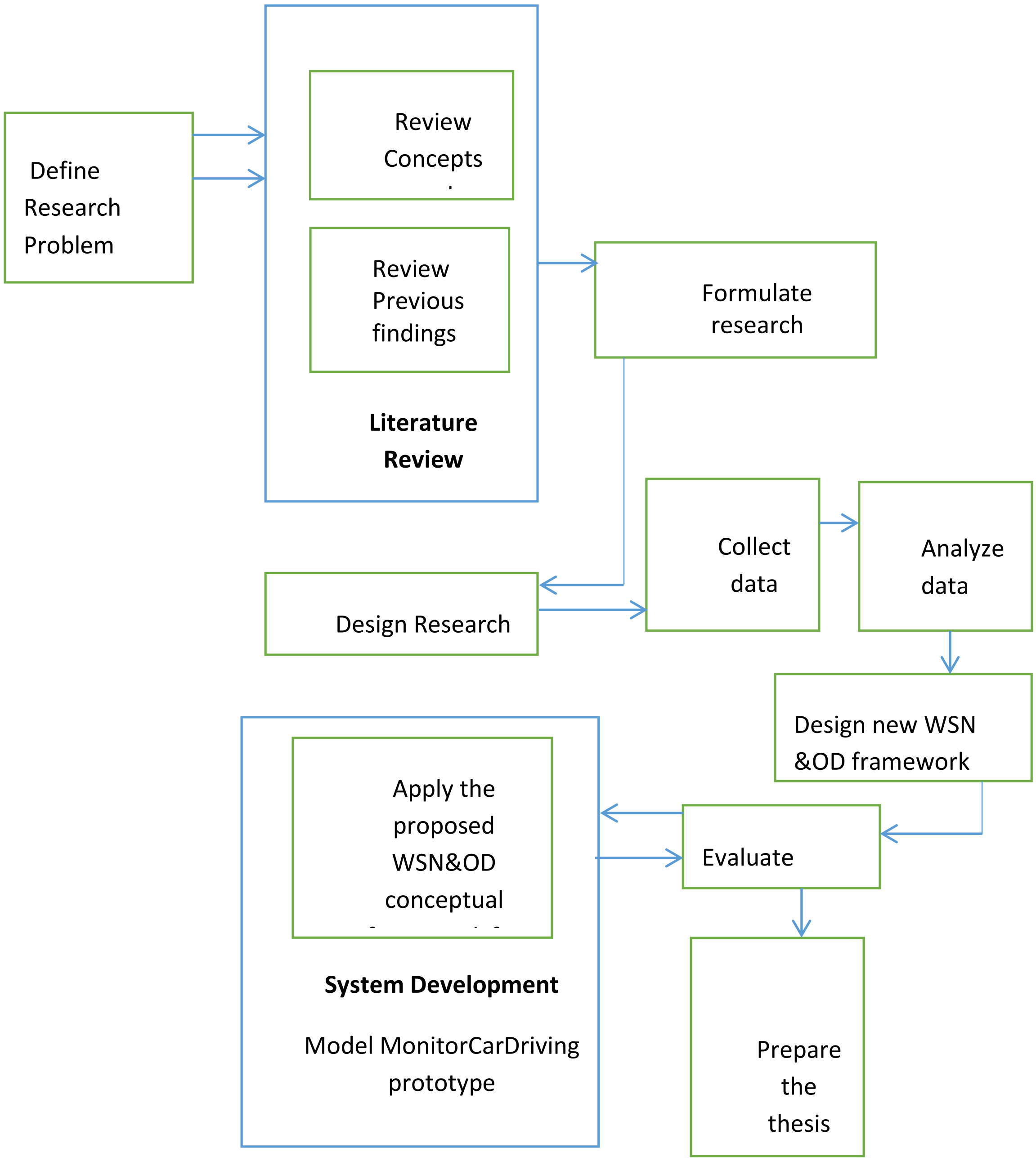

Fig.3.1 Research Process Flowchart (Partially Adapted[35])

The Overview of the process steps are as follows:

A. Identify the problem and Motivation

The study defines the specific research problem and justifies the value of the solution which is done via interviews of concerned stakeholders and with continuous analysis of literature reviews of frameworks on related areas and others.

B. Define the objectives

The researchers infer the objectives of a solution from the definition of the problem and knowledge of what is possible and feasible. The objective of the solution is derived from the problem statement.

C. Design and development

Create an artifact the output of which is an approach for designing the framework for integration of wireless sensor network and object detection system to monitoring careless driving.

D. Demonstration

Demonstrate the use of the artifact to solve one or more instances of the problem. The artifact is demonstrated by applying the proposed approach to the case.

E. Evaluation

Observation and measuring how well the artifact supports the solution to the problem. The evaluation is aimed at comparing the objectives of the solution to the actual results derived in the design and development phase. Experts in this field are interviewed and discussed.

F. Communication

Here, the problem and its importance, the artifact, its utility and novelty, rigor of its design, and effectiveness to researchers and other audiences.

3.2 Software and Hardware Requirements

Software and hardware tools, those that are relevant for development and simulation, have been identified and studied. The following detail describes the software selection part.

TinyOS: is a free and open-source component-based operating system and platform targeting Wireless Sensor Networks (WSNs) and it is an embedded operating system written in the nesC programming language. It is an operating system environment designed to run on embedded devices used in distributed wireless sensor networks [22].

nesC: is a component-based, event-driven programming language used to build applications for the TinyOS platform. It is built as an extension to the C programming language with components "wired" together to run applications on TinyOS [27].

Telosb: Crossbow's TelosB mote is an open-source platform designed to enable cutting-edge experimentation for the research community. The TelosB mote bundles all the essentials for lab studies into a single platform including USB programming capability, an IEEE 802.15.4 radio with integrated antenna, a low-power MCU (Micro Controller Unit) with extended memory, and an optional sensor suite (TPR2420). TPR2400 offers many features, including:-

- IEEE 802.15.4/ZigBee compliant RF transceiver
- 2.4 to 2.4835 GHz, a globally compatible ISM band
- 250 kbps data rate
- Integrated onboard antenna
- 8 MHz TI MSP430 microcontroller with 10kB RAM
- Less Current Consumption
- 1MB external flash for data logging
- Programming and data collection via USB
- Optional sensor suite including integrated light, temperature and humidity sensor (TPR2420)
- Runs TinyOS 1.1.10 or higher

3.3 Network Design, Network Protocol, Communication Layers and Simulation

A commonly used network topology which is Mesh topology due to its multiple advantages is selected for this specific study. Within devices, different communication layers are used and there are different network protocols such as HTTPS, FTP, SMTP, SNMP, and POP3 are applied in a wireless sensor network to do user authentication resource checking and others.

3.4 Design of the Proposed Solution

The proposed research design is a blueprint for conducting research. It has been proposed a framework for monitoring careless driving by integrating wireless sensor networks and object detection systems to be applied on the sensitive roadside (which is listed as a traffic accident hot spot) in Ethiopia together with an LED system to monitor careless driving in that specific area.

The system consists of three types of Sensors namely Speed Sensors, Vibration Sensors, and Object Detecting Sensors. Roadside Camera is about to capture specific scenario as depending on the sensing result when necessary and Light-emitting diodes are used to warn drivers as a result of speed reading from the speed sensor.

3.5 Data Collection and Analysis

Existing research work analysis is of course very relevant to the research title, to express its relation with investigations is done shows and justifies how the investigation helps answer the research question or fill the research gap.

The secondary data sources which are like Conference articles, white papers from selected and mostly recommended websites of reliable authors and organizations, reports, books, and journal articles have been used as a reference to understand wireless sensor network, object detection system, and framework of integration of wireless sensor network and object detection system for careless driving monitoring.

3.5.1 Population and Sample

The study consisted of the selected staff members of the Ethiopian transport authority (ETA), selected taxi drivers, long-distance vehicle drivers, Private Company employed drivers including NGO and Government employed drivers as a sample of the population.

The sample is done on 420 Taxi drivers, 22 Ethiopian Transport authority staff members, 5 private company-employed drivers including own car drivers, 20 government-employed drivers, and 15 long vehicle drivers. The questionnaire is distributed randomly to drivers individually and professionals to gather a good response rate considering their situations well.

The questionnaire is designed according to the software that is known to as the Custom insight software which is found on (http://www.custominsight.com/articles/randomsample-calculator.asp) was used to select sample size with 5% tolerable error level, with 90% of confidence level requires 272 samples but the sample was taken more than that i.e. 370.

The method is called the random sampling technique is used to select the 60 taxi drivers from the individual direction of those who are engaged in Addis Ababa Sub cities. From a selected sample of 420 taxi drivers, 370 of them made the correct response which the overall response rate of taxi drivers achieved in the study was 88.1%. On the Other hand from the questionnaire distributed to 22 Professionals which were the staff members of ETA 20 were correctly responded were the overall response rate of the professionals in the study achieved was 96.7%. This means that from the distributed questionnaire, the response of 55 taxi drivers and 20 professionals of the Ethiopian Transport Authority were used in this research study.

3.5.2 Summary of Responses from the questions

The questionnaire is designed Table 3.1 summary of responses from taxi drivers and long vehicle drivers which is gathered by the means of a questionnaire

Table 3.1 Taxi driver's response summary

No	Questions	Strongly Agree	Agree	Uncertain	Strongly Disagree	Disagree
1	I believe I am concerned about ensuring traffic safety.	72%	13%	9%	4%	2%
2	I used to drink and drive.	25%	53%	2%	8%	12%
3	Always wear a seat belt.	20%	29%	10%	22%	19%
4	Always follow too closely while driving.	29	25%	22%	15%	9%
5	Always watch out for drivers on the road.	29%	28%	9%	12%	22%
6	Follow three-second rules.	19%	21%	30%	22%	18%
7	Used to violate red light	25%	10%	14%	21%	30%
8	Respect and drive within the speed limit.	35%	22%	13%	18%	12%
9	Regularly maintain the vehicle.	10%	12%	32%	29%	17%
10	Always void distraction while driving.	27%	25%	34%	8%	6%

Table 3.2 summarizes the response from Ethiopian Transport Authority staff as professionals of transportation on the questionnaire which is about the means of monitoring driving behavior.

Table 3.2 Summary of ETA Staff

No	Questions	Strongly Agree	Agree	Uncertain	Disagree	Strongly Disagree
1	ETA has a traffic monitoring system.	6%	14%	7%	73%	-
2	Driving behavior monitoring tools are sufficient enough.	-	2%	9%	17%	72%
3	If yes for Question 1, do you think the current system is effective in traffic flow monitoring?	-	15%	6%	79%	-
4	Careless driving is a common cause of traffic accidents?	45%	21%	12%	15%	7%
5	I have frequently noticed the wrong evidence given about what caused an accident.	29%	25%	11%	21%	9%
6	Traffic accident cases are Justified Correctly.	9%	25%	10%	7%	49%
7	I believe I am concerned about ensuring traffic safety	47%	19%	22%	8%	4%

As is read from the above tables more than 85% of respondents said there are very little care and respect to the rules of traffic. The main reason why drivers are not respecting traffic rules is the absence of a monitoring system and carelessness.

In the same search, respondents were asked to record their concerns on the driving behavior monitoring tools used to do monitoring careless driving behavior and 52% recorded that it is due to the lack of tools to justify what caused an accident, while 34% were uncertain. These statistics show that there is a significant barrier in monitoring driving behavior in Ethiopia.

Chapter Four

Design of Proposed Conceptual Framework

4.1 Introduction

Every component of this research work has been developed on the ground truth of individual researches which was used as a reference that ensures that the component already works. Data collection is done by installing MonitorCarDriving software on the computer so that the data is gathered by roadside sensor nodes and feed to the intended computer. The following are different kinds of components that are used to develop the framework for integrating wireless sensor networks and object detection systems to be able to monitor careless driving.

They are: - proximity sensors, speed sensors, object detecting sensors, surveillance cameras, LED lights, and MonitorcarDriving algorithm. The sensor selection is considered the price, environmental condition, and easiness of management.

4.1.1 System Architecture

The sensors detect the vehicles and other things that are supposed to share the road. The sensors estimate the speed of the vehicle and also proximity while driving concerning the rule and ethics of driving. The data collected by the sensor is transferred to the processing unit to compared to the pre-set rule (Ethics) of driving at a given area and if is against the pre-set rule then the collected data is sent to the central control unit through the wireless network.

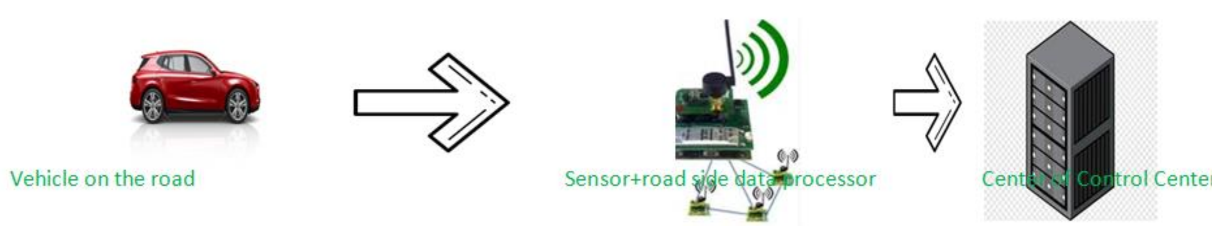

Fig.4.1General System Architecture of the proposed solution.

To check the functionality of the proposed system researchers referred to different systems and researches on the same concept which worked as a functional unit in each of these proposed works [25]. Commercially available sensors are selected under consideration of geographical and climatic conditions.

4.2 Sensor Installation, Connection and Data Generation

Sensors are installed in the form of an electric pole vertically in the order of object detector, speed sensor, proximity sensor from up. The camera and LED can take the upper positions of the poll. This form is chosen due to the idea of saving space, good lookup of the system components, and easy management of in one place.

4.2.1 Required Sensor Nodes

The Number of sensor requirement can be calculated by considering things which are described under the following formula [26].

$$\mathbf{N}=\frac{2.A.\pi}{r^2\sqrt{27}}$$

Where N is the number of sensor Nodes to calculate, A is the area that is supposed to be under monitoring and the effective sensing radius of the sensor node is represented by r.

4.2.2 Management

This part discusses both Node Management and Network Management. The importance of Node management is to improve the nodes' performance, reliability, and stability, and to achieve this following the following steps is a must.

1. Low power use for alarming: - this technique is used to increase task independence which prevents nodes from getting stuck in endless loops when tasks fail to execute due to insufficient available energy.
2. Sleep modes: - Sleep modes include deep sleep, normal and hibernate modes where deep sleep and hibernate modes consumes higher power efficiency than regular sleep because in these states the node components are detached from power sources.

3. Algorithm Integration: - the proposed algorithm activates the Surveillance Camera to capture an LED light for alarming, enables communication during MonitorCarDrivng operations, and finally controls overall sensing progress.
4. Event Priority Management: - this technique enables nodes to respond efficiently to events happening around them based on importance and Priority.

   For example, if a node detects an accident occurred at the same time it needs to access the SD module; then it needs to distinguish which event is more important i.e. which one to prioritize either sending an alarm or save data. I need to instruct the node that when an accident is detected, it needs to prioritize sending the alarm message before saving data into the SD card. This doesn't' mean not saving the data but it's about priority.

   Furthermore, I aim to apply event management techniques to all aspects related to node functions and task administration.

   On the other hand, the network management technique is to provide a wireless sensor network management technique. Its aim is for accessing & storing sensor data, secure transmission, and synchronization of tasks in real-time. To achieve this, the following steps are to be followed.

   Cloud Storage: - Data storage is implemented in a cloud-based network including various host workstations acting together to store information of numerous Wireless sensor networks.

   Encrypted Data Transmission: - Unsecured transmission of data causes unpredictable and unreliable operations in a WSN. Manipulation of data to trick the network into doing something else(e.g. false alarms), overwhelming the network by flooding it with packets in which leads to network failure, stealing data are kinds of common WSN attacks where Encryption of transmission is about to reduce. I aim to implement application and link-level features to improve the security in WSNs.

   Application-level entails communication between nodes inside the node's application, not the network whereas link security refers to encryption of data-payload before transmission and the only way a networked device can make sense of the data is by previous knowledge of the encryption key.

Real-time system: - every networked device including all node components, sensors, and radio transmission is synchronized in real-time. WSNs respond to each command exclusively without interfering with normal network operations. The commands include Retrieving sensor data, changing Operating Channels, Uploading data to web-server, or device maintenance.

This study work aims towards implementing a Wireless Mesh Sensor Network (WMSN). This type of network sustains the uniformity of long-distance communications by breaking connection links into a series of smaller hops, thus intermediate nodes cooperatively make forwarding decisions based on the structure of the network.

We also aim to integrate the Wireless Mesh Sensor Network (WMSN) with an IP network to improve application capabilities in terms of access to live sensor data, web server support, Secure Shell (SSH) connections, and cloud-based storage.

4.3 MonitorCarDriving

Finally, the software is developed using python programming language and python socket programming is used for the proposed system to help manage data in traffic monitoring and controlling. The software is called as MonitorCarDriving stands for Monitoring Careless driving.

Chapter Five

Software Development and Result Evaluation

Its proposed 'MonitorCarDriving' software which stands for Monitoring Careless driving and the software is mainly aimed at monitoring driver's activity relative to the traffic law. First of all MonitorCarDriving software learns traffic laws which were used as a source of knowledge to implement the activity of the driver while driving. Traffic light violation, Over-speeding, protecting priority to the road crossers, accident detection, and Notification are events that MonitorCarDriving records and Notify the concerned body. I used SQLite database with python to manage the whole data of this application. The database mainly contained 9 tables in BCNF or 3.5 NF which are listed below with a description.

Cars: this table held detected and recorded cars by the camera. The car entity is a Unique Identifier (ID), color (Color), license number (License), license image (License-Image), an image of the Car (Car-image), and the number of rules broken (num-rules-broken).

Rules: this table held all the rules, their descriptions, and punishments for breaking that rule.

Camera: camera table contained a unique identifier for the camera (ID), location descriptions (Location) longitude (X-coordinate), and latitude (Y-coordinate), where the camera feeds its data (Feed), and which group the camera belongs to.

Sensors: this table held a unique identifier (ID), type of sensor (Type) location descriptions (Location) longitude (X-coordinate), and the latitude (Y-coordinate), where the sensor feed its data (Feed), and which group the sensor belongs to.

LED: LED table contained a unique identifier for LED (ID), location descriptions (Location) longitude (X-coordinate), and the latitude (Y-coordinate), where the LED is controlled by (Controlled-By), and which group the camera belongs to. LED took the group name of sensors.

Group of Camera: this table held the unique group names of the camera group (Cam-Group)

Group of sensor: this table held the unique group names of the sensor group (Sensor-Group)

Type of Sensor: this table held the unique Type of the Sensor (Sensor-Type)

Violations: this table took all the unique identifiers of the other tables as a foreign key and then creates a semantic record. It did like this; a car with ID X has broken the rule Y at an indicated time which is captured by camera Z.

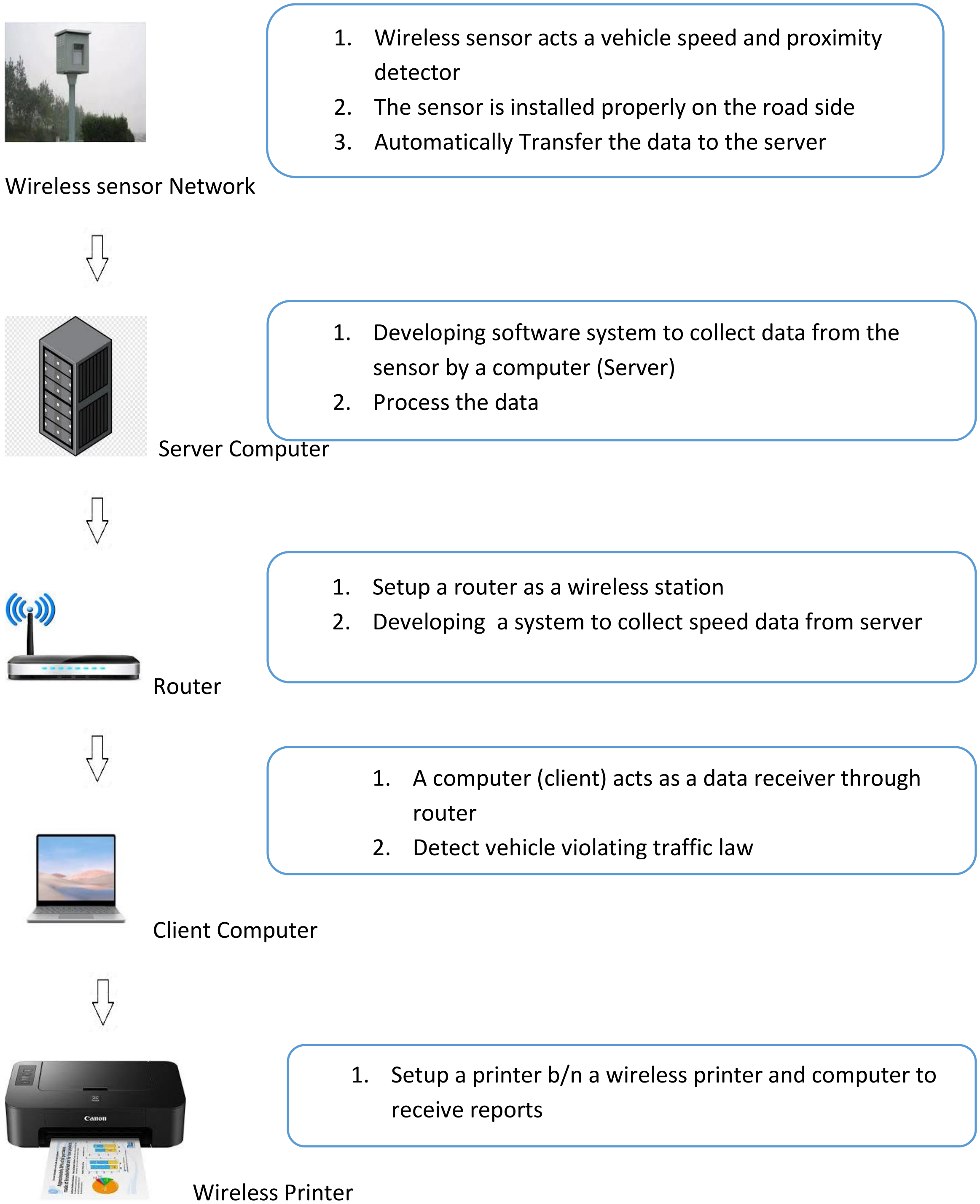

Fig.5.1 System concept for developing MonitorCarDriving system

5.1 Real-Time Object Detection

Creating accurate Machine learning models that are capable of identifying and localizing multiple objects in a single image remained a core challenge in computer vision.[41]

In recent advancements noticed in deep learning, object detection applications are easier to develop than ever before. TensorFlow's object detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train, and deploy object detection models.

5.2 Detection Workflow

Even though every algorithm has a different way of working, but all of them have the same working principle [37]. The features are extracted from the input images at hands and use these features to determine the class of the image. The main purpose of this project is to identify the object type.

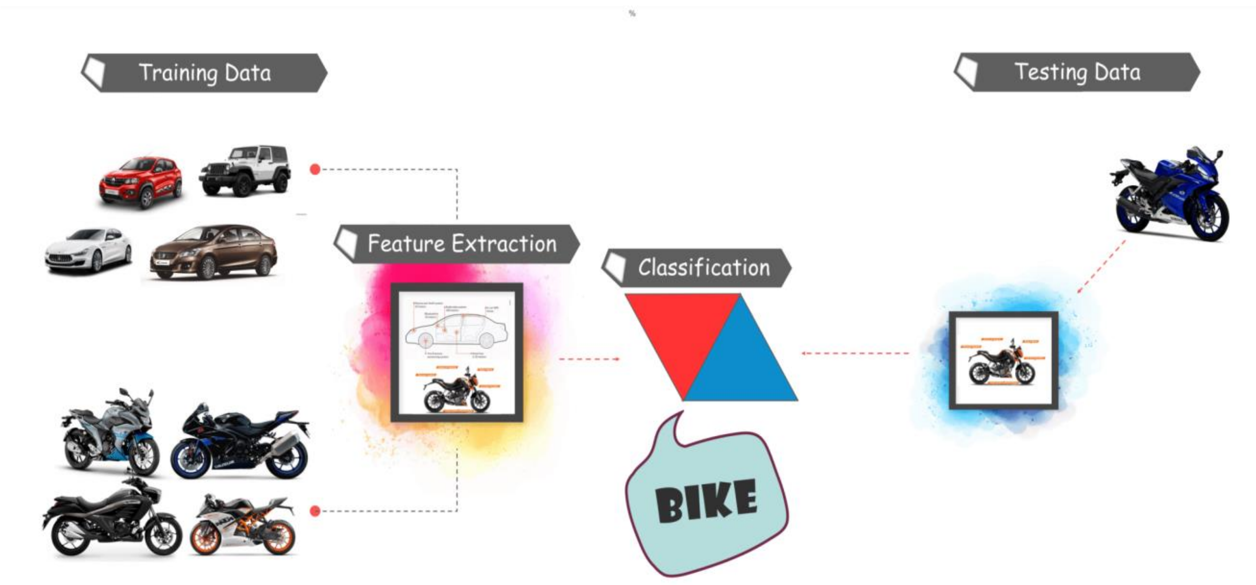

Fig.5.2 Detection of vehicles

All other libraries are downloaded using pip conda by the following code:-

Pip install –user cython

Pip install –user contextlib2

Pip install –user pillow

Pip install –user lxml

Pip install –user matplotlib

Hereafter, I have protobuf, protocol buffers which are google's language-neural, platform-neural, extensible mechanism for serializing structured data which is like XML, but smaller, faster and simpler.

For the sake of simplicity, Models and Protobuf are kept under one folder "TensorFlow"

By using the following command, I run protobuf which is inside the Tensorflow folder.

"path of protobuf's bin"./bin/protoc object_detection/protos/

Whether this worked or not byit cabe be checked by going to the protos folder inside Models>object_detection>protos and there I can see that for every proto file there is one python file created.

After the environment is set up, the next step is creating a new python file inside the object_detection directory. Before all, all Libraries have to be imported and here is how:-

import numpy as np

import six.moves.urllib as urllib

import sys

```
import tarfile
import tensorflow as tf
import zipfile
from collections import defaultdict
from io import stringio
from matplotlib importpyplot as plt from pil import image
sys.path.append(“. .”)
from object_detection.utils import ops as utils_ops
from utils import label_map_util
from utils import visualization_utils as vis_util
```

5.3 Estimating the speed of the Vehicle

Speed estimation is one of the functional requirements of MonitorCarDriving software so here is the speed estimating part of the software.

5.4 Traffic-Signal-Violation-Detection-System

A Computer Vision-based Traffic Signal Violation Detection System from video footage using YOLOv3 & Tkinter.

This is software for the practice of developing a system from completely scratch. Understanding this helps a lot in system development and the basic structure of a system along with computer vision, GUI with python library Tkinter, and basic opencv.

5.5 Monitoring Driver Activity

One of the aims of this research work is to make way to be able to monitor driver behavior so that it is possible to reduce unethical news in the driver’s world. To make it possible, monitoring driver activity is included as a component.

5.6 Lane Detection

Lane detection is a phase that I used as an example to show how the overall proposed solution works. The purpose of Lane Identification is to help identify lanes so that the system considers it as one of the traffic laws and when the lane is not kept while driving the system records it as a Traffic law violation of not keeping lane.

Lines drawn on roads indicate to human drivers where the lanes are and act as a guiding reference to which direction to steer the vehicle accordingly and convention to how vehicle agents interact harmoniously on the road. Likewise, the ability to identify and track lanes is cardinal for developing algorithms for driverless vehicles.

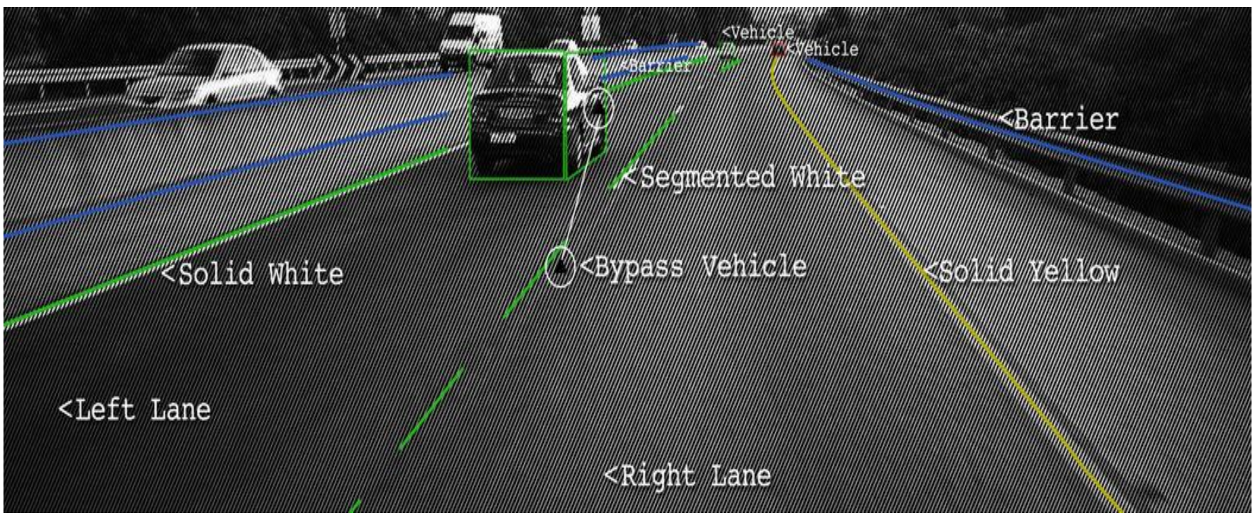

Fig.5.3 Lane detection with OpenCV

5.7 Result Evaluation

The software has been developed in the previous section of this chapter. Now under this subsection of the chapter, the demonstration of the functioning of the system with example scenarios presented. GUI of the software displays vehicle information and generates the report in the event of lane-keeping, speeding limits, and traffic light violation.

5.7.1 Video processing Steps

Almost all steps for video processing follow the same process where more or less the same python tools and libraries are used in algorithm development. In this subsection, full lane detection steps and procedures are shown.

5.7.2 Setting Environment for Video Process

Since I already have installed OpenCV, the next step is to clone the project run so the only thing I have to do is to run from the website link given in https://github.com/chuanenlin/lane-detector.git. Then by opening directory.py with a text editor python code for this section is written.

5.7.3 Next Video Processing

Videos were feed for lane detection as a series of continuous frames (images) by intervals of 10 milliseconds.

```
import cv2 as cv
# The video feed is read in as a VideoCapture object
cap = cv.VideoCapture("input.mp4")
while (cap.isOpened()):
# ret = a boolean return value from getting the frame,
frame = the current frame being projected in the video
ret, frame = cap.read()
# Frames are read by intervals of 10 milliseconds. The programs breaks out of the while loop
when the user presses the 'q' key
if cv.waitKey(10) & 0xFF == ord('q'):
break
# The following frees up resources and closes all windows
cap.release()
cv.destroyAllWindows()
```

5. 7.4 Apply Canny Detector

The Canny detector is a multi-stage algorithm optimized for fast real-time edge detection. Its basic goal is to detect the sharp changes in large gradients (luminosity) like a shift from white to black, then defines them as edges, given a set of the threshold. The following are four parts of the Canny algorithm.

a. Noise reduction

To reduce noise which often leads to false detection a 5x5 Gaussian filter is applied to convolve (smooth) the image to lower the detector's sensitivity to noise. The kernel which is 5x5 of normally distributed numbers to runs across the entire image by setting each pixel value equal to the weighted average of its neighboring pixels.

b. Intensity Gradient

The smoothened image is then applied with a Sobel, Roberts, or Prewitt kernel (where Sobel is used in OpenCV) along the x-axis to detect whether the edges are horizontal, vertical, or diagonal. The following calculates the first derivative of horizontal and vertical directions [21].

$$Edge\_Gradient(G) = \sqrt{G^2 - bG^2}$$

$$Angle = \frac{G_y}{G_x}$$

c. Non-maximum suppression

Non-maximum suppression is applied to "thin" and effectively sharpen the edges. For each pixel, the value check is done if it is a local maximum in the direction of the gradient calculated previously.

d. Hysteresis threshold

In the end, strong pixels are confirmed to be in the final map of edges. Weak pixels should be further analyzed to determine whether it constitutes an edge or noise. Applying two pre-defined minVal and maxVal threshold values, we set that any pixel with intensity gradient higher than maxVal are not edges and discarded.

Pixels with intensity gradient in between minVal and maxVal are only considered edges if they are connected to a pixel with an intensity gradient above maxVal.

Here is the code:-

```
# import cv2 as cv
def do_canny(frame):
# Converts frame to grayscale because we only need the luminance channel for detecting
edges - less computationally expensive
gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
# Applies a 5x5 gaussian blur with deviation of 0 to frame - not mandatory since Canny did
this for me.
blur = cv.GaussianBlur(gray, (5, 5), 0)
# Applies Canny edge detector with minVal of 50 and maxVal of 150
canny = cv.Canny(blur, 50, 150)
return canny
# cap = cv.VideoCapture("input.mp4")
# while (cap.isOpened()):
#  ret, frame = cap.read()
canny = do_canny(frame)
# if cv.waitKey(10) & 0xFF == ord('q'):
#   break
# cap.release()
# cv.destroyAllWindows()
```

The frame is first grayscaled because only luminance is needed for detecting edges and a 5x5 Gaussian blur is applied to decrease noise to reduce false edges.

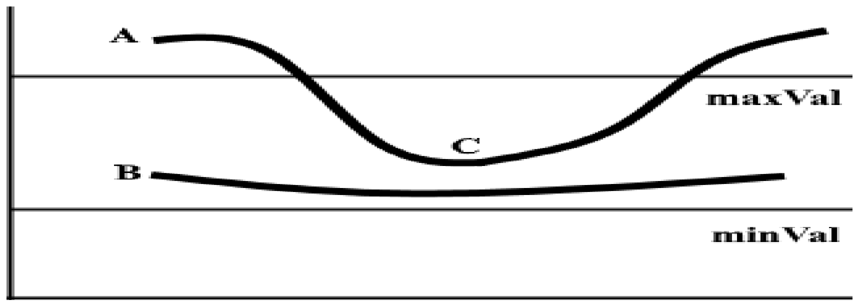

Fig.5.4 Hysteresis threshold on two lines

5.7.5 Segmenting lane Area

By handcrafting a triangular mask to segment the lane area and discard irrelevant areas in the frame to increase the effectiveness of later stages.

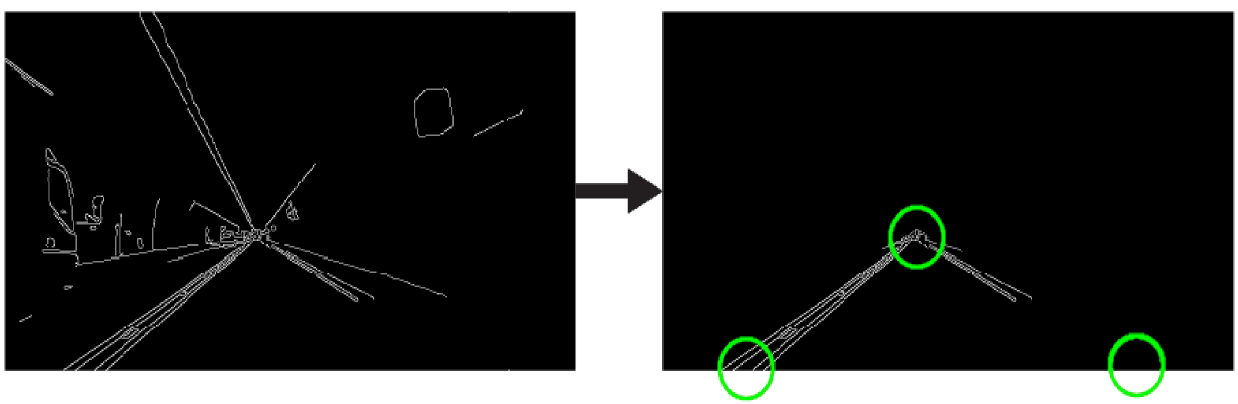

Fig.5.5 the triangular mask is defined by three coordinates which are indicated by green circles.

5.7.6 Hough Transform

In the Cartesian coordinate system, we represent a straight line as y=mx+b by plotting y against x. However, it can be represented as a single point in Hough space by plotting b against m. a line with equation y=2x+1 may be represented (2, 1) in Hough space.

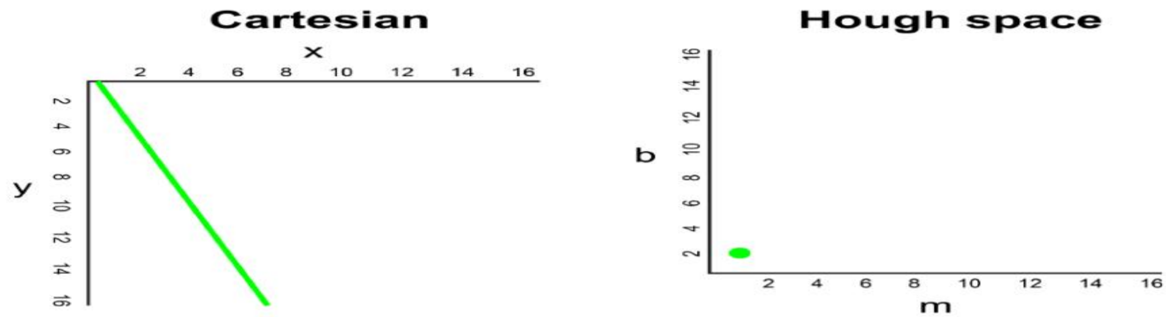

Fig.5.6 Hough Transform

Many possible lines can pass through this point, each line with different values for parameters m and b. To show this, a point at (2,12) can be passed by y=2x+8, y=3x+6, y=4x+4, y=5x+2, y=6x, and so on. These possible lines can be plotted in Hough space as (2,8), (3,6), (4,4), (5,2), (6,0). This produces a line of m against b coordinates in Hough space.

It is possible to find an equation for a series of points in the Cartesian coordinate system which are connected by some line.

This is done by plotting each point in the Cartesian coordinate system to the corresponding line in Hough space and then finding the point of intersection in Hough space where the point of intersection in Hough represents the values of m and b that passes through all points in the series.

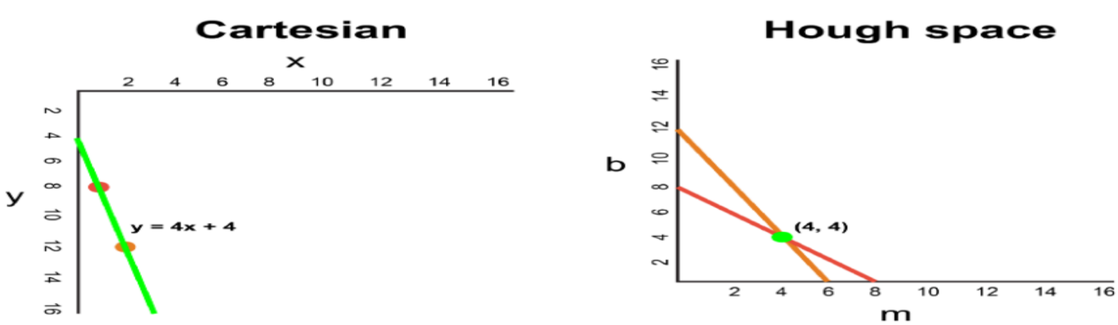

Fig. 5.7 Cartesian coordinate system

The frame can be interpreted simply as a series of white points representing edges in the image space because it is passed through a canny detector. Now the same technique is applied to identify which of these points are connected to the same line and if they are connected, what its equation is so that it's possible to plot this line on the frame.

To correspond to Hough Space, a Cartesian coordinate is used for simplicity of explanation. The Gradient is infinity and cannot be represented in Hough Space when the line is vertical so polar coordinates are used instead which is an almost same process where against θ against r is used instead of m against b.

For example, for polar coordinate system points with x=8 and y=6, x=4 and y=9, x=12 and y=3, it is plotted on the corresponding Hough space.

Here the lines in Hough space intersect at the point $\theta = 0.925$ and $r = 9.6$. Since a line in the Polar coordinate system is given by $r = x\cos\theta + y\sin\theta$, we can induce that a single line crossing through all these points is defined as $9.6 = x\cos 0.925 + y\sin 0.925$.

When more curves intersect in Hough space, it means lines indicated by the intersection correspond to more points. The minimum threshold number of intersections in Hough space to detect a line is defined for implementation. So then, Hough transform keeps track of Hough Intersections of every point in the frame. In case of the number of Intersections exceeding defined, a line with matching θ and r parameters are identified. Finally, the Hough Transform is applied to identify two straight lines which are the four left and right lane boundaries.

5.7.7 Visualization

Every component has an output expected to be visualized at the end same way in this project the lane is visualized as two light green which are linearly fitted polynomials that were overlaid on as output frame which is a different outcome for the other projects.

5.8 Python socket Programs

To establish the connection between server and client computer I used the python socket programming technique.

To act as a server and listen for the incoming messages, or connect to other applications as a client, Sockets are configured. Communication is made bi-directional after the socket gets to connect with both ends of TCP/IP.

5.8.1 Python socket Server

Echo Server

This program receives incoming messages and echoes them back to sender where it starts by creating TCP/IP socket.

import socket

Import sys

#Create a TCP/IP Socket

sock = socket.socket(socket.AF_INET, socket.SOCKET_STREAM)

#bind() is used to associate socket with server address where in this case the address is localhost

#bind the socket to the port

Server address = (‘localhost’, 10000)

Print >>sys.stderr, ‘starting up on %port%’% server_address

Sock.blind(server_address)

Calling **liste()** puts the socket into server mode and **accept()** wats for the incoming connection whereas **accept()** returns an open connection between server and client along client address.

#listen for incoming connection

Sock.listen(1)

While True:

#wait for connection

Print >>sys.stderr, ‘waiting for connecion’

Connection, client_address = sock.accept()

After Connection is Established, data is Read from the connection with a **recv()** and transmitted with a **sendall()**

```
try:
    print >>sys.stderr, 'connection from', client_address
    # Receive the data in small chunks and retransmit it
    while True:
        data = connection.recv(16)
        print >>sys.stderr, 'received "%s"' % data
        if data:
            print >>sys.stderr, 'sending data back to the client'
            connection.sendall(data)
        else:
            print >>sys.stderr, 'no more data from', client_address
            break
finally:
    # Clean up the connection
    connection.close()
```

After communication with client is finished, the connection needs to be cleaned using **close().**

5.8.2 Python socket Client

Echo Client

The client Programs sets up its Socket in differently way from server. Rather than using binding to the port and listening, it uses **connect()** to attack socket directly to the remote address.

```
import socket

import sys

#Create a TCP/IP socket

Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#connect the socket to the port where the server is listening

Server_address = ('localhost', 10000)
```

```
Print >>sys,stderr, ‘connecting to %port%%’server_address

Sock.connect(server_address)
```

Now in same way with server, after Connection is Established, data is Read from the connection with a **recv()** and transmitted with a **sendall().**

```
try:
   # Send data
   message = 'This is the message.  It can be repeated.'
   print >>sys.stderr, 'sending "%s"' % message
   sock.sendall(message)
   # Look for the response
   amount_received = 0
   amount_expected = len(message)
   while amount_received < amount_expected:
      data = sock.recv(16)
      amount_received += len(data)
      print >>sys.stderr, 'received "%s"' % data
finally:
   print >>sys.stderr, 'closing socket'
   sock.close()
```

After the entire message is sent and copies received, the socket is closed to free up the port.

5.8.3 Client and Server Together

The two communicating devices (Client and Server) should be on separate terminal windows and can communicate with each other. Accordingly here is the server output:

```
$ python ./socket_echo_server.py

starting up on localhost port 10000

waiting for connection

connection from (‘127.0.0.1’,52186)
```

received “this is the mess”

sending data back to the client

received “age. It can be”

received “ repeated. ”

sending data back to the client

received “’

no more data drom (‘127.0.0.1’.52186)

waiting for connection

The Client Out Put is the following:

$ python socket-echo_client.py

connecting to localhost port 10000

sending “this is the message. It can be repeated.”

received “this is the mess”

received “age. it can be”

received “ repeated. ”

closing socket()

$

5.9 Evaluation of result and its implication

The objectives have been compared to the research outcome and confirmed that the research questions are answered. About 136 different video data captured from different roads of different geography for testing the system’s functionality with lane detection and traffic light violation detection. Then by directly feeding to MonitorCarDriving software which was installed on the server computer the evaluation of the result was done.

On the other hand, the functionality of the proposed system regarding speed estimation and proximity was done with a shimmer sensor connected to a toy car. The shimmer sensor was attached with a remote control toy car and a wireless connection by Bluetooth is established between the sensor and the laptop as shown in figure 5.7 below.

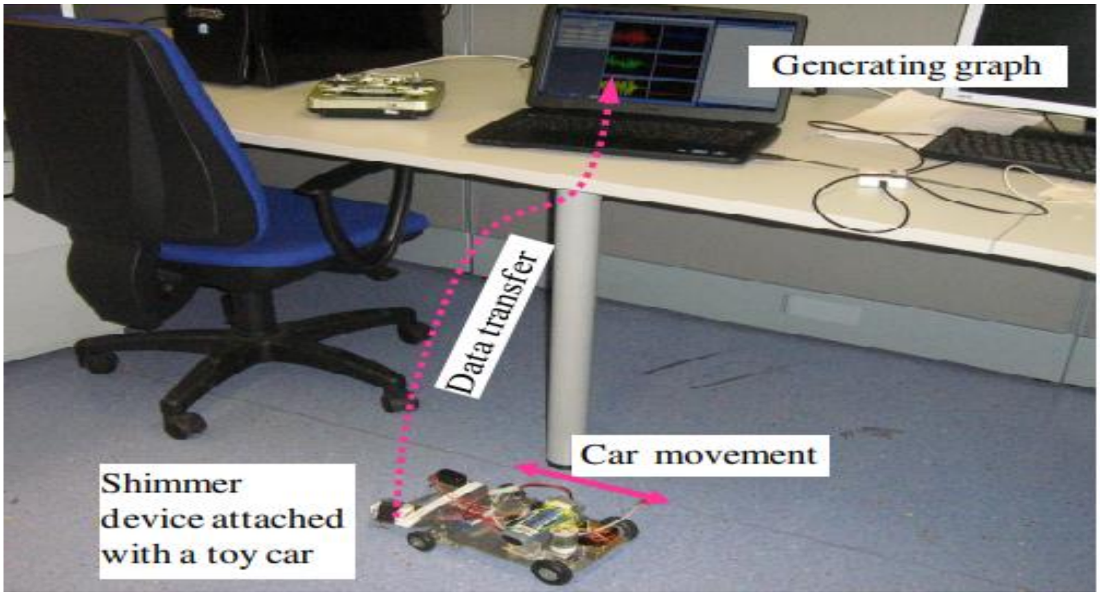

Fig.5.8 shimmer sensor attached with a remote-controlled toy car

Then the toy car is moved by the remote controller and vibration acceleration data was generated due to vibration from toy car movement as a graph.

The generated data (rotational speed data) doesn't represent a real-life vehicle speed hence it has to be converted into linear speed data by using the following formulae.

Speed in (km/Hr.) = $\frac{2\pi N}{60}++\frac{1}{1000}X3600$

Where r is the radius of the wheel and N is the speed of the wheel in rpm.

The system has shown 85% accuracy in lane detection and traffic light violation detection whereas speed estimation has also an average accuracy.

5.9.1 Implementation plan

The study is aimed at reducing accident and done having priority for sensitive areas to traffic accident. Through additional recommendation from concerned (traffic management office) body and deep study on area selection has to be done. All sensor node specifications in the previous section of this study are acceptable. Other components can be gathered according to latest model availability. The implementation time depends on the number of experts to be involved.

Chapter Six

Conclusions, Research Contribution and Recommendation

This chapter is about the general conclusions of the whole project work and thesis contribution. Furthermore, the chapter provides future directions on the use of wireless sensor networks in the intelligent transportation system for monitoring vehicle speed and intelligently managing the relevant data.

6.1 Conclusions

Based on the works carried out in this thesis, the conclusion is made with the following points.

1. An experimental setup has been developed with python to simulate speed estimation, lane detection, traffic light violation, and Object detection. The experiment was done on pre-captured video data. Different Research papers have been reviewed on the performance of wireless sensor networks in related area which proves the functionality of every component in this work.
2. The software so-called MonitorCarDriving has been developed with python socket programming which converts wheel vibration acceleration into speed format automatically, detect and visualize lane violation, detect traffic light violation and monitor priority rules in specific areas through object detection.
3. The software is tested successfully with lane detection video which was taken from a different street, traffic light violation from different location and position, and Object detection which was of people and other animals. The speed estimation was also tested with different speed limits of 40, 60, and 80 KM/Hr.

6.2 Research Contribution

This research work has visible and invisible contributions. Psychologically, whether the monitoring system is actively on work or not, drivers tend to drive under the respect of the traffic law and this I count as an invisible contribution. The following part discusses the visible contributions of this research.

1. The existing state of the art in the application of wireless sensor networks in many different fields of study has been presented in a well-documented form.
2. The novel concept of developing the framework for integrating wireless sensor network to object detection system which is aimed at monitoring careless driving has been proposed.
3. The software has been developed with Python socket programming which is to connect the client to the server, object detection, proximity checkup, lane detection, Traffic light violation, and giving priority when necessary.

6.3 Recommendations for Future Work

According to the WHO Global report of 2018, African countries are exposed to an accident that is related to careless driving. This research work is limited to monitoring Careless driving within the specific area which is sensitive to traffic accident, but the scope needs to be widened to areas which might be an entire city and all express and high ways which are common to car accident due to high traffic of vehicle and traffic laws that allows high speed, but with series care.

Careless driving cannot be expressed with only those things for which this work made solution which is over speeding, traffic light violation, not keeping lane and driving in the proper distance to cars around which are moving. Therefore, it’s better to create systems that monitor driver activities like texting or making the call with a mobile phone while driving, side talk, and driving only with feet.

References

[1] Dastan Bamwesigye and Petra Hlavackova. (2019) Analysis of Sustainable Transport for Smart Cities

[2] Diyar N. Al-Talabani. (2017). Traffic engineering

[3] Samuel Labi, Asif Faiz, Tariq Usman Saeed, Bortiorkor Nii Tsui Alabi and Wubeshet Woldemariam. (2019) Connectivity, Accessibility, and Mobility Relationships in the Context of Low-Volume Road Networks

[4] Kumar, R., P. Paridaa, E. Madhu, and A. Bharat Kumar. (2017) Does Connectivity Index of Transport Network have Impact on Delay for Driver?

[5] Alstadt, B., G. Weisbrod, and D. Cutler. (2012) Relationship of Transportation Access and Connectivity to Local Economic Outcomes: Statistical Analysis

[6] Muhammed Alam, Joaquim Castro Ferreira and Jose Fonseca. (2016) Introduction to Intelligent Transportation Systems

[7] World Health Organization (WHO). (2014) Global Health Estimates.

[8] Fanueal Samson. (2006) Analysis of road traffic accidents in Addis Ababa: Traffic simulation

[9] WHO. (2015) Global status report on road safety, Violence and Injury prevention

[10] World Health Organization. (2002) World report on road traffic injury prevention

[11] Debela Deme MS. (2019) Road Traffic Accident in Ethiopia from 2007/08-2017/18 (Since Ethiopia Millennium

[12] Weng-Fong Cheung, Tzu-Hsuan Lin and Yu-Cheng Lin. (2018) A Real-Time Construction Safety Monitoring System for Hazardous Gas Integrating Wireless Sensor Network and Building Information Modeling Technologies

[13] Abdul Vahab, Maruti S Naik, Prasanna G Raikar, Prasad S R. (2019) Applications of Object Detection System

[14] Yao Xiao. (2019) Vehicle Detection in Deep Learning

[15] Subramanian Arumugam and R. Bhargavi. (2019) A survey on driving behavior analysis in usage-based insurance using big data

[16] Fukai Zhang,Ce Li, and Feng Yang. (2019) Vehicle Detection in Urban Traffic Surveillance Images Based on Convolutional Neural Networks with Feature Concatenation

[17] Mohammad Abojaradeh, P.E., Basim Jrew, Hadeel Al-Ababsah, Ammar Al-Talafeeh. (2014) The Effect of Driver Behavior Mistakes on Traffic Safety

[18] Jonathan J. Rolisona, Shirley Regeva, Salissou Moutarib, Aidan Feeneyc. (2018) what are the factors that contribute to road accidents? An assessment of law enforcement views, ordinary drivers' opinions, and road accident records

[19] Awtachew Berhe Woldu, Abraham Aregay Desta, Tewolde Wubayehu Woldearegay. (2020) Magnitude and determinants of road traffic accidents in Northern Ethiopia: a cross-sectional study

[20] Mikiyas Alemeshet Abera. (2019) Evaluation of road traffic accident in Addis Ababa – Adama Expressway

[21] Teferi Girma Bekele. (2019) Road Traffic Accident Cause and Effect on Socio Economy of Addis Ababa City

[22] Mubarek Dubiso Ofolcha. (2019) Camera Based Flood Detection and Recognition with Wireless Sensor Network; In Case Of Awash Fluvial

[23] Dionisis Kandris, Christos Nakas, Dimitrios Vomvas, and Grigorios Koulouras. (2020) Applications of Wireless Sensor Networks: An Up-to-Date Survey

[24] João Teixeira. (2017) Wireless Sensor Network for Forest Fire Detection

[25] Md Mominul Ahsan. (2014) Development of a Prototype Wireless Vehicle Speed Monitoring System

[26] Eric Thorn, Shawn Kimmel, Michelle Chaka. (2018) A Framework for Automated Driving System Testable Cases and Scenarios

[27] Maminul Islam, Rabiul Hasan, Imran Chowdhury, Towhid Chowdhury. (2013) Internet of Car: Accident Sensing, Indication and Safety with Alert system

[28] M. Prabha, M. Seema, P. Saraswathi. (2015) Distance based Accident Avoidance System using Arduino

[29] J. Oliveiraa, J. Soaresa, A. R. Lourencoa, R. P. Duartec. (2017) Intelligent Sensors for Real-Time Hazard Detection and Visual Indication on Highways

[30]. Daxin Tian, Chuang Zhang, Xuting Duan, and Xixian Wang. (2019) An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems

[31] R.Pavithra, M.Jagadeesan. (2018) Automatic Object Detection and Collision Avoidance System using Cascaded Random Classifier

[32] Chen Wang, Yulu Dai, Wei Zhou, and Yifei Geng. (2019) A Vision-Based Video Crash Detection Framework for Mixed Traffic Flow Environment Considering Low-Visibility Condition

[33] Aravinda, Chaithralakshmi, Deeksha, Ashutha K. (2016) Sensor Based Accident Prevention System

[34] Fizzah Bhatti, Muna Ali Shah, Carsten Maple, and Saif Ul Islam. (2013) A Novel Internet of Things-Enabled Accident Detection and Reporting System for Smart City Environments

[35] Betelhem Belete G/Hiwot. (2017) Conceptual Security Framework for Mobile Banking Key Authentication and Message Exchange Protocols: Case of Ethiopian Banks

[36] João Teixeira. (2017) Wireless Sensor Network for Forest Fire Detection.

[37] Longhai Yang, Xiqiao Zhang, Xiaoyan Zhu, Yule Luo, and Yi Luo. (2019) Research on Risky Driving Behavior of Novice Drivers

[38] Hua Wang, Wei Quan 1, YinhaiWang 2 and Gregory R. Miller. (2014) Dual Roadside Seismic Sensor for Moving Road Vehicle Detection and Characterization

[39] Alexander A S Gunawana, Deasy Aprilia Tanjung, Fergyanto E. Gunawan. (2019) Detection of Vehicle Position and Speed using Camera Calibration and Image Projection Methods

[40] Saleh Javadi Mattias Dahl, Mats I, Pettersson. (2019) Vehicle speed measurement model for video-based systems

[41] Ashenafi Lambebo, Sasan Haghani. (2014) A Wireless Sensor Network for Environmental Monitoring of Greenhouse Gases

[42] Oktay Cetinkaya, Prof. Dr. Ozgur Baris Akan.(2016) Use of Wireless Sensor Networks in Smart Homes

[43] Ramgopal Kashyap. (2020) Applications of Wireless Sensor Networks in Healthcare

[44] Ahsan Adeel, Mandar Gogate, Saadullah Farooq, Cosimo Ieracitano, Kia Dashtipour, Hadi Larijani, Amir Hussain. (2019) A Survey on the Role of Wireless Sensor Networks and IoT in Disaster Management

[45] ()Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches

Appendix

Estimating the speed of the Vehicle

Speed estimation is one of the functional requirements of MonitorCarDriving software so here is the speed estimating part of the software.

TrackableObject

open up the trackableobject.py file and insert the following code:

OpenCV Vehicle Detection, Tracking, and Speed Estimation

```
# import the necessary packages

import numpy as np

class TrackableObject:

def __init__(self, objectID, centroid):

# store the object ID, then initialize a list of centroids

# store the object ID, then initialize a list of centroids

# using the current centroid

self.objectID = objectID

self.centroids = [centroid]

# initialize a dictionaries to store the timestamp and

# position of the object at various points

self.timestamp = {"A": 0, "B": 0, "C": 0, "D": 0}

self.position = {"A": None, "B": None, "C": None, "D": None}
```

```
self.lastPoint = False

# initialize the object speeds in MPH and KMPH

self.speedMPH = None

self.speedKMPH = None

# initialize two booleans, (1) used to indicate if the

# object's speed has already been estimated or not, and (2)

# used to indidicate if the object's speed has been logged or

# not

self.estimated = False

self.logged = False

# initialize the direction of the object

self.direction = None
```

Python code for Lane area Segmenting

```
# import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
# def do_canny(frame):
#     gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
#     blur = cv.GaussianBlur(gray, (5, 5), 0)
#     canny = cv.Canny(blur, 50, 150)
#     return canny
def do_segment(frame):
# Since an image is a multi-directional array containing the relative intensities of each pixel in the image, we can use frame.shape to return a tuple: [number of rows, number of columns, number of channels] of the dimensions of the frame
```

frame.shape[0] give us the number of rows of pixels the frame has. Since height begins from 0 at the top, the y-coordinate of the bottom of the frame is its height

height = frame.shape[0]

Creates a triangular polygon for the mask defined by three (x, y) coordinates

polygons = np.array([

[(0, height), (800, height), (380, 290)]

])

Creates an image filled with zero intensities with the same dimensions as the frame

mask = np.zeros_like(frame)

Allows the mask to be filled with values of 1 and the other areas to be filled with values of 0

cv.fillPoly(mask, polygons, 255)

A bitwise and operation between the mask and frame keeps only the triangular area of the frame

segment = cv.bitwise_and(frame, mask)

return segment

cap = cv.VideoCapture("input.mp4")

while (cap.isOpened()):

ret, frame = cap.read()

canny = do_canny(frame)

First, visualize the frame to figure out the three coordinates defining the triangular mask

plt.imshow(frame)

plt.show()

segment = do_segment(canny)

if cv.waitKey(10) & 0xFF == ord('q'):

break

cap.release()

cv.destroyAllWindows()

The Hough Transform is applied to identify two straight lines which are the four left and right lane boundaries.# import cv2 as cv

import numpy as np

```
# # import matplotlib.pyplot as plt
# def do_canny(frame):
#     gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
#     blur = cv.GaussianBlur(gray, (5, 5), 0)
#     canny = cv.Canny(blur, 50, 150)
#     return canny
# def do_segment(frame):
#     height = frame.shape[0]
#     polygons = np.array([
#                           [(0, height), (800, height), (380, 290)]
#                       ])
#     mask = np.zeros_like(frame)
#     cv.fillPoly(mask, polygons, 255)
#     segment = cv.bitwise_and(frame, mask)
#     return segment
# cap = cv.VideoCapture("input.mp4")
# while (cap.isOpened()):
#     ret, frame = cap.read()
#     canny = do_canny(frame)
#     # plt.imshow(frame)
#     # plt.show()
#     segment = do_segment(canny)
   # cv.HoughLinesP(frame, distance resolution of accumulator in pixels (larger = less precision), angle resolution of accumulator in radians (larger = less precision), threshold of minimum number of intersections, empty placeholder array, minimum length of line in pixels, maximum distance in pixels between disconnected lines)
   hough = cv.HoughLinesP(segment, 2, np.pi / 180, 100, np.array([]), minLineLength = 100, maxLineGap = 50)
#     if cv.waitKey(10) & 0xFF == ord('q'):
#         break
# cap.release()
```

```
# cv.destroyAllWindows()
```

Visualization

```
# import cv2 as cv
# import numpy as np
# # import matplotlib.pyplot as plt
# def do_canny(frame):
#     gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
#     blur = cv.GaussianBlur(gray, (5, 5), 0)
#     canny = cv.Canny(blur, 50, 150)
#     return canny
# def do_segment(frame):
#     height = frame.shape[0]
#     polygons = np.array([
#                             [(0, height), (800, height), (380, 290)]
#                         ])
#     mask = np.zeros_like(frame)
#     cv.fillPoly(mask, polygons, 255)
#     segment = cv.bitwise_and(frame, mask)
#     return segment
def calculate_lines(frame, lines):
    # Empty arrays to store the coordinates of the left and right lines
    left = []
    right = []
    # Loops through every detected line
    for line in lines:
        # Reshapes line from 2D array to 1D array
        x1, y1, x2, y2 = line.reshape(4)
        # Fits a linear polynomial to the x and y coordinates and returns a vector of coefficients
which describe the slope and y-intercept
        parameters = np.polyfit((x1, x2), (y1, y2), 1)
```

```
        slope = parameters[0]
        y_intercept = parameters[1]
        # If slope is negative, the line is to the left of the lane, and otherwise, the line is to the right of the lane
        if slope < 0:
            left.append((slope, y_intercept))
        else:
            right.append((slope, y_intercept))
    # Averages out all the values for left and right into a single slope and y-intercept value for each line
    left_avg = np.average(left, axis = 0)
    right_avg = np.average(right, axis = 0)
    # Calculates the x1, y1, x2, y2 coordinates for the left and right lines
    left_line = calculate_coordinates(frame, left_avg)
    right_line = calculate_coordinates(frame, right_avg)
    return np.array([left_line, right_line])
def calculate_coordinates(frame, parameters):
    slope, intercept = parameters
    # Sets initial y-coordinate as height from top down (bottom of the frame)
    y1 = frame.shape[0]
    # Sets final y-coordinate as 150 above the bottom of the frame
    y2 = int(y1 - 150)
    # Sets initial x-coordinate as (y1 - b) / m since y1 = mx1 + b
    x1 = int((y1 - intercept) / slope)
    # Sets final x-coordinate as (y2 - b) / m since y2 = mx2 + b
    x2 = int((y2 - intercept) / slope)
    return np.array([x1, y1, x2, y2])
def visualize_lines(frame, lines):
    # Creates an image filled with zero intensities with the same dimensions as the frame
    lines_visualize = np.zeros_like(frame)
    # Checks if any lines are detected
```

```
    if lines is not None:
        for x1, y1, x2, y2 in lines:
            # Draws lines between two coordinates with green color and 5 thickness
            cv.line(lines_visualize, (x1, y1), (x2, y2), (0, 255, 0), 5)
    return lines_visualize
# cap = cv.VideoCapture("input.mp4")
# while (cap.isOpened()):
#     ret, frame = cap.read()
#     canny = do_canny(frame)
#     # plt.imshow(frame)
#     # plt.show()
#     segment = do_segment(canny)
#     hough = cv.HoughLinesP(segment, 2, np.pi / 180, 100, np.array([]), minLineLength = 100, maxLineGap = 50)
    # Averages multiple detected lines from hough into one line for left border of lane and one line for right border of lane
    lines = calculate_lines(frame, hough)
    # Visualizes the lines
    lines_visualize = visualize_lines(frame, lines)


    # Overlays lines on frame by taking their weighted sums and adding an arbitrary scalar value of 1 as the gamma argument
    output = cv.addWeighted(frame, 0.9, lines_visualize, 1, 1)
    # Opens a new window and displays the output frame
    cv.imshow("output", output)
#     if cv.waitKey(10) & 0xFF == ord('q'):
#         break
# cap.release()
# cv.destroyAllWindows()
```

Here is the code in python detector.py to test lane detector.

```
import cv2 as cv
```

```
import numpy as np
# import matplotlib.pyplot as plt
def do_canny(frame):
    # Converts frame to grayscale because we only need the luminance channel for detecting edges - less computationally expensive
    gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
    # Applies a 5x5 Gaussian blur with deviation of 0 to frame - not mandatory since Canny again did this for me.
    blur = cv.GaussianBlur(gray, (5, 5), 0)
    # Applies Canny edge detector with minVal of 50 and maxVal of 150
    canny = cv.Canny(blur, 50, 150)
    return canny
def do_segment(frame):
    # Since an image is a multi-directional array containing the relative intensities of each pixel in the image, we can use the frame.shape to return a tuple: [number of rows, number of columns, number of channels] of the dimensions of the frame
    # frame.shape[0] give us the number of rows of pixels the frame has. Since height begins from 0 at the top, the y-coordinate of the bottom of the frame is its height
    height = frame.shape[0]
    # Creates a triangular polygon for the mask defined by three (x, y) coordinates
    polygons = np.array([
                            [(0, height), (800, height), (380, 290)]
                        ])
    # Creates an image filled with zero intensities with the same dimensions as the frame
    mask = np.zeros_like(frame)
    # Allows the mask to be filled with values of 1 and the other areas to be filled with values of 0
    cv.fillPoly(mask, polygons, 255)
    # A bitwise and operation between the mask and frame keeps only the triangular area of the frame
    segment = cv.bitwise_and(frame, mask)
    return segment
```

```
def calculate_lines(frame, lines):
    # Empty arrays to store the coordinates of the left and right lines
    left = []
    right = []
    # Loops through every detected line
    for line in lines:
        # Reshapes line from 2D array to 1D array
        x1, y1, x2, y2 = line.reshape(4)
        # Fits a linear polynomial to the x and y coordinates and returns a vector of coefficients
which describe the slope and y-intercept
        parameters = np.polyfit((x1, x2), (y1, y2), 1)
        slope = parameters[0]
        y_intercept = parameters[1]
        # If slope is negative, the line is to the left of the lane, and otherwise, the line is to the
right of the lane
        if slope < 0:
            left.append((slope, y_intercept))
        else:
            right.append((slope, y_intercept))
    # Averages out all the values for left and right into a single slope and y-intercept value for
each line
    left_avg = np.average(left, axis = 0)
    right_avg = np.average(right, axis = 0)
    # Calculates the x1, y1, x2, y2 coordinates for the left and right lines
    left_line = calculate_coordinates(frame, left_avg)
    right_line = calculate_coordinates(frame, right_avg)
    return np.array([left_line, right_line])
def calculate_coordinates(frame, parameters):
    slope, intercept = parameters
    # Sets initial y-coordinate as height from top down (bottom of the frame)
    y1 = frame.shape[0]
```

```
    # Sets final y-coordinate as 150 above the bottom of the frame
    y2 = int(y1 - 150)
    # Sets initial x-coordinate as (y1 - b) / m since y1 = mx1 + b
    x1 = int((y1 - intercept) / slope)
    # Sets final x-coordinate as (y2 - b) / m since y2 = mx2 + b
    x2 = int((y2 - intercept) / slope)
    return np.array([x1, y1, x2, y2])
def visualize_lines(frame, lines):
    # Creates an image filled with zero intensities with the same dimensions as the frame
    lines_visualize = np.zeros_like(frame)
    # Checks if any lines are detected
    if lines is not None:
        for x1, y1, x2, y2 in lines:
            # Draws lines between two coordinates with green color and 5 thickness
            cv.line(lines_visualize, (x1, y1), (x2, y2), (0, 255, 0), 5)
    return lines_visualize
# The video feed is read in as a VideoCapture object
cap = cv.VideoCapture("input.mp4")
while (cap.isOpened()):
    # ret = a boolean return value from getting the frame, frame = the current frame being
projected in the video
    ret, frame = cap.read()
    canny = do_canny(frame)
    cv.imshow("canny", canny)
    # plt.imshow(frame)
    # plt.show()
    segment = do_segment(canny)
    hough = cv.HoughLinesP(segment, 2, np.pi / 180, 100, np.array([]), minLineLength = 100,
maxLineGap = 50)
    # Averages multiple detected lines from hough into one line for left border of lane and one
line for right border of lane
```

```python
    lines = calculate_lines(frame, hough)
    # Visualizes the lines
    lines_visualize = visualize_lines(frame, lines)
    cv.imshow("hough", lines_visualize)
    # Overlays lines on frame by taking their weighted sums and adding an arbitrary scalar value of 1 as the gamma argument
    output = cv.addWeighted(frame, 0.9, lines_visualize, 1, 1)
    # Opens a new window and displays the output frame
    cv.imshow("output", output)
    # Frames are read by intervals of 10 milliseconds. The programs breaks out of the while loop when the user presses the 'q' key
    if cv.waitKey(10) & 0xFF == ord('q'):
        break
# The following frees up resources and closes all windows
cap.release()
cv.destroyAllWindows()
```