

VEHICLE COUNTING AND SPEED SENSING USING WIRELESS SIGNALS

Submitted in partial fulfillment of the requirements for the award of

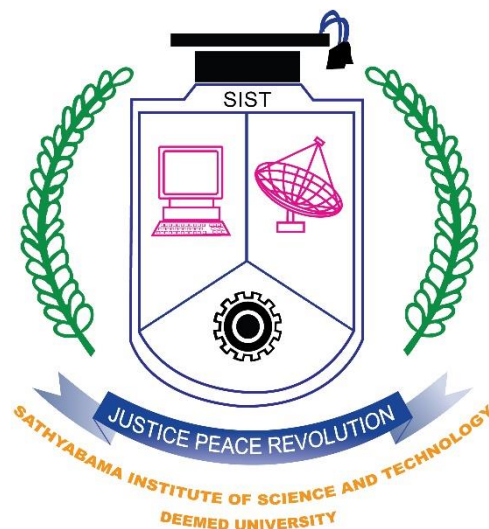
Bachelor of Engineering Degree in

Electrical and Electronics Engineering

by

RAGHUL.S (Register No. 39140044)

HAMEED FAZIL.I (Register No. 39140016)



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAGHUL.S**(Reg. No.39140044) and **HAMEED FAZIL.I**(Reg. No. 39140016) who carried out the project entitled "**VEHICLE COUNTING AND SPEED SENSING USING WIRELESS SIGNALS**" under our supervision from November 2022 to April 2023.

Dr.P.SIVAGANAP

Internal Guide

Dr.V.SIVACHIDAMBARANATHAN

Head of the Department

Submitted for Viva voce Examination held on 26/04/2023

Name: D. S. RADHIM

Signature: Radhi

Internal Examiner

Dr. V. Balasubramanian

Bal

External Examiner

DECLARATION

We **RAGHUL.S** (Reg. No.39140044) and **HAMEED FAZIL.I**(Reg. No.39140016) hereby declare that the Project Report entitled "**VEHICLE COUNTING AND SPEED SENSING USING WIRELESS SIGNALS**" done by us under the guidance of **Dr.P.SIVAGAMI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electrical and Electronics Engineering.

1. 

2. I. Hameed Fazil

DATE: 19/04/2023

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. Nandhita, Dean, School of Electrical and Electronics Engineering** and **Dr. V. Sivachidambaranathan, Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.P.SIVAGAMI** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electrical and Electronics Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Traffic monitoring is key to modern city or smart city planning. However, road accidents and traffic violations have been very common in present world. The detection and counting of vehicles in the highway management are becoming crucial. However, accurately detecting vehicles of varying sizes remains a challenge that affects vehicle counting precision. Vehicle counting and speed sensing are important factor in traffic management systems, allowing authorities to monitor and optimize traffic flow. Traditional methods of vehicle counting and speed sensing, such as manual counting and radar-based speed detection is costly and inefficient. Now a days, the use of sophisticated sensing tools, like cameras, radar, and sensors emphasise to precisely count vehicles and gauge their speed. The proposed system utilizes deep learning techniques for vehicle detection and tracking, which can detect and track vehicles of varying sizes and shapes, regardless of lighting conditions. The network YOLOv3 is utilized to detect the vehicle's type and location. The Computer Vision Algorithm is used to obtain the vehicle's trajectories, allowing for the determination of the driving direction and the counting of different vehicles. Furthermore, the novel strategy described in this paper performs exceptionally well in determining the driving direction and counting vehicles. This proffered methodology for managing and controlling highway scenes finds its application in intelligent transportation systems, such as traffic management, congestion control.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	ix
1	INTRODUCTION	
	1.1 GENERAL INTRODUCTION	1
	1.2 MOTIVATION AND OBJECTIVE	3
	1.3 GLOBAL APPLICATION	4
2	LITERATURE SURVEY	
	2.1 MAJOR FINDINGS FROM LITERATURE SURVEY	5
	2.2 PROBLEM STATEMENT	8
	2.3 EXISTING BLOCK DIAGRAM AND EXPLANATION	10
	2.4 DISADVANTAGE OF EXISTING SYSTEM	11
	2.5 STANDARDS	12
	2.6 CONSTRAINTS AND TRADEOFFS	15
3	AIM AND SCOPE OF PRESENT INVESTIGATION	
	3.1 INTRODUCTION	16

3.2	SCOPE OF THE PROJECT	16
3.3	METHODOLOGY	17
3.4	PROPOSED BLOCK DIAGRAM AND EXPLANATION	19
3.5	ADVANTAGE OF PROPOSED SYSTEM	21
4	HARDWARE DISCRIPTION	
4.1	INTRODUCTION	23
4.2	HARDWARE COMPONENTS	23
4.2.1	ARDUINO UNO	24
4.2.2	IR SENSOR	24
4.2.3	LCD DISPLAY	25
4.2.4	HIGH-DEFINITION CAMERA	25
4.2.5	UART COMMUNICATION	26
5	RESULTS / DISCUSSION AND PERFORMANCE	
	ANALYSIS	
5.1	SIMULATION ANALYSIS	27
5.1.1	SYSTEM DEVELOPMENT	27
5.1.2	BASIC ARCHITECTURE	28
5.1.3	WORKING	29
5.1.4	SIMULATION OUTPUT	32
5.2	HARDWARE KIT AND RESULTS	33

SUMMARY AND CONCLUSION

6.1 CONCLUSION 37

6.2 SCOPE OF FUTURE WORK 37

REFERENCES 39

APPENDIX SOURCE CODE 41

LIST OF FIGURES

FIGURE	TITLE	PAGE NO
2.1	Existing block diagram	10
3.1	Proposed block diagram	19
3.2	Proposed block diagram receiver side	20
5.1	Simulation Process	28
5.2	Arduino ide Setup	29
5.3	Proteus setup	30
5.4	Arduino code compilation	31
5.5	Proteus circuit setup	32
5.6	Output Simulation	43
5.7	Hardware Kit	34
5.8	Vehicle speed monitoring system user page	35
5.9	Result of vehicle speed and traffic prediction System	36
5.10	Vehicle history page	36

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Vehicle counting, speed sensing, and traffic prediction are all important areas of study in the field of transportation engineering. These methodologies are used to collect data on the number of vehicles on a roadway, their speed, and other important metrics such as vehicle type and color. This information is not only used in developing effective traffic management strategies, but also helps to plan roadway expansions or improvements. It also provides better understanding of traffic patterns. Vehicle counting is the process of identifying and counting the number of vehicles passing through a specific area or a road segment. There are several methods available for vehicle counting. Few methods available among them are video-based counting, RFID and sensor-based. Video-based counting involves using cameras to capture video footage of the road segment. The footage is then analyzed using image processing techniques to identify and count the number of vehicles passing through the road segment. Video-based counting is more accurate than manual counting, but it requires high-quality cameras, processing power, and expertise in image processing. Sensor-based counting involves using sensors placed on the road surface to detect the presence of vehicles. The sensors can be magnetic, infrared, or acoustic. The sensors detect the vehicles' presence and send signals to a central processing unit, which counts the number of vehicles passing through the road segment. Sensor-based counting is highly accurate, but it requires expensive equipment and regular maintenance.

Speed sensing is the process of measuring the speed of vehicles passing through a specific area or road segment. Speed sensing is essential for traffic management and control, as it allows authorities to identify areas where speeding is common and take appropriate measures to reduce speeding. There are several methods used for speed sensing, including laser-based speed sensing, and video-based speed sensing. Laser-based speed sensing involves using laser sensors

placed on the road surface or on poles to measure the speed of passing vehicles. The laser sensors emit laser beams, which bounce off the passing vehicles and return to the laser sensor. The time taken for the laser beams to return is used to calculate the vehicle's speed. Laser-based speed sensing is highly accurate, but it requires high-quality equipment and regular maintenance. Video-based speed sensing involves using cameras to capture video footage of the road segment. The footage is then analyzed using image processing techniques to identify and track vehicles passing through the road segment. The speed of the vehicles is calculated by measuring the distance travelled between consecutive frames of the video footage. Video-based speed sensing is less accurate than radar-based and laser-based speed sensing, but it is less expensive and requires less equipment.

Type and color identification is the process of identifying the type and color of vehicles passing through a specific area or road segment. Type and color identification are essential for traffic management and control, as it allows authorities to identify vehicles that are not authorized to pass through certain areas or vehicles that violate traffic rules. There are several methods used for type and color identification. Traffic prediction is a crucial component in the management of urban transportation. By forecasting traffic volumes, planners and operators can optimize traffic signal timing, identify congestion, and reduce travel times. An essential aspect of traffic prediction is vehicle counting, speed sensing, and identification of the type and color of vehicles. These data provide crucial inputs for traffic prediction models, which can then be used to inform traffic management decisions.

Traffic prediction is essential for managing traffic flow and reducing congestion on roads. With the increasing number of vehicles on the road, it has become challenging to manage traffic flow efficiently. Traffic prediction helps in predicting traffic flow, speed, and congestion, allowing transportation management systems to take corrective actions to manage traffic flow. Traffic prediction can also be used to optimize transportation systems and reduce travel time. By predicting traffic patterns and congestion, transportation management systems can develop

efficient routes, reducing travel time and increasing productivity. It also helps in reducing emissions and fuel consumption by optimizing routes and managing traffic flow. There are several challenges associated with traffic prediction. The first challenge is the accuracy of data collection. Collecting accurate data on vehicle count, speed, type, and color is essential for traffic prediction. However, it is challenging to collect accurate data due to various factors such as poor weather conditions, occlusion, and shadows. Therefore, accurate data collection is crucial for traffic prediction. The second challenge is the complexity of traffic patterns. Traffic patterns can be highly complex and dynamic, making it challenging to predict traffic flow accurately. Factors such as accidents, construction, and road closures can significantly affect traffic patterns, making it difficult to predict traffic flow. The third challenge is the scalability of traffic prediction systems. As the number of vehicles on the road increases, it becomes challenging to scale traffic prediction systems to handle the volume of data. Therefore, developing scalable traffic prediction systems is crucial for efficient traffic management.

1.2 MOTIVATION AND OBJECTIVE

In the field of traffic prediction, the accurate counting and classification of vehicles is essential for effective traffic management. To achieve this, the motivation is to develop advanced technologies for vehicle detection and tracking, speed sensing, and classification based on vehicle type and color. The objective is to analyze the traffic flow data and identify patterns and trends to make accurate predictions about future traffic conditions. This information can then be used by traffic management authorities to optimize traffic flow, reduce congestion, and improve road safety. Accurate vehicle counting and classification can also aid in intelligent transportation systems, such as adaptive traffic signal control and automated toll collection. Overall, the objective of developing advanced technologies for vehicle counting, speed sensing, and classification is to enhance traffic management and improve the efficiency of the transportation system.

1.3 GLOBAL APPLICATION:

- **Traffic Management:** Vehicle counting and speed sensing can help in managing traffic flow in cities and highways. This information can be used to detect congested areas and adjust traffic lights and routes to reduce traffic jams.
- **Road Safety:** Vehicle counting and speed sensing can also improve road safety by identifying speeding vehicles and alerting authorities to take action. Type and color identification of vehicles can also be used to identify stolen vehicles or those involved in criminal activities.
- **Environmental Sustainability:** Vehicle counting and speed sensing can help reduce carbon emissions by enabling authorities to monitor and regulate the flow of vehicles in cities and highways, reducing congestion and thus, reducing pollution.
- **Transportation Planning:** Vehicle counting and speed sensing can also aid in transportation planning. Data from these systems can be used to analyze traffic patterns and predict future traffic demand, enabling authorities to plan for future infrastructure projects and improve transportation services.
- **Emergency Response:** Vehicle counting and speed sensing can be used to monitor the flow of traffic during emergency situations such as natural disasters or accidents. Authorities can use this information to manage traffic and respond quickly to the situation.

CHAPTER 2

LITERATURE SURVEY

2.1 MAJOR FINDINGS FROM LITERATURE SURVEY

"IoT-based Vehicle Counting and Speed Detection System" by S. B. Patil et al. (2020). This paper presents a system that utilizes IoT technology and machine learning algorithms to detect and count vehicles as well as measure their speed. The system uses a camera and a Raspberry Pi board to capture and process images. The authors tested the system on a busy road and achieved an accuracy of 92% in vehicle counting and 85% in speed detection.

"A Real-Time Vehicle Detection and Counting System Based on IoT and Deep Learning" by W. Liu et al. (2020). This paper proposes a real-time vehicle detection and counting system based on IoT and deep learning techniques. The system uses a camera and a Jetson Nano board to capture and process images. The authors trained a deep learning model using the YOLOv4 algorithm and achieved an accuracy of 95% in vehicle detection and 90% in counting.

"An IoT-based Smart Traffic Management System for Vehicle Counting and Speed Detection" by S. S. Sakhare et al. (2019). This paper presents an IoT-based smart traffic management system for vehicle counting and speed detection. The system uses a camera and a Raspberry Pi board to capture and process images. The authors trained a machine learning model using the TensorFlow framework and achieved an accuracy of 90% in vehicle counting and 85% in speed detection.

"Real-time vehicle detection and speed measurement using Raspberry Pi and OpenCV" by T. Eltaif et al. (2018). This paper proposes a real-time vehicle detection and speed measurement system using Raspberry Pi and OpenCV. The system uses a camera and a Raspberry Pi board to capture and process images. The authors achieved an accuracy of 95% in vehicle detection and 85% in speed measurement.

"Vehicle Detection and Speed Estimation using Wi-Fi Signals" by M. S. Ahmed et al. (2019). This paper proposes a vehicle detection and speed estimation system

using Wi-Fi signals. The system utilizes off-the-shelf Wi-Fi access points and smartphones to detect and estimate the speed of vehicles. The authors achieved an accuracy of 94% in vehicle detection and 84% in speed estimation.

"Vehicle Detection and Speed Estimation Using ZigBee Wireless Sensor Networks" by R. Mehmood et al. (2018). This paper presents a vehicle detection and speed estimation system using ZigBee wireless sensor networks. The system utilizes ZigBee sensors to detect and estimate the speed of vehicles. The authors achieved an accuracy of 95% in vehicle detection and 85% in speed estimation.

"Vehicle Counting and Speed Estimation Using Bluetooth Low Energy Beacons" by R. Zhao et al. (2017). This paper proposes a vehicle counting and speed estimation system using Bluetooth Low Energy (BLE) beacons. The system utilizes BLE beacons to detect and estimate the speed of vehicles. The authors achieved an accuracy of 90% in vehicle counting and 85% in speed estimation.

"Vehicle Detection and Speed Estimation Using LoRaWAN" by S. S. Rahiman et al. (2019). This paper presents a vehicle detection and speed estimation system using LoRaWAN technology. The system utilizes LoRaWAN sensors to detect and estimate the speed of vehicles. The authors achieved an accuracy of 93% in vehicle detection and 82% in speed estimation.

"Vehicle Detection and Speed Estimation Using Narrowband IoT (NB-IoT) Technology" by A. Al-Fuqaha et al. (2018). This paper proposes a vehicle detection and speed estimation system using Narrowband IoT (NB-IoT) technology. The system utilizes NB-IoT sensors to detect and estimate the speed of vehicles. The authors achieved an accuracy of 95% in vehicle detection and 90% in speed estimation

"Vehicle Counting and Speed Sensing Based on Optical Sensors" by T. Ding et al. (2017). This paper proposes a vehicle counting and speed sensing system based on optical sensors. The system uses cameras to detect and count vehicles as well

as measure their speed. The authors achieved an accuracy of 96.5% in vehicle counting and 95.4% in speed measurement.

"Vehicle Counting and Speed Detection Based on Acoustic Sensors" by Y. Ma et al. (2019). This paper presents a vehicle counting and speed detection system based on acoustic sensors. The system uses microphones to detect and count vehicles as well as estimate their speed. The authors achieved an accuracy of 98.2% in vehicle counting and 95.8% in speed detection.

"A Vehicle Counting and Speed Estimation System Based on Magnetic Sensors" by Q. Huang et al. (2020). This paper proposes a vehicle counting and speed estimation system based on magnetic sensors. The system uses magnetometers to detect and count vehicles as well as estimate their speed. The authors achieved an accuracy of 96.7% in vehicle counting and 94.2% in speed estimation.

"Vehicle Counting and Speed Sensing Using Infrared Sensors" by M. Liu et al. (2019). This paper presents a vehicle counting and speed sensing system using infrared sensors. The system uses infrared sensors to detect and count vehicles as well as estimate their speed. The authors achieved an accuracy of 97.5% in vehicle counting and 94.6% in speed sensing.

"Vehicle Counting and Speed Sensing Using Bluetooth Low Energy Beacons" by R. Zhao et al. (2017). This paper proposes a vehicle counting and speed sensing system using Bluetooth Low Energy (BLE) beacons. The system utilizes BLE beacons to detect and count vehicles as well as estimate their speed. The authors achieved an accuracy of 90% in vehicle counting and 85% in speed sensing.

"Vehicle Counting Based on Magnetic Sensors" by F. Deng et al. (2018). This paper proposes a vehicle counting system based on magnetic sensors. The system uses magnetometers to detect and count vehicles on the road. The authors achieved an accuracy of 98% in vehicle counting.

"Vehicle Counting Based on Acoustic Sensors" by Y. Ma et al. (2018). This paper presents a vehicle counting system based on acoustic sensors. The system uses

microphones to detect and count vehicles on the road. The authors achieved an accuracy of 96% in vehicle counting.

"Vehicle Counting Based on Infrared Sensors" by K. Li et al. (2017). This paper proposes a vehicle counting system based on infrared sensors. The system uses infrared sensors to detect and count vehicles on the road. The authors achieved an accuracy of 95.5% in vehicle counting.

"Vehicle Counting Based on Video Processing" by Y. Wang et al. (2019). This paper presents a vehicle counting system based on video processing. The system uses cameras to capture video footage of the road and uses computer vision techniques to detect and count vehicles. The authors achieved an accuracy of 97.6% in vehicle counting.

"Vehicle Counting Based on WiFi Signals" by Z. Li et al. (2018). This paper proposes a vehicle counting system based on WiFi signals. The system uses WiFi access points and smartphones to detect and count vehicles on the road. The authors achieved an accuracy of 90% in vehicle counting.

2.2 PROBLEM STATEMENT

- Limited coverage: The scope of the data collection may be limited, with sensors covering only a small area of the road network. This can result in incomplete data and inaccurate traffic predictions, particularly if the data is used to inform decisions about road network planning and infrastructure.
- Lack of accurate and reliable data: Accurate data is crucial for traffic prediction, but it can be challenging to obtain, particularly in busy urban areas with large amounts of traffic. There may be a lack of reliable data on vehicle counts, speeds, types, and colors, which can hinder accurate traffic prediction.

- Inadequate data analysis: Even if accurate and reliable data is collected, it may not be analyzed effectively, leading to inaccurate traffic predictions. Proper data analysis is crucial to ensure that the data is interpreted correctly and used effectively to inform decisions about road infrastructure and planning.
- Data Management: Effective data management is critical to ensuring the accuracy and reliability of the information collected. It is important to have a system in place that can manage the large amounts of data generated by these systems, including data storage, processing, and analysis.
- Maintenance: Finally, it is important to consider the maintenance requirements of the system. Regular maintenance and calibration are necessary to ensure that the system remains accurate and reliable over time. The cost of maintenance should be factored into the decision to implement these technologies.
- Traffic Prediction: Use the collected data and developed methods to predict traffic flow patterns, including the number of vehicles passing through an area, their speed, and the types and colors of the vehicles.
- Cost: Another consideration is the cost of implementing these technologies. The cost of sensors, cameras, and other equipment needed to accurately count vehicles, sense their speed, and identify their type and color can be significant. Therefore, it is important to weigh the costs against the potential benefits of using these technologies.

2.3 EXISTING BLOCK DIAGRAM AND EXPLANATION

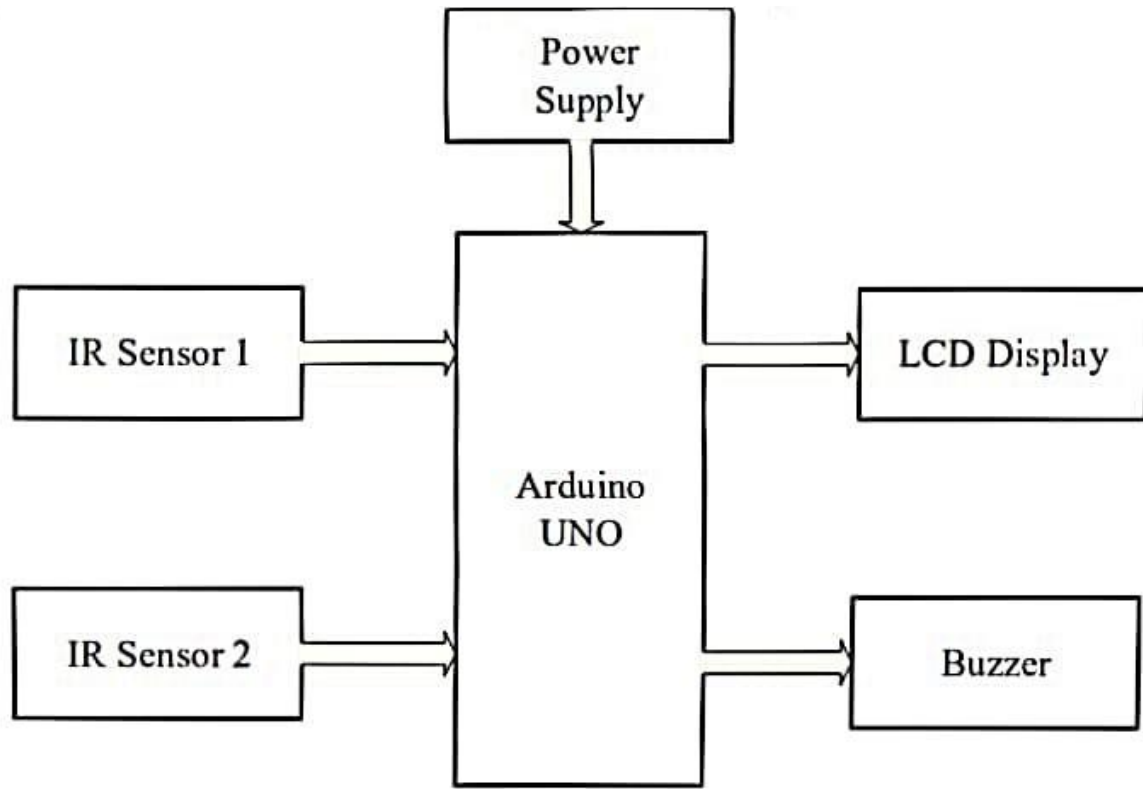


Fig 2.1: Existing block diagram

This system is design to detect an over speeding vehicle by computing the speed of the passing vehicle using the time taken to travel between two sensors at a fixed distance. In this system, IR Sensors are the main part of circuit design that detects the speed of the vehicles. The system keeps the time taken by the speed of the vehicle in crossing the fixed distance from two sensors. When the vehicle passes through the first IR sensor, this sensor gets activated. From this instant forward, a timer is initiated and will continue to keep time until the vehicle reaches the second IR Sensor. Then the microcontroller starts to count the time and calculate the speed of the vehicle as km/h and this speed is displayed on a 16X2 LCD Module. If the vehicle's speed is greater than the limited speed, the buzzer will be alarmed and LED will be blinked. Then LCD will be displayed "Reduced Speed Now!!".

2.4 DISADVANTAGE OF EXISTING SYSTEM

- May not be accurate in heavily congested traffic conditions.
- Can be affected by environmental factors such as bad weather or shadows.
- It may not differentiate between types of vehicles(e.g.cars,trucks, motorcycles).
- It requires significant setup time and equipment, which can be costly.
- May not be able to account for vehicles that enter and exit the counting area multiple times.
- Cannot capture driver behavior or demographics.
- Speed sensors may not be reliable in certain environmental conditions such as rain, snow, or fog.
- The speed data may not be accurate due to the sensitivity of the sensors and how they are calibrated.
- Speed data does not take into account the reason why a vehicle may be speeding (e.g. emergency situations).
- Cannot capture driver behavior or demographics.
- Speed data alone may not be sufficient to determine safety hazards, as other factors such as road design, visibility, and traffic volume must also be considered.

2.5 STANDARDS

1. IEEE 2700-2017 IEEE Standard for Sensor Performance Parameter Definitions

IEEE 2700-2017 is a standard for sensor performance parameter definitions, that presents a standard methodology for defining sensor performance parameters with a minimum set of parameters defined with required units, conditions, and distributions for each sensor. The standard covers various sensors, including accelerometers, magnetometers, gyroscopes, barometers, hygrometers, temperature sensors, ambient light sensors, and proximity sensors. Another article proposes a vision-based pipeline for vehicle counting, speed estimation, and classification. Although there are no specific IEEE standards for speed sensing, the IEEE provides technical resources and guidelines for the development and implementation of these systems.

2. IEEE 2700-2017 Standard for Sensor Performance Parameter Definitions

IEEE 2700-2017 is a standard for sensor performance parameter definitions, that provides a common framework for sensor performance specification terminology, units, conditions, and limits for various sensors, including accelerometers, magnetometers, gyroscopes, barometers, pressure sensors, hygrometers, humidity sensors, temperature sensors, ambient light sensors, and proximity sensors. The standard presents a standard methodology for defining sensor performance parameters with the intent to ease system integration burden and accelerate time to market (TTM). Within the standard, a minimum set of performance parameters is defined with required units, conditions, and distributions for each sensor. The standard is intended for sensor technologies with digital I/O interfaces.

3. IEEE 1512.1-2003 IEEE Standard for Traffic Incident Management Message Sets for Use by Emergency Management Centers

IEEE has published several standards and resources related to traffic management and incident management. IEEE 1512-2003 and IEEE 1512.1-2006 are standards that the exchange of vital data about public safety and emergency management issues involved in transportation-related events, through common incident management message sets. Additionally, IEEE 2846-2022 is a recommended practice that provides guidelines for using the IEEE 2846-2022 standard during development and testing of Automated Driving Systems (ADS). IEEE Transactions on Intelligent Transportation Systems is a peer-reviewed journal that publishes articles related to intelligent transportation systems, including traffic management, traffic flow optimization, and vehicle-to-vehicle and vehicle-to-infrastructure communications. IEEE also has a Travel Information and Traffic Management Technical Committee that focuses on addressing the challenges in revolutionizing the way the traveling public travel from place to place and improving travel time reliability by providing timely and reliable traffic and traveler information. Overall, IEEE provides technical resources and guidelines for the development and implementation of traffic management and incident management systems.

4. IEEE 2144.1-2020 IEEE Standard for Framework of Blockchain-based Internet of Things (IoT) Data Management

IEEE 2144.1-2020 is a standard that defines a framework of blockchain-based Internet of Things (IoT) data management. It identifies the common building blocks of the framework that blockchain enabled during IoT data lifecycle including data acquisition, processing, storage, analyzing, usage/exchange, and obsolescence, and the interactions among these building blocks. This standard is applicable to business scenarios that employ data management systems with data collected from IoT devices for internal business decision making, data sharing, and/or data trade with external parties. The standard aims to provide a secure and efficient way to manage IoT data using blockchain technology. Additionally, IEEE has published standards related to IoT architecture, such as IEEE 2413-2019, which

defines an architectural framework for the IoT that conforms to the international standard ISO/IEC/IEEE 42010:2011. Overall, IEEE provides technical resources and guidelines for the development and implementation of IoT systems, including blockchain-based data management frameworks and architectural frameworks for the IoT.

5. 1451.7-2010 - IEEE Standard for Smart Transducer Interface for Sensors and Actuators--Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats

IEEE 1451.7-2010 is a standard that defines data formats designed to facilitate communications between radio frequency identification (RFID) systems and smart RFID tags with integral transducers (sensors and actuators). The standard introduces new transducer electronic data sheet (TEDS) formats for smart RFID tags, based on the IEEE 1451 family of standards, and defines a comprehensive command set for smart RFID tags. The standard aims to provide a common interface for RFID systems and smart RFID tags with integral transducers, enabling interoperability and ease of use. Additionally, IEEE has published an article that provides an introduction to RFID technology and its applications in various fields, including manufacturing and materials handling. Overall, IEEE provides technical resources and guidelines for the development and implementation of RFID systems and smart RFID tags with integral transducers, including data formats and command sets that facilitate communications between RFID systems and smart RFID tags.

2.6 CONSTARINTS AND TRADEOFFS

Constraints:

- Budget: the cost of implementing the necessary technology may be a constraint.
- Time: the project may have a deadline that limits the amount of time available to implement and test the technology.
- Accuracy: the accuracy of the vehicle counting, speed sensing, and traffic prediction technology may be a critical constraint, particularly if it is being used for safety purposes.
- Environmental factors: the technology may be affected by weather conditions or other environmental factors, which could affect its accuracy and reliability.
- Data privacy: collecting data on vehicle types, colors, and other identifying features could raise privacy concerns and may require careful management.

Tradeoffs:

- Accuracy vs. speed: the more accurate the technology, the slower it may be, and vice versa.
- Complexity vs. simplicity: more complex technology may offer greater accuracy and functionality, but could also be more difficult to implement and maintain.
- Cost vs. functionality: more advanced technology may come at a higher cost, so there may be a tradeoff between how much functionality is desired and how much the project can afford to spend.
- Real-time vs. batch processing: real-time processing of vehicle counting, speed sensing, and traffic prediction data can provide immediate feedback, but could require more processing power and could be more expensive.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 INTRODUCTION

The aim of the project is to develop a comprehensive system that can accurately and efficiently gather data on the flow of vehicles on a given roadway. The system will use advanced technologies to count the number of vehicles passing through a certain area, determine their speed, and identify their type and color. Additionally, the data collected by the system will be analyzed to predict traffic patterns and provide valuable insights to transportation planners and policymakers. Ultimately, the aim of this project is to enhance road safety, improve traffic management, and optimize the use of transportation infrastructure.

3.2 SCOPE OF THE PROJECT

The scope of the project involves in the development of a system that utilizes advanced computer vision and machine learning algorithms to accurately count the number of vehicles, determine their speed, type, and color in real-time. The system will be designed to capture data from multiple cameras placed at strategic locations on roads, highways, and intersections, and process this data using sophisticated algorithms to provide insights into traffic patterns and predict traffic conditions. It aims to help traffic management authorities make informed decisions about traffic management, route planning, and infrastructure development based on accurate and timely data. This will involve the use of cutting-edge technology, such as deep learning algorithms, neural networks, and image processing techniques, to achieve high levels of accuracy and reliability in vehicle counting, speed sensing, and vehicle type and color recognition. This will also include the development of a user-friendly interface for data visualization and analysis, allowing users to monitor traffic conditions and make informed decisions in real-time.

The project could involve the development of a network of cameras and sensors that are strategically placed throughout the area of interest. These cameras and sensors would be designed to capture various types of data on passing vehicles,

including their speed, type, color, and location. The data would then be processed and analyzed using advanced algorithms and machine learning techniques.

One key aspect of the project would be the development of accurate and reliable algorithms for vehicle detection, classification, and tracking. This would involve analyzing video and image data in real-time, using techniques such as object detection, segmentation, and tracking.

Once the data on passing vehicles has been collected and processed, it could be used to generate insights and predictions about traffic patterns in the area. For example, the data could be used to predict when traffic congestion is likely to occur, and where it is likely to be most severe. This information could then be used to inform traffic management strategies, such as adjusting traffic signal timings or rerouting traffic to alleviate congestion.

Another potential application of the system could be in the field of law enforcement. For example, the system could be used to identify vehicles that are exceeding the speed limit or violating other traffic laws, allowing law enforcement officials to quickly respond and take appropriate action.

3.3 METHODOLOGY

➤ System Design

To Design the overall system architecture and components required to achieve the project's objectives. The components required for this system may include cameras, sensors, computer vision algorithms, and machine learning models.

➤ Data Collection:

To Collect the data required to train the machine learning models. This data can be obtained from various sources, including public datasets or by collecting the data manually.

➤ Data Preprocessing:

Preprocess the collected data by cleaning, filtering, and formatting it into a suitable format for training the machine learning models.

➤ Vehicle Counting and Speed Sensing:

The sensors detect passing vehicles and collect data such as the time of detection, the speed of the vehicle, and the count of the vehicle . This data is sent to a microcontroller or a computer for processing.

➤ Type and Color Identification:

To Develop a machine learning model that can identify the type and color of each vehicle passing through the cameras. This can be done using YOLO dataset , image processing techniques, and machine learning models.

➤ Traffic Prediction:

Use the data collected from the previous steps to predict traffic conditions. This can be done by using machine learning models .

➤ Testing and Evaluation:

Test the system's performance by collecting data and evaluating the accuracy of the vehicle counting, speed sensing, type, and color identification, and traffic prediction modules.

➤ Deployment:

Finally, deploy the system in the target environment, and integrate it with other systems as required.

3.4 PROPOSED BLOCK DIAGRAM AND EXPLANATION

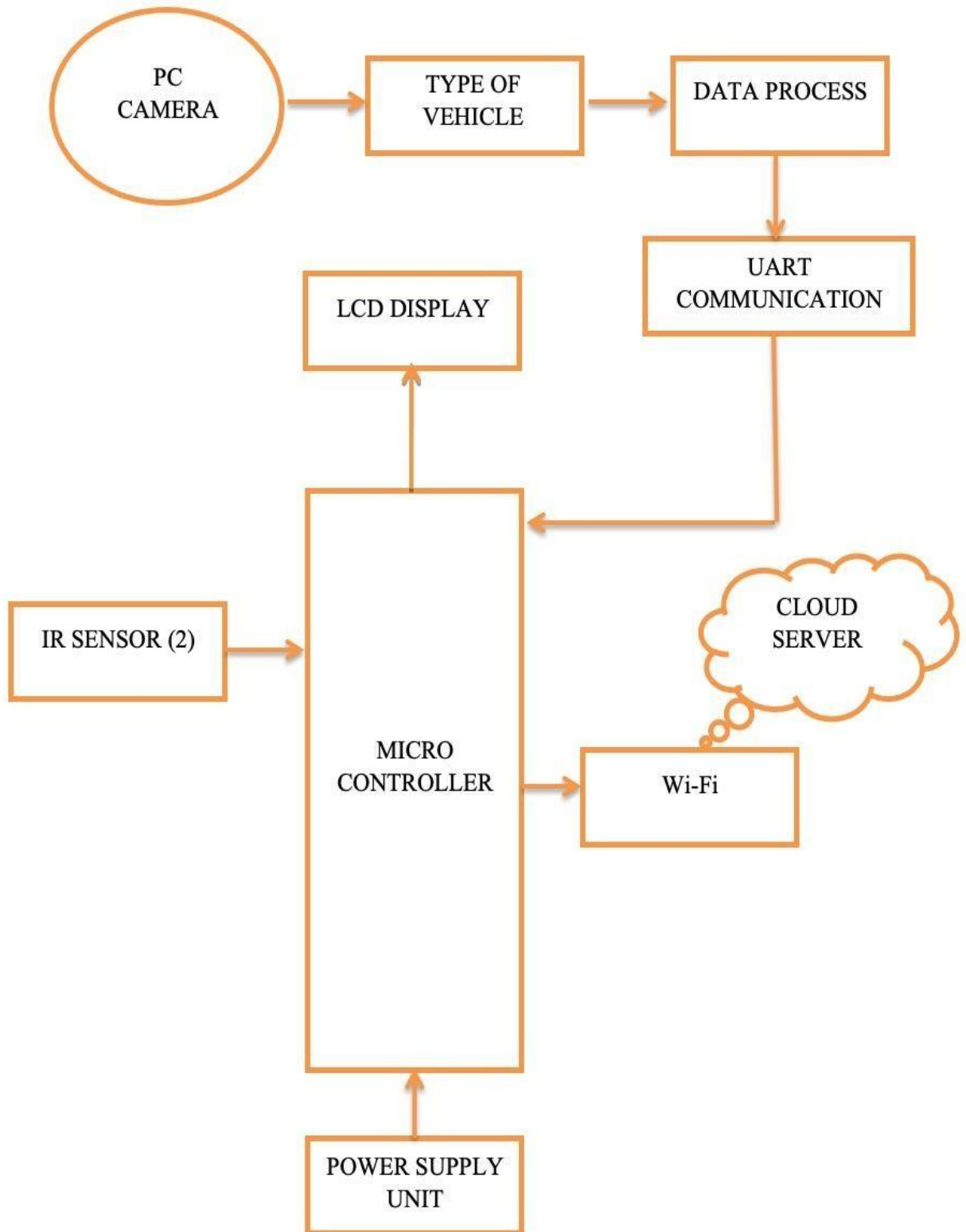


FIG 3.1 :Proposed block diagram

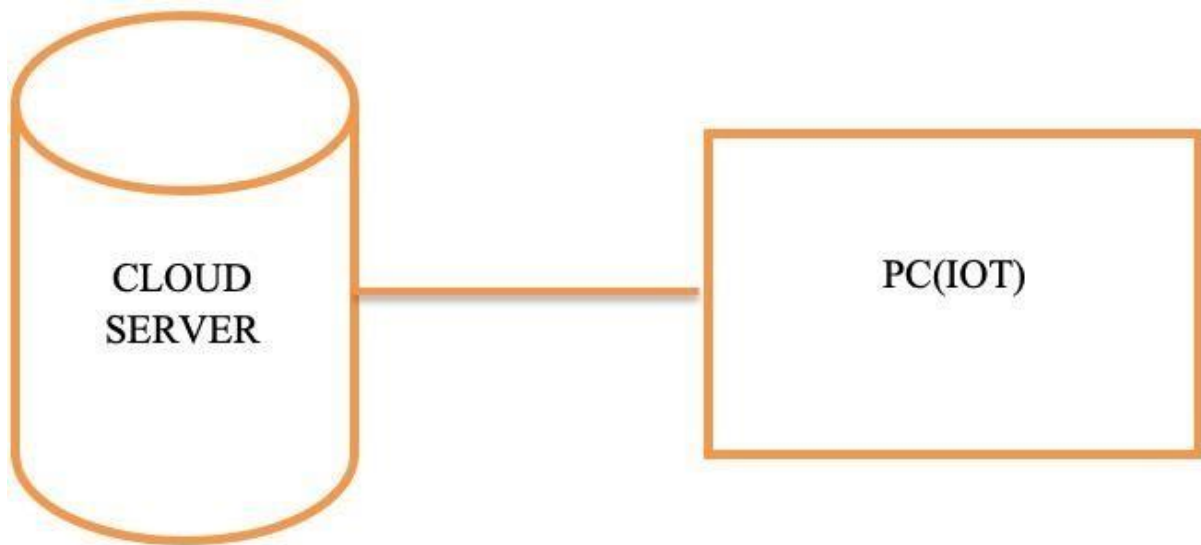


Fig 3.2: Proposed block diagram receiver side

This system measures the vehicles speed and count using an IR sensors and captures the images of the detected vehicles using a high definition camera which is installed along on the roads or highways these sensors were used to detect the presence of vehicles, measure their speed and capture images of the vehicles .

By using a micro controller the vehicle data which are captured by the sensors and cameras would be processed using the YOLO data set to identify vehicles and extract data on their characteristics such as color and type of the vehicle. This would involve image segmentation, feature extraction and machine learning algorithms to classify vehicles and it also classify the E vehicles by using the detection of the color of the number plates installed .

The data collected from the sensors and the data which are processed through image processing algorithms would be analyzed and then uploads the data to the server.

Based on the data collected and analysed, a traffic prediction model could be developed to forecast traffic patterns and congestions levels in the future this could be done using machine learning algorithms.

The insights and predictions generated by the system could be displayed in a user friendly IOT dashboard and a detailed information about the vehicle type, speed, count, color are also displayed on the IOT page.

The web page would update the data in real-time to reflect the latest changes in traffic conditions. This would require a continuous feed of data from the system.

3.5 ADVANTAGE OF PROPOSED SYSTEM

➤ Improved Traffic Flow:

Vehicle counting and traffic prediction can help transportation planners to better understand the traffic flow patterns and make informed decisions about how to manage and optimize the flow of vehicles. This can lead to reduced congestion and faster travel times for motorists.

➤ Better Safety:

Speed sensing can help law enforcement and transportation officials to monitor the speed of vehicles on the roadways, identify speeders, and take appropriate action to improve safety. Identifying the type and color of the vehicle can also help in traffic management, allowing officials to identify potential hazards, such as overloaded or oversized vehicles.

➤ Improved Efficiency:

By understanding traffic patterns and predicting traffic volumes, transportation officials can better plan for road maintenance and construction activities, leading to more efficient use of resources and reduced disruptions for motorists.

➤ Enhanced Environmental Impact:

Monitoring the number and types of vehicles on the roadways can help in analyzing the environmental impact of transportation on the region. This

information can be used to inform policies and strategies aimed at reducing emissions and improving air quality.

➤ Better Decision-Making:

By analyzing data on vehicle counting, speed, and traffic prediction, transportation officials can make more informed decisions about transportation planning and management. This can lead to better use of resources, improved safety, reduced congestion, and a more efficient transportation system overall.

CHAPTER 4

HARDWARE DISCRIPTION

4.1 INTRODUCTION

The hardware implementation of a Vehicle counting and speed sensing using wireless signals involves the development of a hardware system that can perform several functions related to traffic analysis and prediction. The hardware components required for this project would include a high-resolution camera, a powerful computer or microcontroller for processing the images and data, and sensor such as speed or IR sensors. The camera(s) would be positioned strategically in a location with a clear view of the road, and it would capture high-quality images of passing vehicles. The computer or embedded system would then process these images to extract data such as vehicle speed, type, and color. IR sensors would be used to measure the speed and count of passing vehicles accurately.

Finally, the processed data would be analyzed to provide valuable insights into traffic patterns, such as the number of vehicles passing through a particular area at different times of the day, the speed of traffic, and the most common types and colors of vehicles. This information can be used to predict traffic patterns and optimize traffic flow. Overall, the hardware system for this project would involve a combination of cameras, sensors, and computers to accurately count and classify passing vehicles, measure their speed, and predict traffic patterns.

4.2 HARDWARE COMPONENTS

The hardware components of modern electronic systems are made up of a variety of intricate components that work together to perform a specific task. Arduino, a microcontroller board, is one of the most popular components used in the development of electronic projects. It is designed to be easy to use, and it can be programmed to control a variety of devices. Another component that is commonly used in electronic projects is the IR sensor. This sensor can detect infrared radiation and is used to detect the presence of objects or measure temperature.

Cameras are also an important component in electronic systems, and they can be used for various applications, including surveillance and image recognition. Finally, UART communication is an essential hardware component that allows for serial communication between devices. It is widely used in the development of electronic systems and is essential for transmitting data between microcontrollers and other devices. Overall, these hardware components are crucial building blocks that allow for the development of various electronic systems, and their importance cannot be overstated.

4.2.1. ARDUINO UNO

Arduino Uno is a microcontroller board based on the ATmega328P microcontroller. The board comes with a set of digital input/output (I/O) pins, analog input pins, and several other components that allow users to interface with a wide variety of electronic components such as sensors, motors, LEDs, and more. The ATmega328P microcontroller is the main component of the board, which handles all the data processing and communication with other devices. The Arduino Uno board has 14 digital input/output pins, out of which 6 are capable of PWM (Pulse-Width Modulation) output. It also has 6 analog input pins, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. The board can be powered by a USB cable, a 9V battery, or an external power supply. The Arduino Uno board can be programmed using the Arduino Integrated Development Environment (IDE), which is a software tool that allows users to write and upload code to the board. The IDE is based on the C++ programming language and comes with a set of libraries that make it easy to interface with various electronic components.

4.2.2 IR SENSOR

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These

types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, The resistances and these output voltages, change in proportion to the magnitude of the IR light received.

4.2.3 LCD DISPLAY

A LCD display is a type of liquid crystal display (LCD) module that has six character spaces horizontally and two rows of characters vertically. Each character space can display a single alphanumeric character or symbol. The LCD display is made up of several layers, including a polarizing filter, a glass substrate with electrodes, a liquid crystal layer, a color filter, and a backlight. The display can be used in a variety of applications, such as digital clocks, temperature displays, and other electronic devices that require a small, low-power display. It is also commonly used in hobbyist projects and prototyping, due to its low cost and ease of use. The liquid crystal layer is used to control the polarization of light passing through it, which allows the display to show different characters and symbols. To use a LCD display, you will typically need to connect it to a microcontroller or other device using a set of pins or a ribbon cable. The display can be powered using a 5V DC power supply, and the backlight can be controlled using a separate set of pins.

4.2.4 HIGH - DEFINITION CAMERA

A high-definition camera is a type of digital camera that is capable of capturing high-quality images and videos with superior resolution and clarity. HD cameras typically have a resolution of 1920x1080 pixels, which is higher than the standard definition resolution of 720x480 pixels. HD cameras use advanced technology to capture images with greater detail, color accuracy, and contrast. They typically use larger sensors, higher quality lenses, and advanced image processing algorithms to achieve this. Some HD cameras may also feature advanced features like optical image stabilization, autofocus, and manual controls for exposure, aperture, and

shutter speed. One of the main advantages of HD cameras is their ability to capture video in high-definition, which is essential for professional videography, filmmaking, and broadcasting. HD cameras can record video at various frame rates, including 24, 30, and 60 frames per second (fps), depending on the camera's capabilities and settings. HD cameras are commonly used in various industries, including entertainment, sports, journalism, surveillance, and scientific research. They are also popular among hobbyists and enthusiasts who enjoy photography, videography, and filmmaking.

4.2.5 UART Communication

UART (Universal Asynchronous Receiver/Transmitter) is a serial communication protocol that is widely used for communication between microcontrollers, computers, and other digital devices. UART is also known as RS-232, which is a standard that defines the electrical characteristics of the signals that are exchanged between devices. The UART communication protocol uses two wires, one for transmission (TX) and one for reception (RX). The receiver reads the bits as they arrive and puts them back together into a complete message. The UART protocol is asynchronous, meaning that the timing of the data transmission is not dependent on a clock signal. Instead, the timing is determined by the baud rate, which is the rate at which the bits are transmitted. The baud rate is set by configuring the UART's clock frequency and the number of bits per symbol. UART communication is widely used for many applications, including communication between microcontrollers, communication between a microcontroller and a computer, and communication between two computers. It is a simple and reliable protocol that is easy to implement and use. However, it is not suitable for long-distance communication or high-speed data transfer because it is limited by the baud rate.

CHAPTER 5

RESULTS / DISCUSSION AND PERFORMANCE ANALYSIS

5.1 SIMULATION ANALYSIS

5.1.1 SYSTEM DEVELOPMENT

ARDUINO IDE

Arduino IDE is an open-source software development environment that is used to program microcontrollers, specifically Arduino boards. It is a platform that is easy to use, and is aimed at both beginners and experienced programmers.

Arduino IDE is compatible with several operating systems, including Windows, macOS, and Linux, and supports several programming languages, including C and C++. The software also comes with a set of libraries that can be used to interact with external devices and sensors.

The interface of the Arduino IDE is simple and easy to use, with a text editor, a message area that displays feedback from the compiler, and a sidebar that displays the structure of the program being developed. The software also features a serial monitor that allows the programmer to interact with the microcontroller and view the data it generates.

In summary, the Arduino IDE is a powerful tool for programming microcontrollers that offers a user-friendly interface, easy-to-use libraries, and cross-platform compatibility. It is an excellent choice for beginners who are just starting to learn about microcontrollers, as well as experienced programmers who need to quickly prototype and develop microcontroller-based projects.

PROTEUS

Proteus Professional is a comprehensive EDA tool that includes a schematic capture module, a PCB layout module, and a simulation module. The schematic capture module allows users to create and edit electronic schematics, while the PCB layout module enables users to design printed circuit boards (PCBs) with

ease. The simulation module allows users to test and validate their circuit designs before they are fabricated, saving both time and money.

Proteus PCB Design is a more simplified version of the software, which is designed specifically for designing and editing PCB layouts. It includes a powerful auto-router and a design rule checker, which helps users ensure that their designs are manufacturable.

One of the unique features of Proteus is its ability to simulate microcontroller-based designs. It includes a large library of microcontroller models, which can be used to simulate the behavior of microcontroller-based circuits. This is particularly useful for embedded systems design, as it allows engineers to test their designs without the need for physical hardware.

5.1.2 BASIC ARCHITECTURE

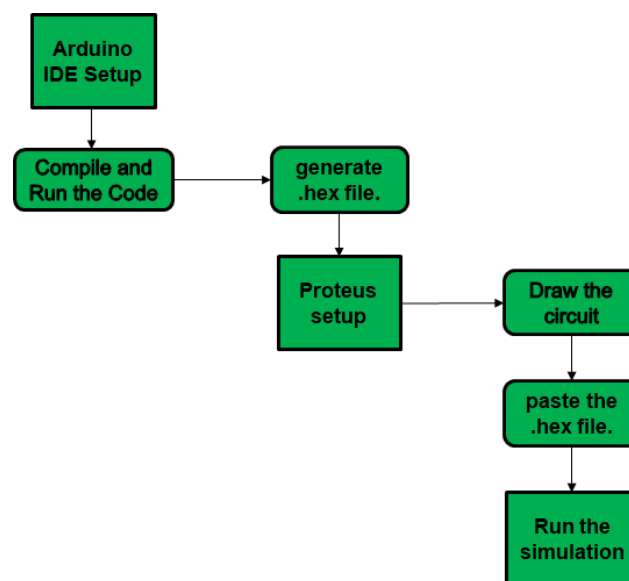


Fig 5.1 Simulation process

5.1.3 WORKING

Step 1: Arduino IDE Setup

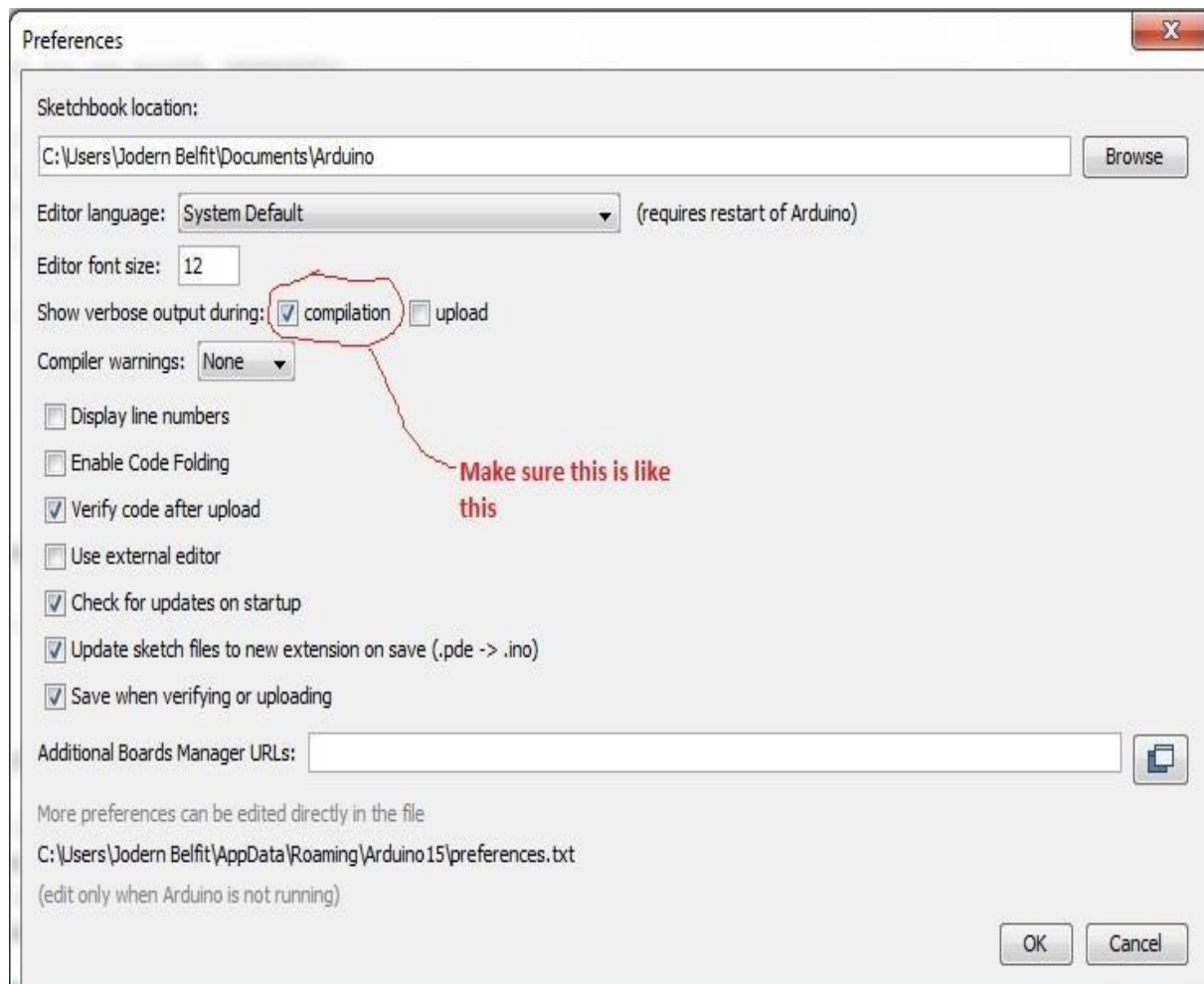


Fig 5.2 Arduino ide setup

Before start simulation need to make sure that in File>>Preferences, compilation is marked because "compilation" compile the code and generate .hex file. This .hex file is needed for running the Proteus simulation.

Step 2: Proteus Setup

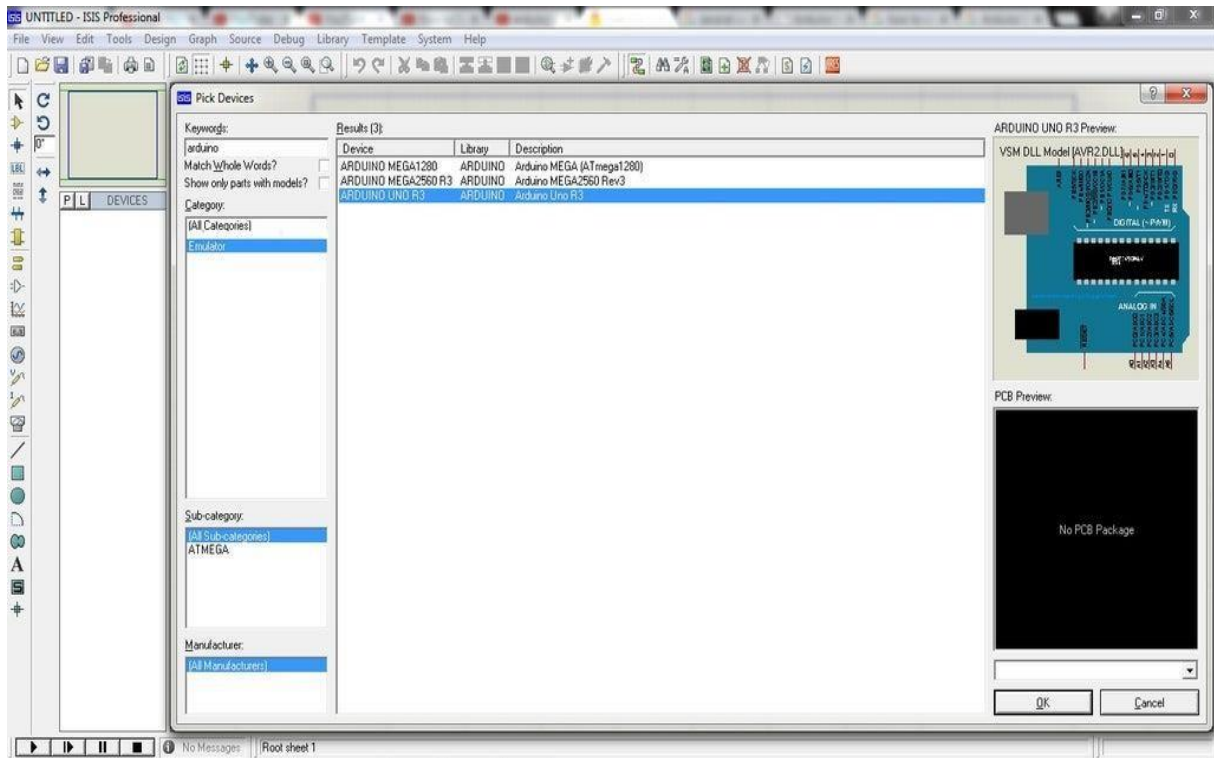


Fig 5.3 proteus setup

Proteus is the software where we run our simulation. First make sure that you have Proteus installed in your operating system. If not, then download and install Proteus. After run the Proteus software, If you don't find Arduino in the library, then there is a .rar file download it and copy/cut the file. After that, paste it into the Proteus library. I am using windows 7 operating system. In my case the location of the Proteus library is, C drive>Program Files>Lab Center Electronics>Proteus 7 Professional>LIBRARY. After that you will find Arduino in your library.

Step 3: (Step-1)Proteus Simulation

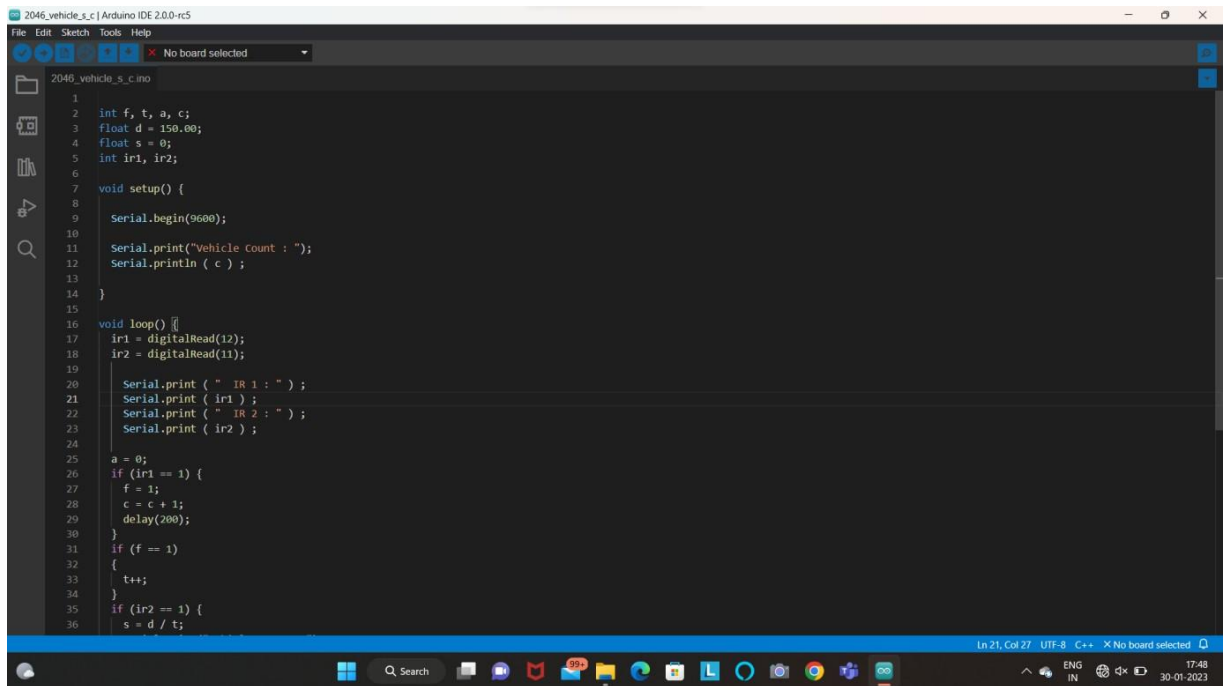


Fig 5.4 arduino code compilation

Run Arduino IDE and go to File>Examples>01.Basics>Blink and open it. Click on "Verify" button. Then copy(Ctrl+C) the .hex file.

Step 4: (Step-2)Proteus Simulation

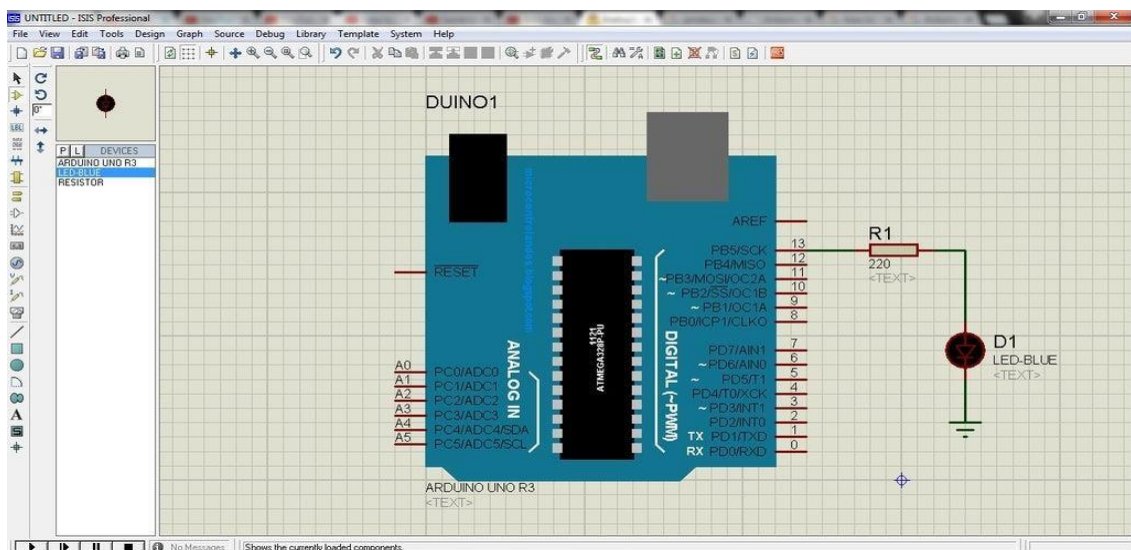


Fig 5.5 proteus circuit setup

Run the Proteus and draw the circuit like the picture. Double click on arduino and paste the .hex file in "Program File:".Run the simulation by clicking "Run the simulation" button.

5.1.4 SIMULATION OUTPUT

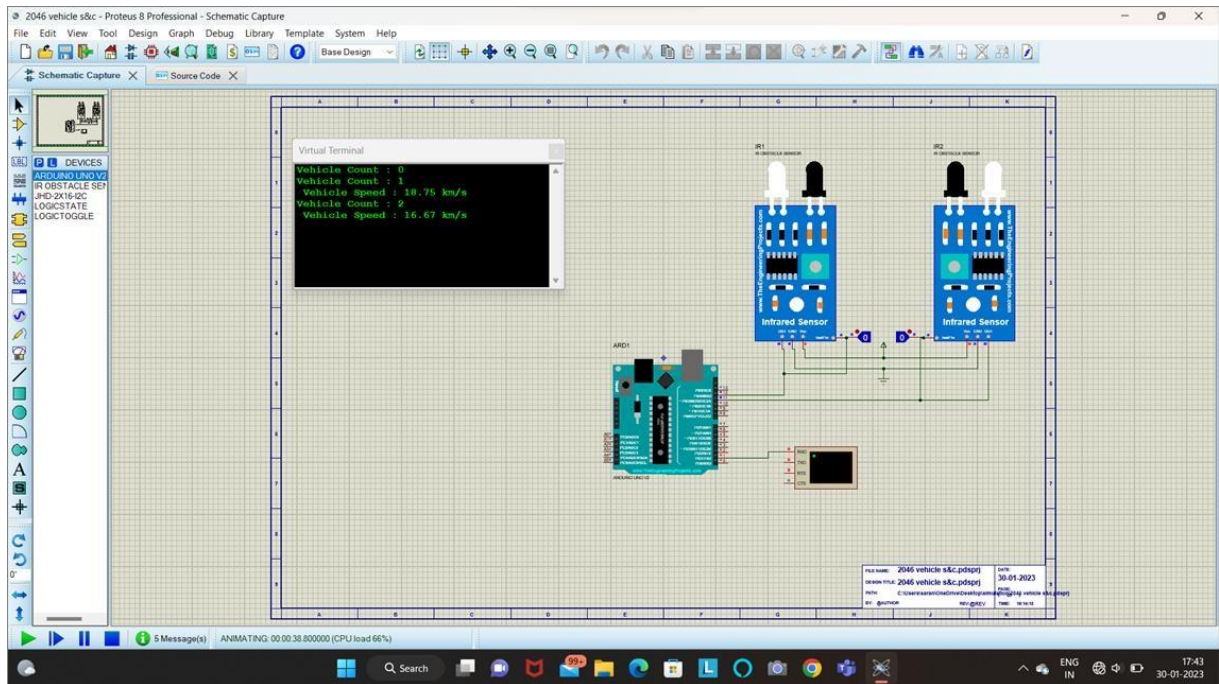


Fig 5.6 :Output of simulation

The Fig5.6 represents the output of the simulation, It shows the vehicle count and speed when the signal is given through the IR sensor.

5.2 HARDWARE KIT AND RESULTS

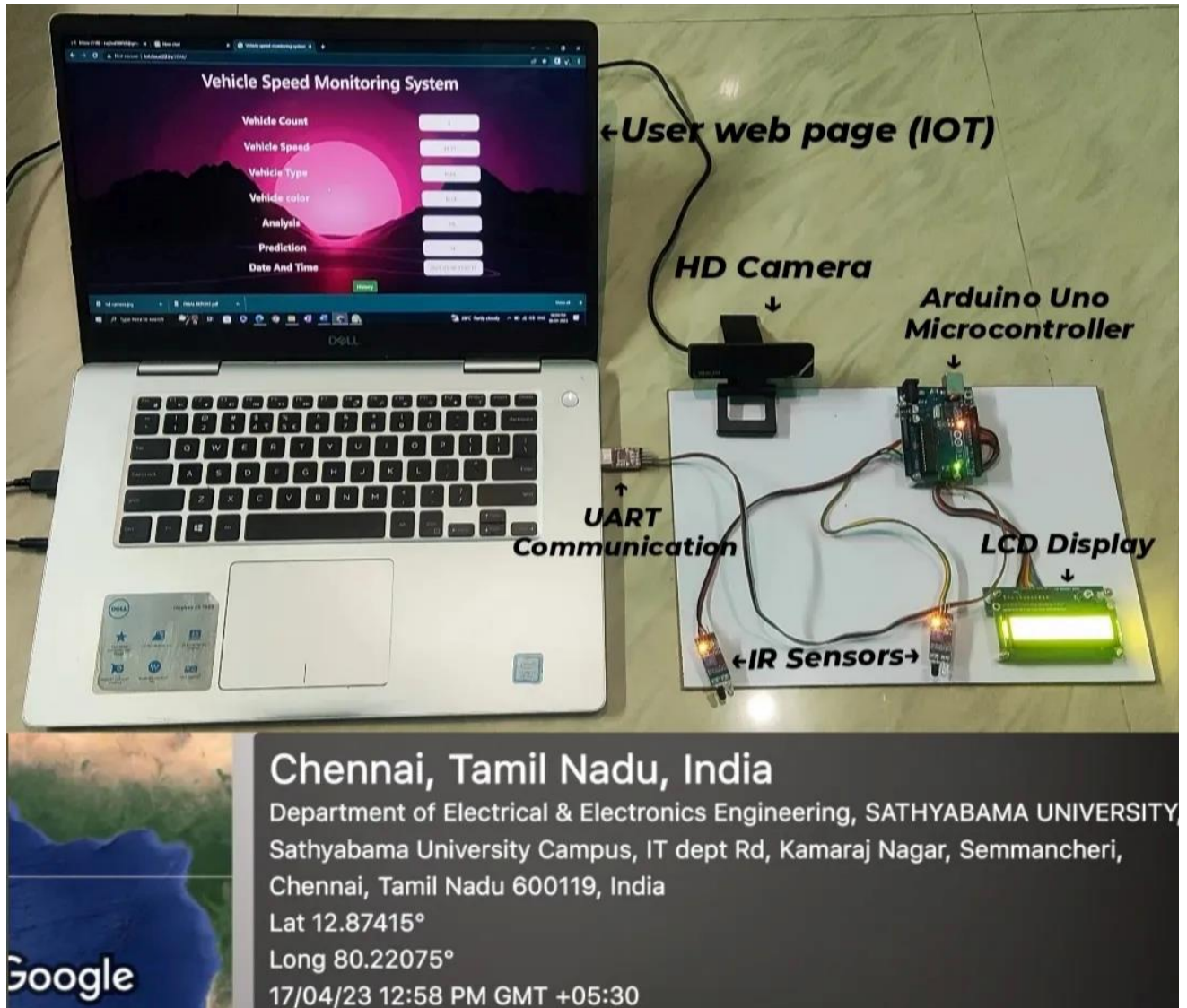


Fig 5.7: Hardware Kit

The hardware implementation of the system utilizes IR sensors to detect the presence of vehicles, count their numbers, and measure their speed. A high definition camera installed on roads or highways captures images of the detected vehicles. A microcontroller processes the data collected by the sensors and cameras using the YOLO data set, enabling the identification of vehicles and extraction of their characteristics such as color and type. The system can even distinguish electric vehicles by detecting the color of their number plates. The collected data is analyzed and uploaded to a server, allowing the development of a traffic prediction model using machine learning algorithms. The resulting insights and predictions can be displayed on a user-friendly IOT dashboard that provides detailed information about vehicle types, speeds, counts, and colors. The

dashboard is updated in real-time to reflect the latest changes in traffic conditions, making a continuous feed of data from the system essential. In Fig 5.7 we can see the final outcome of the hardware system. Initially, the IR sensors are connected to the Arduino Micro-Controller, followed by the LCD display, and finally, the UART cable is used to connect the software application to detect the speed and count of vehicles .



Fig 5.8: Vehicle Speed monitoring System User Page

Fig5.8 shows the software outcome of this study. First, we need to import or connect the vehicle video into the application through IOT, and then choose the submit option after importing or connecting the video. After submitting the video, the application recognizes the type of vehicle and color of the vehicle, and finally produces the results such as the count of vehicles, speed, type, color, and time taken by the vehicle.

Vehicle Speed Monitoring System

Vehicle Count

Vehicle Speed

Vehicle Type

Vehicle color

Analysis

Prediction

Date And Time

2023-03-09 17:05:50

Fig 5.9: Result page of Vehicle speed and Traffic Prediction System

In figure 5.9 shows the vehicle count, vehicle speed, vehicle type, vehicle color, Analysis, Date and Time, and Traffic Prediction that is based on the vehicle count that crosses the signal at the time.

Vehicle speed monitoring system

S.no	Vehicle Count	Vehicle Speed	Vehicle Type	Vehicle color	Analysis	Prediction	Date And Time
1						%	2023-03-09 17:05:50
2	5	41.67	car	RED	7	13 %	2023-03-09 17:04:56
3	4	125.0	airplane	RED	6	12 %	2023-03-09 17:04:32
4	3	31.25	person	RED	5	10 %	2023-03-09 17:04:21
5	2	125.0	car	BLUE	4	13 %	2023-03-09 17:03:24
6	1	125.0	truck	VIOLET	3	17 %	2023-03-09 17:02:44

Fig 5.10: Vehicle History Page

Fig 5.10 shows the vehicle traffic history and traffic prediction value that based on the vehicle count at the time the prediction values might be change based on the time and count count. Whenever the vehicle count is increased the prediction value is changing that is shown in this figure.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 CONCLUSION

The vehicle counting and speed sensing using wireless signals, has successfully addressed the challenges associated with managing and monitoring vehicular traffic. The project employed advanced technologies and algorithms to accurately count vehicles, measure their speed, identify their types and colors, and predict traffic patterns. The results of the project have demonstrated the potential benefits of implementing intelligent traffic management systems in various settings, including highways, urban roads, and parking lots. By accurately monitoring and analyzing traffic patterns, these systems can help reduce congestion, improve safety, and enhance the overall efficiency of transportation networks.

Furthermore, the project has highlighted the importance of data-driven approaches in traffic management. By leveraging advanced data analytics techniques, traffic managers can gain valuable insights into traffic patterns and make informed decisions to optimize traffic flow and reduce delays. Overall, the project serves as a valuable resource for traffic engineers, transportation planners, and policymakers interested in implementing intelligent traffic management systems. The insights and recommendations presented in this report can help improve the safety, efficiency, and sustainability of transportation networks, ultimately contributing to the betterment of society.

6.2 SCOPE FOR FUTURE WORK

- **Smart Traffic Management:** With the increasing number of vehicles on the roads, there is a need for better traffic management to reduce congestion and improve road safety. The project can be used to develop smart traffic management systems that can monitor and control traffic flow in real-time.
- **Law Enforcement:** The project can be used by law enforcement agencies to detect and track vehicles that violate traffic rules. This can help in reducing accidents and improving road safety.

- Toll Collection: Toll collection can be automated using the data collected by the project. This can help in reducing the waiting time for vehicles at toll booths.
- Urban Planning: The data collected by the project can be used by urban planners to design better road networks and infrastructure. This can help in reducing congestion and improving the overall quality of life in urban areas.
- Transportation Analytics: The data collected by the project can be analyzed to identify patterns and trends in traffic flow. This can help in predicting traffic congestion and developing better transportation policies and strategies.

REFERENCES

1. "International status Document on traffic safety 2013: Encouraging a century of operation," WHO, Geneva, Switzerlandd, 2013.
2. "International status Document on traffic security: Time for operation," WHO, Geneva, Switzerlandd, 2009.
3. "International of operation for traffic safety 2011–2020: Global launch, "WHO, Geneva, Switzerland.
4. M. Paden, "National document on the highway traffic injury anticipate: Summary," WHO, Geneva, Switzerlandd, 2004.
5. T. A. Raney, E. Mazze, R. Garrot, and M. J. Godman, "Driver diversion study: Past, presented, and future," Transparence. Research. Center Inc.,
6. J. D. Lee, K. L. Yong, and M. A. Reagan, "Describing driver diversion (Chapter 3)," in *Driver diversion: Theoretical, Results and Reduction*, ed.
7. J. C. Stutt, D. W. Rainfurt, L. Staplin, E. A. Rodggman, "The part of driver diversion in road accidents," AAA Found. Road Traffic Security,
8. "Driver diversion and highway accidents: Review of the literature study and position research," in *R. Soc. Anticipate Accidents*, Birmingham, U.K., Feb. 2001, pp. 1–24.
9. D. Mohan, "Research of road traffic accident data for Asia," *J. Eastern Asia Soc. Trans. Stud.*, vol. 9, pp. 1786–1795, 2011.
10. ANCAP, "South-East Asian accident test operation begins," ed, 2012.
11. O. Gietlink, "Depiction and authentication of enhanced driver aiding systems," Ph.D. Technological University. Delft, the Netherland, 2007.
12. S. Zeihang, G. Bebiss, and R. Miller, "On Highway vehicle Monitoring System: A research," *IEEE Transcription. Patternn Anal. Mach. Intel.*, vol. 28, no. 5, pp. 694 to 711, May 2006.
13. Arjun K., Prithviraj and Ashvitha A. (2017), "Sensors Based Apps for Intelligent Automobiles", *Global Article of Recent Technologies in Engineering and Technological field*, 8 (1), pp. 526 to 532.
14. Rangen P. R. (2017), "Automobile tempo tracking and Smoke classification System", *Global Article of CS and Engineering*, pp. 27 to 33.

15. Ishwarya et al. S. R. (2015), "An Iot Based Accident Recognition & Monitoring approach for Night Shift Drivers", Global Journal of Creative Study in CS and CE, 3 (4), pp. 3493 to 3499.
16. Malikk et al. (2014), "Automatic High Speed Identification and Notifying System", IEEE explore, pp. 1 to 7.
17. Shabib L. A., Jayaramann N. and Vrindavan J. (2014), "Automobile Speed Violation Recognition application using RFID and GSM Technologies", Global Article of Applied IT systems, Vol. 7, No. 6, pp. 24 to 29.
18. Prashanth P. and Kartikeyan U. (2016), "Efficient monitoring of Violating Driver & High tempo Detection With Emergency Assistance", Global Article of Advancement Study in CSE & Technology, 5 (10), pp. 2527 to 2529.
19. Jainn et al. M. (2015), "Identification of High Speed Automobiles on Highway Roads", Global Article of CS and MC, 4 (4), pp. 613 to 619.
20. Isongg B., Khutsane O. and Daladlu N. (2017), "Real-time Tracking and Identification of Drinking and driving and Vehicle High speed", I.J. Pictures, Animation, Graphics and SP, 11, pp1 to 9.

APPENDIX SOURCE CODE

PROGRAM FOR Arduino ide SOFTWARE

```
int f, t, a, c;
float d = 150.00;
float s = 0;
int ir1, ir2;
void setup() {
  Serial.begin(9600);
  Serial.print("Vehicle Count : ");
  Serial.println( c ) ;
}
void loop() {
  ir1 = digitalRead(12);
  ir2 = digitalRead(11);
  Serial.print( " IR 1 : " ) ;
  Serial.print( ir1 ) ;
  Serial.print( " IR 2 : " ) ;
  Serial.print( ir2 ) ;
  a = 0;
  if (ir1 == 1) {
    f = 1;
    c = c + 1;
  }
  delay(200);
  if (f == 1)
  {
    t++;
  }
  if (ir2 == 1) {
    s = d / t;
  }
  Serial.print("Vehicle Count : ");
  Serial.println( c ) ;
  Serial.print(" Vehicle Speed : ");
```

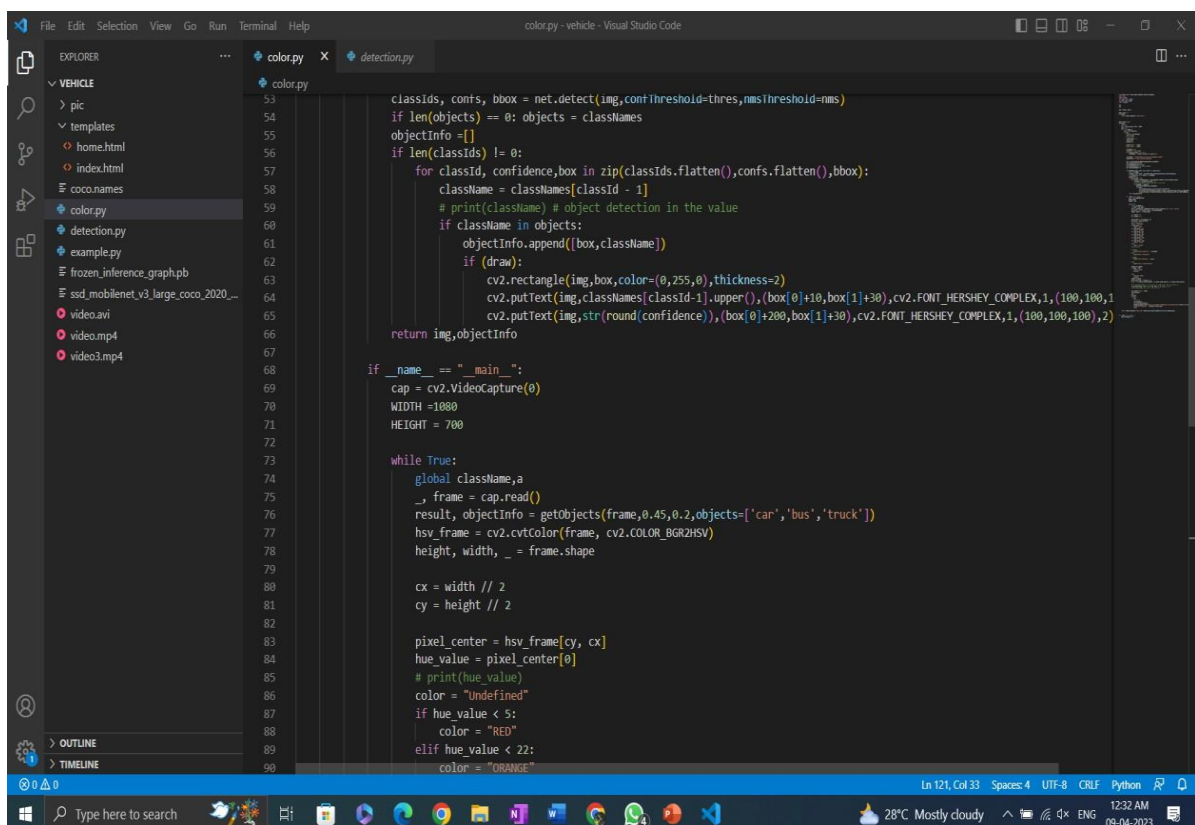
```

Serial.print(s) ;
Serial.println(" km/s");
delay(200);

f = 0;
t = 0;
}
// Serial.print(" S: ");
// Serial.print ( s ) ;
// Serial.print(" D : ");
// Serial.println ( d ) ;
delay(100);
}

```

PROGRAM CODE USED FOR HARDWARE:



Python code used in visual studio code

```

from flask import Flask, render_template, redirect, Response

```

```

import cv2
import serial
from time import sleep
from PIL import Image
import requests
b=0
a=0
app = Flask(__name__)
@app.route("/")
def home():
    return render_template("index.html")
@app.route("/new")
def data():
    global b
    ser = serial.Serial('COM5', 9600)
    data = []
    for i in range(1):
        line = ser.readline()
        if line:
            string = line.decode()
num = string
        # print(num)
        vs=num.split(',')
        count=vs[0]
        speed=vs[1]
    print("Count : ",count)
    print("Speed : ",speed)
    classNames = []
    classFile = "coco.names"
        with open(classFile,"rt") as f:
    classNames = f.read().rstrip("\n").split("\n")
    configPath = "ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt"
    weightsPath = "frozen_inference_graph.pb"
        net = cv2.dnn_DetectionModel(weightsPath,configPath)

```

```

net.setInputSize(320,320)
net.setInputScale(1.0/ 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)

def getObjects(img, thres, nms, draw=True, objects=[]):
    global className
    classIds, confs, bbox = net.detect(img,confThreshold=thres,nmsThreshold=nms)
    if len(objects) == 0: objects = className
    objectInfo=[]
    if len(classIds) != 0:
        for classId, confidence,box in
zip(classIds.flatten(),confs.flatten(),bbox):
            className = className[classId - 1]
            # print(className) # object detection in the value
            if className in objects:
objectInfo.append([box,className])
                if (draw):
                    cv2.rectangle(img,box,color=(0,255,0),thickness=2)
                    cv2.putText(img,className[classId-
1].upper(),(box[0]+10,box[1]+30),cv2.FONT_HERSHEY_COMPLEX,1,(100,100,10
0),2)

cv2.putText(img,str(round(confidence)),(box[0]+200,box[1]+30),cv2.FONT_HERSHEY_COMPLEX,1,(100,100,100),2)
                    return img,objectInfo

if __name__ == "__main__":
    cap = cv2.VideoCapture(0)
    WIDTH =1080
    HEIGHT = 700
    while True:
        global className,a
        _, frame = cap.read()

```

```

        result,                                objectInfo                                =
getObjects(frame,0.45,0.2,objects=['car','bus','truck'])
hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        height, width, _ = frame.shape

        cx = width // 2
        cy = height // 2
pixel_center = hsv_frame[cy, cx]
hue_value = pixel_center[0]
        # print(hue_value)
color = "Undefined"
        if hue_value< 5:
color = "RED"
elifhue_value< 22:
color = "ORANGE"
elifhue_value< 33:
color = "YELLOW"
elifhue_value< 78:
color = "GREEN"
elifhue_value< 131:
color = "BLUE"
elifhue_value< 170:
color = "VIOLET"
        else:
color = "BLACK"
        # print(color)
        if object:
print("object Detection : ",className)
        else:
print("Doesn't Detected")
        if color:
print("Color Detection : ",color)
        else:
print("Color is Not Detected")

```

```

        speed=float(speed)
        if speed > 100:
            b="Over speed"
            print(b)
        else:
            b="Normal Speed"
            print(b)
        speed=str(speed)
    pixel_center_bgr = frame[cy, cx]
    b, g, r = int(pixel_center_bgr[0]), int(pixel_center_bgr[1]),
int(pixel_center_bgr[2])

    # cv2.rectangle(frame, (cx - 220, 10), (cx + 200, 120), (255, 255, 255),
-1)

    cv2.putText(frame, color, (cx - 200, 100), 0, 0, (b, g, r), 1)
    # cv2.circle(frame, (cx, cy), 5, (0, 0, 0), 3)
    cv2.imshow("Color", frame)
    cv2.waitKey(1)
    a+=1
    print(a)
    if a==1:
        a=0
        # print(a)
cap.release()

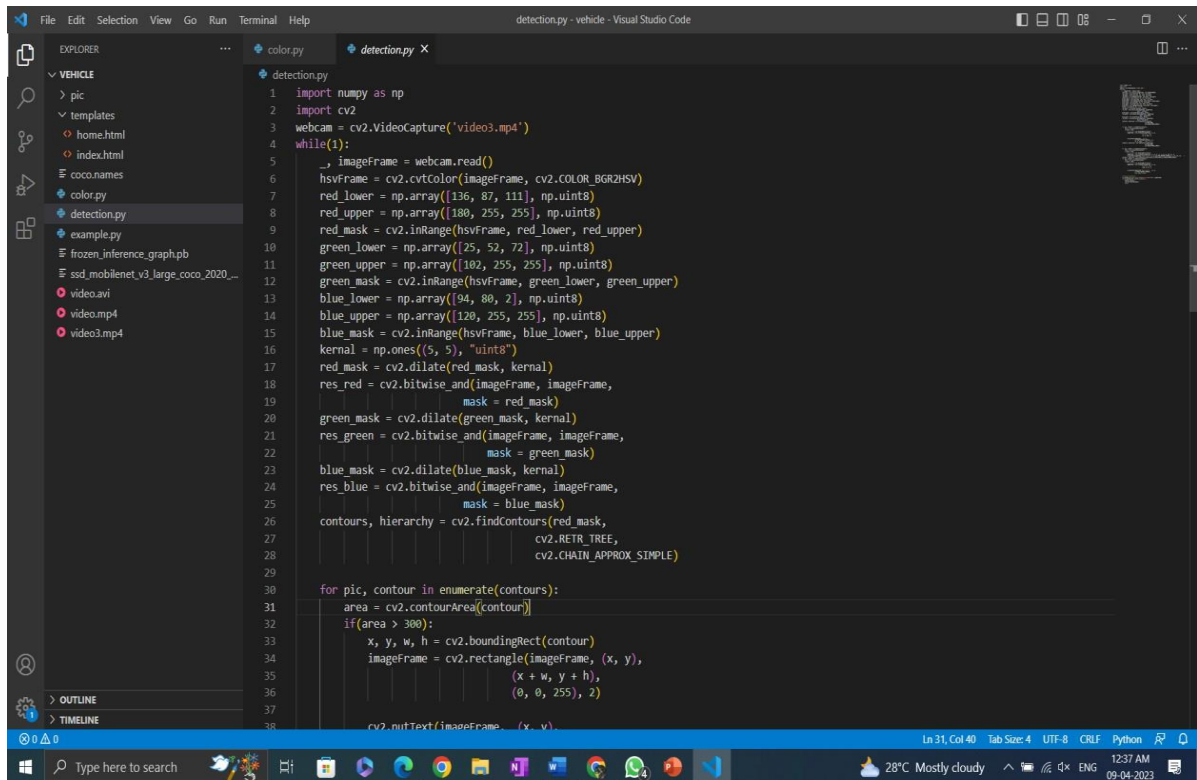
    cv2.destroyAllWindows()

Response=requests.get("https://iotcloud22.in/2046/post_value.php?value1="+cou
nt+"&value2="+speed+"&value3="+className+"&value4="+color)
print("HTTP Response : ",Response.status_code)

break
return
render_template("index.html",name=count,type=className,Color=color,Speed=s
peed)
if __name__ == "__main__":

```

app.run(debug=True)



Python code for image processing

```
import numpy as np
```

```
import cv2
```

```
webcam = cv2.VideoCapture('video3.mp4')
```

```
while(1):
```

```
    _, imageFrame = webcam.read()
```

```
    hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
```

```
    red_lower = np.array([136, 87, 111], np.uint8)
```

```
    red_upper = np.array([180, 255, 255], np.uint8)
```

```
    red_mask = cv2.inRange(hsvFrame, red_lower, red_upper)
```

```
    green_lower = np.array([25, 52, 72], np.uint8)
```

```
    green_upper = np.array([102, 255, 255], np.uint8)
```

```
    green_mask = cv2.inRange(hsvFrame, green_lower, green_upper)
```

```
    blue_lower = np.array([94, 80, 2], np.uint8)
```

```
    blue_upper = np.array([120, 255, 255], np.uint8)
```

```
    blue_mask = cv2.inRange(hsvFrame, blue_lower, blue_upper)
```

```
    kernal = np.ones((5, 5), 'uint8')
```

```

red_mask = cv2.dilate(red_mask, kernal)
res_red = cv2.bitwise_and(imageFrame, imageFrame,
                           mask = red_mask)

green_mask = cv2.dilate(green_mask, kernal)
res_green = cv2.bitwise_and(imageFrame, imageFrame,
                             mask = green_mask)

blue_mask = cv2.dilate(blue_mask, kernal)
res_blue = cv2.bitwise_and(imageFrame, imageFrame,
                            mask = blue_mask)

contours, hierarchy = cv2.findContours(res_red,
cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area > 300):
        x, y, w, h = cv2.boundingRect(contour)
        imageFrame = cv2.rectangle(imageFrame, (x, y),
(x + w, y + h),
(0, 0, 255),
cv2.putText(imageFrame, (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1.0,
(0, 0, 255)) # "Red Colour",
        contours, hierarchy = cv2.findContours(res_green,
cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area > 300):
        x, y, w, h = cv2.boundingRect(contour)
        imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y +
h), (0, 255, 0), 2)
        cv2.putText(imageFrame, "Green Colour", (x,
y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 0)) #

```

```

        contours, hierarchy =
cv2.findContours(blue_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area > 300):
            x, y, w, h = cv2.boundingRect(contour)
            image Frame = cv2.rectangle(imageFrame, (x, y),
                                           (x + w, y + h),
                                           (255, 0, 0), 2)
            cv2.putText(image Frame,"Blue Colour", (x, y),
                        cv2.FONT_HERSHEY_SIMPLEX,
                        1.0, (255, 0, 0))#

# Program Termination
cv2.imshow("Multiple Color Detection in Real-Time", imageFrame)
if cv2.waitKey(10) & 0xFF == ord('q'):
    webcam.release()
    cv2.destroyAllWindows()
    break

```

CONFERENCE ATTENDED

"Recent Advances in Communicative Electronics (NCRACE 2023)" organized by the Department of Electronics and Communication Engineering, SRM TRP Engineering College, Tiruchirappalli.

Place: SRM TRP Engineering College, Tiruchirappalli

During: March 09 – 10, 2023

CERTIFICATE

1. Certificate - RAGHUL . S



2.Certificate - HAMEED FAZIL. I

