International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

AI-Based Vehicle Type Detection and Monitoring in a Traffic System

Dr.N.Muthu Selvakumar[1], S.Amutha[2], S.A.Harinie[3], N.Santhiya[4]

[1]Assistant Professor, Department of Civil Engineering, Kongunadu College of Engineering and Technology (Autonomous), Tiruchirappalli - 621 215, Tamil Nadu, India

[2,3,4] UG Student, Department of Civil Engineering, Kongunadu College of Engineering and Technology (Autonomous), Tiruchirappalli - 621 215, Tamil Nadu, India.

How to Cite this Article:
N.Santhiya, , A.Harinie, & S.Amutha, (2026). AI-Based Vehicle Type Detection and Monitoring in a Traffic System. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05). https://doi.org/10.55041/ijcope.v2i4.1025

License:
This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.
© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.

https://doi.org/ 10.55041/ijcope.v2i4.1025

ABSTRACT

This project introduces a cost-effective system for real-time traffic monitoring using AI-Based Vehicle Type Detection and Monitoring System. The system utilizes IR sensors to detect vehicle movement on the road and classify different types of vehicles such as cars and bikes. When a vehicle passes through the sensing area, the sensors capture the signal and send it to the Arduino microcontroller for processing. The Arduino processes the sensor data and identifies vehicle type based on predefined logic. The system continuously monitors traffic flow and updates the vehicle count in real time. The results are displayed through the serial monitor, enabling easy observation and analysis. This solution provides an efficient method for automatic vehicle detection, reducing manual effort and improving accuracy in traffic monitoring. The mini- project focuses on the design and implementation of a sensor-based vehicle detection system, demonstrating a practical and low-cost approach for traffic analysis. The system can be further enhanced by integrating advanced AI techniques such as image processing and machine learning for improved vehicle classification and smart traffic management. This project highlights the importance of intelligent systems in modern transportation and supports the development of smart city infrastructure.

Keywords : Vehicle Detection, Traffic Monitoring, Arduino, IR Sensor, AI-Based System, Smart Traffic, Vehicle Counting.

1. INTRODUCTION

1.1 GENERAL

In recent years, the number of vehicles on roads has increased rapidly due to urbanization, population growth, and economic development. This has led to several challenges such as traffic congestion, increased travel time, fuel wastage, and road accidents. Efficient traffic monitoring and management have become essential to ensure smooth transportation and public safety. However, in many areas, traffic is still monitored manually, which is time-consuming, less accurate, and not suitable for real-time analysis.

Manual traffic monitoring requires continuous human effort and is prone to errors, especially during peak hours. It becomes difficult to manage traffic effectively without proper data and automation. Therefore, there is a need for a smart and automated system that can monitor traffic conditions efficiently and provide accurate real-time information.

This project presents an AI-Based Vehicle Type Detection and Monitoring System, which aims to improve traffic management using simple and cost- effective technology. The system uses sensors and a microcontroller to detect vehicles and monitor traffic flow automatically. It reduces human effort and improves accuracy in data collection.

The proposed system is designed to be simple, reliable, and easy to implement. It can be used in various applications such as traffic signals, toll booths, parking areas, and road intersections. By providing real-time vehicle detection and monitoring, the system helps in better traffic control and decision- making.

1.2 DESCRIPTION

Following the need for an efficient and automated traffic monitoring system, this project focuses on the development of an AI-Based Vehicle Type Detection and Monitoring System using simple and cost-effective components. The system is designed to automatically detect vehicles and monitor their movement in real time, thereby improving traffic management and reducing manual effort.

The proposed system mainly uses Infrared (IR) sensors to detect the presence of vehicles on the road. These sensors are strategically placed along the road or on a model setup to sense vehicle movement accurately. When a vehicle passes in front of the sensor, it interrupts the infrared beam, and the sensor generates a signal. This signal is then sent to the Arduino UNO microcontroller, which acts as the central processing unit of the system.

The Arduino processes the received signal using predefined logic and determines the presence of a vehicle. By using multiple sensors or appropriate positioning, the system can also differentiate between different types of vehicles such as cars and bikes. This classification is based on the detection pattern and timing of sensor signals, making the system simple yet effective.

In addition to vehicle detection, the system also performs vehicle counting. Every time a vehicle is detected, the count is updated automatically. The processed data is displayed in real time through the serial monitor in the Arduino IDE or through output devices such as LEDs or buzzers. This enables easy monitoring and analysis of traffic conditions.

The system is implemented using a small prototype model, where a road is created using cardboard or foam board, and toy vehicles are used to simulate real traffic conditions. This helps in demonstrating the working of the system clearly. In real-world applications, the same concept can be extended using advanced sensors, cameras, and AI algorithms for more accurate vehicle classification and traffic analysis.

Thus, the proposed system provides a simple, efficient, and low-cost solution for real-time vehicle detection and monitoring. It serves as a foundation for developing more advanced intelligent traffic systems and contributes towards improving traffic management, reducing congestion, and enhancing road safety in modern transportation networks.

1.3 OBJECTIVES

- Automated Vehicle Detection:To develop a system that automatically detects vehicles using IR sensors without manual effort.
- Vehicle Type Identification:To differentiate between different types of vehicles such as cars and bikes based on sensor inputs.
- Real-Time Traffic Monitoring:To monitor vehicle movement and traffic flow continuously in real time.
- Improve Traffic Management:To provide useful data for better traffic control and reduce congestion.
- Cost-Effective Solution:To design a low-cost and simple system using Arduino and basic components.
- Future Enhancement Capability:To create a system that can be upgraded using AI, image processing, and smart city technologies.

1.4 SCOPE

The scope of this project is wide and useful in many real-time applications related to traffic monitoring and management. This system can be used in roads, highways, and traffic signals to observe and monitor the movement of vehicles in a simple and effective manner. It helps in understanding how many vehicles are passing through a particular area and gives a clear idea about traffic flow.

This project can also be used in toll booths, parking areas, and entry or exit points where vehicle counting is important. By using this system, the number of vehicles entering and leaving a place can be easily monitored without any manual effort. It can also be installed in colleges, offices, and residential areas for basic traffic observation.

The system is designed as a small prototype, but the same concept can be applied in real-world situations. It can be expanded by adding more sensors to monitor multiple lanes and improve the coverage area. The system can also be connected with display units to show the vehicle count and make the output more visible and useful.

In the future, this project can be improved by adding cameras and advanced technologies to identify more types of vehicles with better accuracy. It can also be connected with smart systems for better traffic control and management. The data collected from this system can be useful for traffic planning and decision-making.

Overall, this project serves as a basic and useful model for developing smart traffic systems. It helps in reducing manual work, improving efficiency, and supporting the development of better transportation systems in the future.

1.5 MATERIALS USED

1.5.1 ELECTRONICS

- Arduino UNO
- IR Sensor
- Bread Board
- Jumper Wires
- Buzzer/LED

- Power Supply

Arduino UNO is the main controller used in this project. It acts as the brain of the system and controls all the operations. It receives input signals from the sensors, processes the data, and gives output based on the program. It is easy to use, low-cost, and widely used for small projects.

Fig. 1.1Arduino Uno & Cable

The IR (Infrared) sensor is used to detect the presence of vehicles. It works by sending infrared rays and receiving them back. When a vehicle passes in front of the sensor, the rays are interrupted, and the sensor detects the object. It then sends a signal to the Arduino.

Fig. 1.2 IR SENSOR

A breadboard is used to connect electronic components without soldering. It makes the circuit setup simple and easy to modify. All the components like sensors and LEDs are connected using a breadboard in this project.

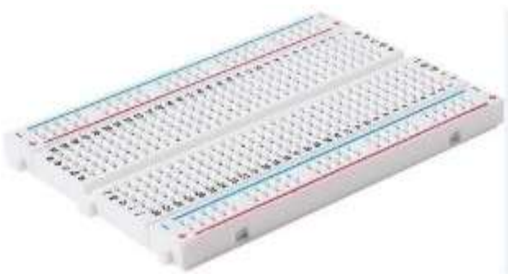


Fig.1.3 BREAD BOARD

Jumper wires are used to connect different components in the circuit. They help in making connections between Arduino, sensors, and other devices. They are flexible and easy to use.

Fig.1.4 JUMPER WIRE

1.6 MECHANISM

The mechanism of the AI-Based Vehicle Type Detection and Monitoring System is designed to work in a simple and automatic way using sensors and a microcontroller. The main aim of the system is to detect vehicles when they pass through a particular area and process that information to monitor traffic flow. The system does not require any manual work and operates continuously once it is powered.

The main components used in this mechanism are the IR sensor, Arduino UNO microcontroller, breadboard, jumper wires, and output devices such as LED or buzzer. These components are connected properly to form a complete working system. The Arduino acts as the central control unit, which manages all the input and output operations.

When the system is switched on using a USB power supply, all the components start functioning. The IR sensor begins to emit infrared rays continuously. These rays are invisible to the human eye but are used for detecting objects. Under normal conditions, when no vehicle is present, the infrared rays travel without any interruption, and the sensor remains in its default state.

As soon as a vehicle comes in front of the IR sensor, the infrared rays are blocked or reflected. This change is detected by the sensor, which then produces an electrical signal. This signal indicates that an object has been detected. The sensor immediately sends this signal to the Arduino microcontroller through the connected input pin.

The Arduino reads the signal and processes it according to the program written in the Arduino IDE. The program contains simple instructions to identify whether a vehicle is present or not. Once the Arduino confirms the detection, it performs the required action such as increasing the vehicle count or activating an output device.

The counting mechanism is an important part of this system. Every time a vehicle is detected, the Arduino increases the count by one. This helps in keeping track of the total number of vehicles passing through that area. The count is displayed in the serial monitor, which allows the user to observe the data in real time.

In some cases, more than one IR sensor can be used to improve the system performance. By placing multiple sensors at different positions, the system can identify different types of vehicles. For example, a small vehicle like a bike may trigger only one sensor, while a larger vehicle like a car may trigger multiple sensors. This method helps in basic classification of vehicles without using complex technology.

After processing the input signal, the Arduino sends an output signal to devices like LED or buzzer. When a vehicle is detected, the LED glows or the buzzer produces sound. This gives a clear indication that the system is working properly. It also helps in visually understanding the detection process.

The system works continuously as long as it is connected to the power supply. Every time a vehicle passes, the same process is repeated. This continuous operation makes the system suitable for real-time traffic monitoring. It provides accurate and updated information about vehicle movement.

The mechanism of this project is simple but effective. It does not require expensive components or complicated setup. It is easy to build and understand, making it suitable for students and beginners. Even though it is a basic model, it clearly explains the working of an automated traffic monitoring system.

This system can also be improved in the future. Advanced technologies like cameras, image processing, and machine learning can be added to increase accuracy and detect more types of vehicles. The system can also be connected to a display unit or cloud system for remote monitoring.

Another important advantage of this mechanism is its flexibility. The system can be modified and expanded based on requirements. Additional sensors can be added for multiple lanes, and the system can be integrated with other traffic control systems. This makes it suitable for both small-scale and large-scale applications.

Overall, the mechanism of this project demonstrates how simple electronic components can be used to develop a useful and intelligent system. It helps in reducing manual effort, improving accuracy, and supporting better traffic management. The project also provides a basic idea for developing advanced smart traffic systems in the future.

2 LITERATURE REVIEW

Jingpei Dan (2024) proposed a deep learning-based approach for vehicle detection and classification in traffic systems. Their method uses Convolutional Neural Networks (CNN) to identify vehicles from real-time video streams. The system improves accuracy and performs well under different environmental conditions such as low light and heavy traffic.

Ittipong Khemapech (2024) introduced an intelligent traffic monitoring system using wireless camera networks. Their system collects real-time traffic data and applies machine learning algorithms to analyze vehicle movement and traffic density. It also provides alerts for congestion and abnormal traffic conditions.

Muhammad Riyansyah (2024) developed a dynamic vehicle classification system based on motion and speed analysis. By processing video frames and extracting features, their system accurately distinguishes between different types of vehicles and improves traffic monitoring efficiency.

Mahta Zakaria (2024) worked on enhancing traffic monitoring using edge computing and artificial intelligence. Their system performs real-time image processing on edge devices, reducing latency and improving system performance without relying heavily on cloud servers.

Joseph Redmon (2016) introduced the YOLO (You Only Look Once) algorithm for real-time object detection. This method processes images in a single step, making it highly efficient and suitable for vehicle detection in traffic systems.

Wei Liu proposed the Single Shot Detector (SSD) algorithm, which allows simultaneous detection and classification of multiple objects. This method is widely used in traffic monitoring due to its speed and accuracy.

Sivaraman and Trivedi (2013) presented a comprehensive survey on vision-based vehicle detection systems. Their study highlights various techniques and challenges involved in detecting vehicles using image processing methods.

Navneet Dalal and Bill Triggs (2005) introduced the Histogram of Oriented Gradients (HOG) method for object detection. This technique is useful in identifying object shapes and has been applied in early vehicle detection systems.

Ross Girshick (2014) developed region-based convolutional neural networks (R- CNN) for accurate object detection. Their work improved detection accuracy and influenced many modern vehicle detection models.

Kaiming He (2016) proposed deep residual networks (ResNet), which improved feature extraction in deep learning models. This advancement helps in better classification of vehicles in complex traffic scenes.

Alex Krizhevsky (2012) introduced a deep convolutional neural network model for image classification. Their work played a major role in the development of modern AI-based detection systems.

OpenCV provides an open-source platform for image processing and computer vision applications. It is widely used for implementing vehicle detection and tracking systems.

ESP8266 Wi-Fi Module enables wireless data transmission in embedded systems. It is commonly used in traffic monitoring systems to send real-time data to cloud platforms.

Paul Viola and Michael Jones (2001) proposed a rapid object detection method using Haar-like features. This technique is one of the earliest methods used in real-time object detection.

Marcel Teichmann (2018) developed real-time semantic reasoning systems for autonomous driving. Their work contributes to understanding complex traffic environments.

Andreas Geiger (2013) introduced the KITTI dataset, which is widely used for training and evaluating vehicle detection models in real-world scenarios.

World Health Organization (2018) reported on global road safety issues, highlighting the need for advanced traffic monitoring systems to reduce accidents.

Ministry of Road Transport and Highways provided reports on traffic and road safety, emphasizing the importance of intelligent transportation systems in India.

3 EXISTING SYSTEM

3.1 MANUAL TRAFFIC MONITORING

Manual traffic monitoring is one of the traditional methods used for controlling and managing traffic. In this system, traffic police officers are deployed at road intersections, highways, and busy areas to observe and regulate the movement of vehicles. They manually control traffic signals, monitor congestion, and ensure smooth vehicle flow.

In manual monitoring, officers visually inspect the traffic conditions and make decisions based on their experience. They also identify traffic violations such as signal jumping, over-speeding, and wrong-way driving. This method is commonly used in areas where automated systems are not available.

However, manual traffic monitoring requires continuous human effort and is highly dependent on the efficiency of the personnel. It becomes difficult to manage traffic effectively during peak hours, bad weather conditions, or at multiple locations simultaneously. Human errors, fatigue, and lack of real-time data analysis reduce the overall efficiency of the system.

3.2 SENSOR-BASED TRAFFIC SYSTEMS

Sensor-based traffic systems are an improvement over manual monitoring methods. These systems use various types of sensors such as inductive loop sensors, infrared sensors, and ultrasonic sensors to detect the presence and movement of vehicles on roads.

Inductive loop sensors are installed beneath the road surface and detect vehicles based on changes in electromagnetic fields. Infrared sensors use heat signals to identify moving vehicles, while ultrasonic sensors measure the distance and movement of objects. These sensors collect data such as vehicle count, speed, and traffic density.

The collected data is sent to a central control system where it is analyzed to manage traffic signals and flow. In some cases, these systems automatically adjust signal timings based on traffic conditions, reducing congestion and improving efficiency.

Although sensor-based systems provide better accuracy than manual methods, they still have limitations. Installation and maintenance costs are high, and sensors may fail due to environmental conditions such as rain, dust, and temperature variations. Additionally, these systems cannot easily classify different types of vehicles with high accuracy.

3.3 LIMITATIONS OF EXISTING SYSTEM

- Despite the use of manual and sensor-based traffic monitoring systems, several limitations still exist:
- High dependency on human effort: Manual systems require continuous monitoring by traffic personnel, leading to fatigue and human errors.
- Limited accuracy: Sensor-based systems cannot accurately classify vehicle types in complex traffic conditions.
- High installation cost: Installing sensors and maintaining them is expensive and time-consuming.
- Lack of real-time analysis: Traditional systems do not provide advanced real- time insights or predictive analysis.
- Environmental issues: Sensors may not function properly during bad weather conditions like rain, fog, or extreme heat.
- Limited scalability: Expanding these systems to cover large areas is difficult and costly.

4 METHODOLOGY

4.1 VEHICLE DETECTION PROCESS

The vehicle detection process is the most important part of this project. It explains how the system identifies the presence of vehicles and processes the information in real time. The system is designed in a simple and efficient way using basic electronic components such as an IR sensor, Arduino UNO, and output devices like LED or buzzer. The entire process works automatically without the need for manual control.

Initially, the system is powered using a USB cable connected to the Arduino UNO. Once the power supply is provided, all the components start functioning. The Arduino microcontroller initializes the program and begins reading input signals from the connected sensor. At the same time, the IR sensor starts emitting infrared rays continuously.

The IR sensor plays a major role in detecting vehicles. It consists of an IR transmitter and an IR receiver. The transmitter emits infrared rays, and the receiver detects the reflected rays. Under normal conditions, when no vehicle is present, the infrared rays travel without interruption, and the sensor remains in its default state. This means that no signal is sent to the Arduino.

When a vehicle passes in front of the IR sensor, the infrared rays are either blocked or reflected back to the receiver. This change in the signal is detected by the sensor immediately. The sensor then generates an output signal indicating that an object has been detected. This signal is sent to the Arduino through the connected input pin.

The Arduino UNO acts as the brain of the system. It continuously monitors the signals coming from the IR sensor. When it receives a signal indicating the presence of a vehicle, it processes the data according to the program written in the Arduino IDE. The program contains instructions to identify the change in sensor value and take appropriate action.

Once the Arduino confirms the detection of a vehicle, it performs several operations. One of the main functions is counting the number of vehicles. Each time a vehicle is detected, the count value is increased by one. This counting process helps in monitoring the traffic flow in a particular area.

In addition to counting, the Arduino also activates output devices such as LED or buzzer. When a vehicle is detected, the LED glows or the buzzer produces a sound. This provides a clear indication that the system is working properly. It also helps the user to visually understand when a vehicle is being detected.

The detected information can also be displayed using the serial monitor in the Arduino IDE. The serial monitor shows the number of vehicles detected in real time. This feature is useful for analyzing traffic data and understanding vehicle movement patterns.

The system works continuously in a loop. This means that after detecting one vehicle, it immediately returns to the initial state and waits for the next vehicle. This continuous operation makes the system suitable for real-time applications. It ensures that every vehicle passing through the detection area is counted and recorded.

The placement of the IR sensor is also an important factor in the detection process. The sensor should be placed at an appropriate height and position so that it can easily detect vehicles passing in front of it. Proper alignment ensures accurate detection and reduces errors.

In some cases, more than one IR sensor can be used to improve the system performance. Multiple sensors can help in detecting vehicles more accurately and can also be used for basic classification. For example, a small vehicle may trigger only one sensor, while a larger vehicle may trigger multiple sensors. This method helps in identifying different types of vehicles to some extent.

Another important aspect of the detection process is timing. The Arduino program can include a small delay to avoid multiple counts for a single vehicle. Without proper timing control, the system may count the same vehicle more than once. Therefore, timing plays a key role in ensuring accurate results.

The system is designed to be simple and cost-effective. It does not require complex components or high-level programming. This makes it suitable for students and beginners who want to understand the basic concept of vehicle detection and traffic monitoring.

Even though the system is simple, it provides a clear idea of how automated traffic systems work. It reduces the need for manual monitoring and improves accuracy. It also provides real-time data, which is useful for traffic analysis and management.

The detection process can be further improved by adding advanced technologies. For example, cameras and image processing techniques can be used to identify different types of vehicles more accurately. The system can also be connected to a cloud platform for remote monitoring and data storage.

Overall, the vehicle detection process in this project is efficient, reliable, and easy to implement. It demonstrates how basic electronic components can be used to develop a smart traffic monitoring system. The system works automatically, provides real-time results, and helps in understanding the concept of vehicle detection in a simple way.

© 2026 The Author(s). Published by IJCOPE Journal. Website: https://ijcope.org/

4.2 BLOCK DIAGRAM

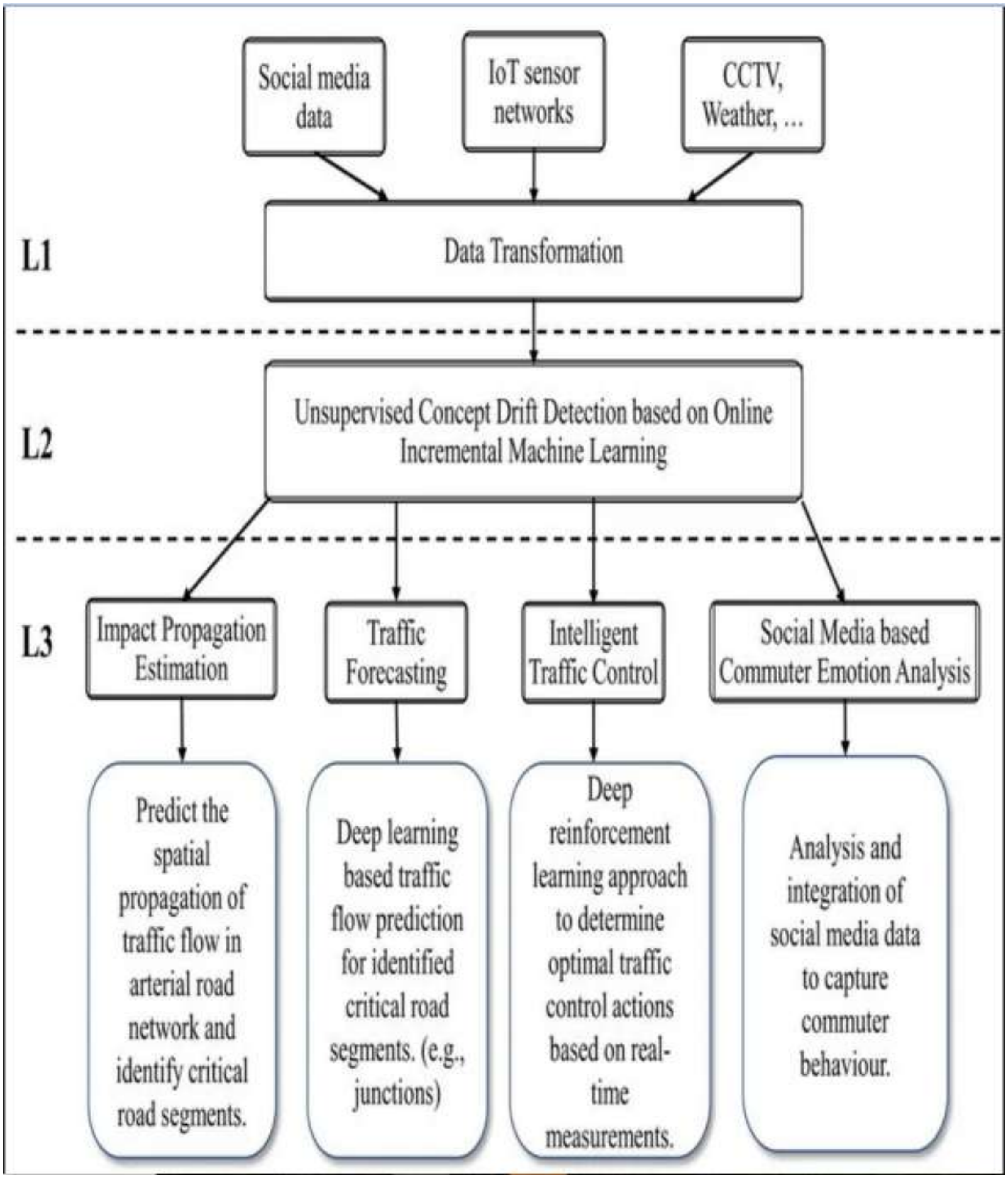


Fig.4.1 AI BASED VEHICLE TYPE DETECTION AND MONITORING IN TRAFFIC SYSTEM

4.2.1 Explanation of The Block Diagram

The block diagram represents the overall structure and working of the vehicle detection system. It shows how different components are connected and how data flows from input to output. The system mainly consists of four important sections: input unit, processing unit, output unit, and power supply.

The input unit of the system is the IR sensor. It is responsible for detecting the presence of vehicles. The IR sensor consists of an infrared transmitter and receiver. The transmitter continuously emits infrared rays, while the receiver detects the reflected rays. When no vehicle is present, the rays travel normally, and the sensor remains in its default condition. When a vehicle passes in front of the sensor, the infrared rays are interrupted or reflected. This change is detected by the sensor, and it generates a signal indicating the presence of a vehicle.

The signal from the IR sensor is then sent to the processing unit, which is the Arduino UNO. The Arduino acts as the brain of the system. It receives the input signal from the sensor and processes it based on the program stored in it. The program is written using Arduino IDE and uploaded to the microcontroller.

The Arduino continuously checks the input signal from the IR sensor. When it detects a change in the signal, it identifies that a vehicle has been detected. Based on this input, the Arduino performs necessary operations such as increasing the vehicle count and activating output devices. The processing unit plays a key role in decision-making and controls the overall functioning of the system.

The output unit of the system consists of LED and buzzer. These components are used to indicate the detection of a vehicle. When the Arduino receives a signal from the sensor, it sends an output signal to the LED or buzzer. The LED glows or the buzzer produces sound, which indicates that a vehicle has been detected. This helps in providing a clear and immediate response to the user.

In addition to LED and buzzer, the system can also display output through the serial monitor. The Arduino sends data to the serial monitor, where the number of detected vehicles can be observed in real time. This feature is useful for monitoring and analyzing traffic flow.

The power supply unit is another important part of the block diagram. It provides the required electrical energy for the system to function. In this project, a USB power supply is used to power the Arduino UNO. The Arduino, in turn, supplies power to the IR sensor and other components. Proper power supply ensures smooth and continuous operation of the system.

The working of the block diagram follows a simple sequence. First, the IR sensor detects the presence of a vehicle. Then, the signal is sent to the Arduino UNO. The Arduino processes the signal and decides the output action. Finally, the output is displayed through LED, buzzer, or serial monitor. This sequence repeats continuously, allowing real-time detection of vehicles.

The block diagram helps in understanding the system in a simple and visual way. It shows how each component is connected and how they interact with each other. It also helps in identifying the role of each component in the system.

One of the advantages of this system is its simplicity. The components used are basic and easily available. The connections are simple, and the system does not require complex setup. This makes it suitable for students and beginners.

The system is also cost-effective compared to advanced traffic monitoring systems. It provides a basic idea of automation and real-time monitoring without requiring expensive equipment. It can be easily modified or expanded by adding more sensors or output devices.

The block diagram also provides flexibility for future improvements. For example, additional sensors can be added to improve detection accuracy. Display units such as LCD screens can be included for better output visualization. Wireless communication modules can also be added for remote monitoring.

Overall, the block diagram clearly explains the working principle of the vehicle detection system. It shows how input, processing, and output units are connected to perform the detection process. The system is simple, reliable, and effective for real-time vehicle monitoring.

4.3 CIRCUIT DIAGRAM

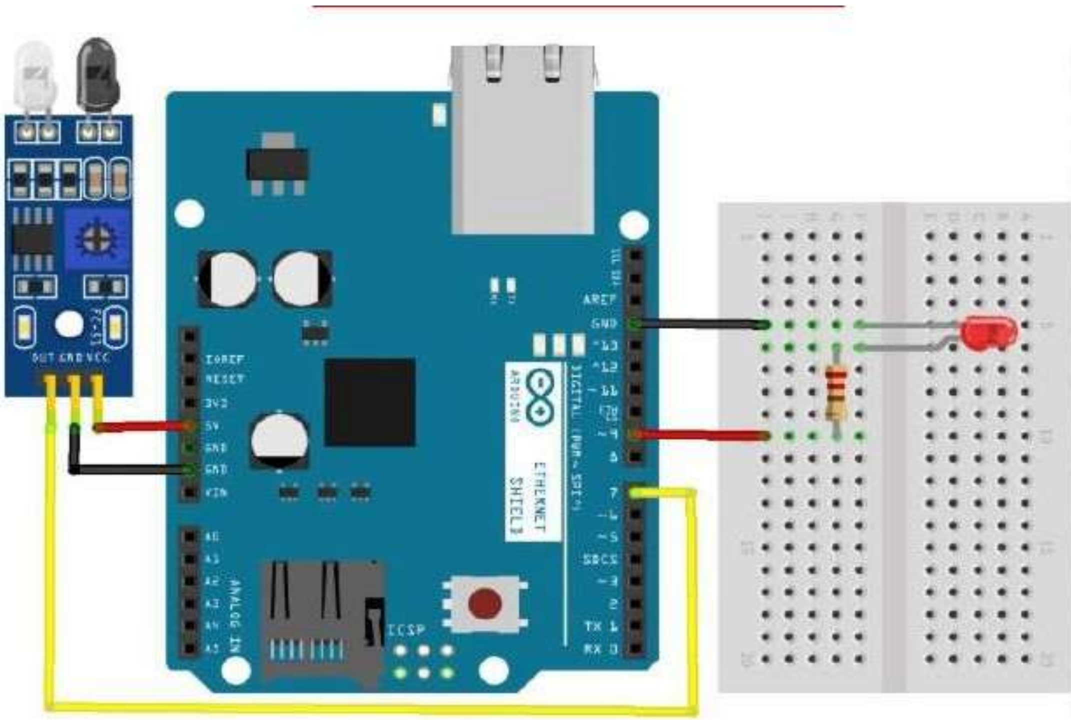


Fig.4.2 Circuit Diagram

4.3.1 Explanation of the Circuit Diagram

The circuit diagram represents the electrical connections between all the components used in the vehicle detection system. It shows how the IR sensor, Arduino UNO, LED, buzzer, and other components are connected to work as a complete system. Proper understanding of the circuit diagram is important to ensure correct assembly and smooth functioning of the project.

The main component in the circuit is the Arduino UNO, which acts as the central processing unit. All input and output devices are connected to the Arduino. It receives signals from the sensor, processes them, and provides output accordingly. The Arduino is powered using a USB cable, which supplies the required voltage for the entire system.

The IR sensor is used as the input device in the circuit. It has three main pins: VCC, GND, and OUT (output). The VCC pin is connected to the 5V supply of the Arduino, and the GND pin is connected to the ground. The output pin of the sensor is connected to one of the digital input pins of the Arduino, usually pin number 2 or 3.

The IR sensor works by emitting infrared rays and detecting their reflection. When a vehicle passes in front of the sensor, the infrared rays are interrupted or reflected, and the sensor produces a signal. This signal is sent to the Arduino through the output pin.

The Arduino continuously reads the input signal from the sensor. When it detects a change in the signal, it identifies that a vehicle is present. Based on this input, the Arduino sends output signals to the connected devices such as LED or buzzer.

The LED is used as a visual indicator in the circuit. It is connected to one of the digital output pins of the Arduino, usually pin number 13. A resistor is connected in series with the LED to limit the current and prevent damage. The other end of the LED is connected to the ground.

When the Arduino detects a vehicle, it sends a HIGH signal to the LED pin, causing the LED to glow. When no vehicle is detected, the LED remains OFF. This simple indication helps in understanding whether the system is working properly.

A buzzer can also be used as an output device. It is connected to another digital output pin of the Arduino. When a vehicle is detected, the Arduino sends a signal to the buzzer, which produces sound. This provides an audio indication along with the visual indication from the LED.

The breadboard is used to make connections between components without soldering. It helps in arranging the circuit neatly and allows easy modifications. Jumper wires are used to connect different components on the breadboard to the Arduino.

The power supply is provided through the USB cable connected to the Arduino. The Arduino distributes power to all connected components such as the IR sensor and LED. It is important to ensure proper voltage levels to avoid damage to components.

The circuit works in a continuous loop. The IR sensor detects the presence of a vehicle and sends a signal to the Arduino. The Arduino processes this signal and activates the LED or buzzer. After that, the system returns to its initial state and waits for the next vehicle.

Proper wiring is very important in the circuit. Incorrect connections may lead to malfunction or damage to components. Therefore, each connection should be checked carefully before powering the system.

One important aspect of the circuit is the use of a resistor with the LED. Without a resistor, excess current may flow through the LED and damage it. The resistor ensures safe operation by controlling the current.

Another important factor is grounding. All components must share a common ground for the circuit to work properly. The GND pin of the Arduino should be connected to the ground of all components.

The circuit is simple and easy to implement. It does not require complex components or advanced knowledge. This makes it suitable for beginners and students who are learning basic electronics and Arduino programming.

The circuit can also be modified or expanded. Additional sensors can be added to improve detection accuracy. More output devices can be connected for better indication. The system can also be integrated with display units or communication modules for advanced applications.

Overall, the circuit diagram explains how different components are connected to perform vehicle detection. It provides a clear understanding of the electrical connections and working of the system. The circuit is reliable, cost- effective, and suitable for real-time vehicle monitoring applications.

5. HARDWARE DESCRIPTION

5.1 HARDWARE COMPONENTS

The AI-based vehicle detection system consists of several hardware components that work together to detect and monitor vehicles efficiently. Each component plays a specific role in the functioning of the system.

5.1.1 IR SENSOR

The Infrared (IR) sensor is used to detect the presence of vehicles. It works based on infrared radiation. When a vehicle passes in front of the sensor, the IR rays are reflected back and detected by the receiver. This change in signal indicates the presence of a vehicle.

IR sensors are simple, cost-effective, and suitable for short-distance detection.They are widely used in basic traffic monitoring systems and automation applications.

5.1.2 ARUINO UNO (MICROCONTROLLER)

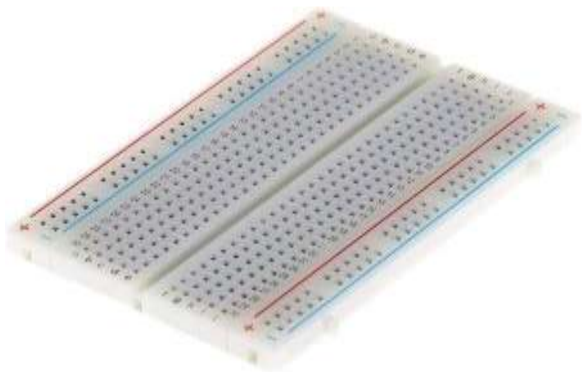
The Arduino UNO is the main controller of the system. It processes the signals received from the IR sensor and controls the output devices such as LED or buzzer.

It is based on the ATmega328P microcontroller and provides digital and analog input/output pins. The Arduino reads the sensor data, processes it using a programmed algorithm, and performs actions like turning on an LED or sending signals to other modules.

5.1.3 BREADBOARD

A breadboard is used to build and test the circuit without soldering. It provides a convenient way to connect multiple components using jumper wires.

The internal connections of the breadboard allow easy distribution of power and signals. It is mainly used for prototyping and experimenting with circuits.



5.1.4 JUMPER WIRES

Jumper wires are used to connect different components in the circuit. They help in establishing electrical connections between the Arduino, sensors, and output devices.

Different types of jumper wires such as male-to-male, male-to-female, and female-to-female are used depending on the connection requirement.

5.1.5 LED INDICATOR / BUZZER

LED or buzzer is used as an output indicator in the system. When a vehicle is detected by the IR sensor, the Arduino sends a signal to turn ON the LED or activate the buzzer.

The LED provides visual indication, while the buzzer gives an audio alert.

This helps in understanding whether the system is detecting vehicles correctly.

5.2 SELECTION OF COMPONENTS

The selection of components plays an important role in the successful implementation of the vehicle detection system. The components used in this project are chosen based on factors such as cost, availability, ease of use, reliability, and suitability for the application. Since this is a small-scale project, simple and easily available components are selected to ensure that the system is easy to build and understand.

The main component used in this project is the Arduino UNO microcontroller. It is selected because it is simple to use and widely available. Arduino UNO is suitable for beginners as well as students who are new to electronics and programming. It provides sufficient input and output pins to connect sensors and other devices. It also supports easy programming through the Arduino IDE. Due to its low cost and flexibility, Arduino UNO is an ideal choice for this project.

The IR sensor is selected as the input device for vehicle detection. It is chosen because of its simplicity and effectiveness in detecting objects. The IR sensor can detect the presence of vehicles by using infrared rays, which makes it suitable for short-range detection. It provides quick response and consumes less power. Compared to other sensors, IR sensors are cost-effective and easy to interface with the Arduino. This makes them a good choice for this application.

A breadboard is used in the project to make circuit connections. It is selected because it allows connections without soldering. This makes the circuit setup flexible and easy to modify. If any changes are required, components can be easily removed and reconnected. This is especially useful during testing and development of the project.

Jumper wires are used to connect different components in the circuit. They are selected because they are flexible and easy to handle. They help in making proper connections between the Arduino, sensors, and output devices. Different types of jumper wires such as male-to-male and male-to-female wires are used based on the requirement.

LED and buzzer are selected as output devices in this project. The LED is used to provide visual indication when a vehicle is detected. It is simple, consumes less power, and is easy to connect. A buzzer is used to provide an audio indication. When a vehicle is detected, the buzzer produces sound, which helps in understanding the system operation clearly. These output devices are chosen because they are simple and effective for demonstration purposes.

The USB power supply is used to provide power to the system. It is selected because it is easily available and can be connected to a laptop or charger. It provides a stable power supply to the Arduino and other components. This ensures continuous operation of the system without interruption.

Another important factor considered in selecting components is cost. All the components used in this project are low-cost and easily available in the market. This makes the project affordable and suitable for students. The use of simple components also reduces the complexity of the system.

Reliability is also an important factor in component selection. The selected components are reliable and provide consistent performance. The Arduino UNO is widely used and tested in many projects, which makes it a dependable choice. Similarly, IR sensors are commonly used for object detection and provide stable results.

Ease of use is another important consideration. The components are selected in such a way that they can be easily connected and used without requiring advanced technical knowledge. This makes the project easy to understand and implement.

The components are also selected based on their compatibility. All the components used in the project are compatible with each other and can work together without any issues. The Arduino provides the required voltage and supports easy interfacing with sensors and output devices.

The selection of components also considers future expansion. The system can be upgraded by adding more sensors or output devices. Additional modules such as display units or communication devices can also be connected to the Arduino. This flexibility makes the system more useful for future improvements.

Overall, the selection of components in this project is done carefully to achieve a balance between cost, performance, and simplicity. The chosen components help in building an efficient and reliable vehicle detection system. They also make the project easy to implement and understand, especially for beginners.

5.3 IMPORTANCE OF SYSTEM DESIGN

System design plays a very important role in the successful development of any project. It refers to the planning and arrangement of different components in such a way that the system works smoothly and efficiently. In this vehicle detection project, proper system design ensures accurate detection, reliable performance, and easy operation.

A well-designed system helps in organizing all the components in a proper manner. It clearly defines how the input, processing, and output units are connected. In this project, the IR sensor acts as the input device, the Arduino UNO acts as the processing unit, and the LED or buzzer acts as the output device. Proper arrangement of these components ensures smooth data flow from one part to another.

One of the main advantages of good system design is improved accuracy. When the components are placed correctly and connected properly, the chances of errors are reduced. For example, correct placement of the IR sensor ensures accurate detection of vehicles. If the sensor is not placed properly, it may fail to detect vehicles or give wrong signals. Therefore, proper design helps in achieving better results.

System design also improves the reliability of the project. A well-planned system works continuously without failure. Proper wiring, correct voltage supply, and stable connections ensure that the system performs consistently. This is important for real-time applications where continuous monitoring is required.

Another important aspect of system design is simplicity. A simple design makes the system easy to understand and implement. In this project, basic components are used, and the connections are kept simple. This helps students and beginners to easily build and test the system. A simple design also reduces the chances of errors during assembly.

System design also helps in easy maintenance and troubleshooting. When the system is properly designed, it becomes easier to identify and fix problems. If any component fails, it can be easily replaced without affecting the entire system. This makes the system more user-friendly and efficient.

Flexibility is another advantage of good system design. A well-designed system can be easily modified or expanded based on requirements. In this project, additional sensors can be added to improve detection accuracy. Output devices such as display units can also be included to enhance the system. This flexibility allows the system to be upgraded in the future.

Proper system design also ensures efficient use of resources. It helps in selecting the right components and avoiding unnecessary complexity. This reduces the overall cost of the project and makes it more economical. Efficient design also reduces power consumption and improves system performance.

Safety is also an important factor in system design. Proper connections and correct voltage levels help in preventing damage to components. For example, using a resistor with the LED prevents excess current flow. Ensuring proper grounding and secure connections also improves safety.

Another important benefit of system design is better performance. When all components work together properly, the system produces accurate and fast results. In this project, the IR sensor quickly detects vehicles, and the Arduino processes the signal without delay. This improves the overall efficiency of the system.

System design also helps in clear understanding of the project. It provides a structured approach, making it easier to explain the working of the system. This is useful during presentations and viva examinations. A well-designed system creates a good impression and shows proper planning.

In addition, good system design supports real-time operation. Since the system is designed to work continuously, it can monitor vehicle movement without interruption. This is important for traffic monitoring applications where real-time data is required.

The design of the system also plays a key role in future development. A properly designed system can be integrated with advanced technologies such as cameras, wireless communication, and cloud storage. This allows the project to be converted into a smart traffic monitoring system.

Overall, system design is an essential part of this project. It ensures smooth operation, improves accuracy, and enhances performance. A well-designed system is reliable, flexible, and easy to use. It also provides a strong foundation for future improvements and advanced applications

6. SOFTWARE DESCRIPTION

6.1 SOFTWARE REQUIRMENT

The Arduino Uno's software ecosystem revolves around its open-source Arduino IDE, available for Windows, macOS, and Linux, enabling users to write and upload code to its ATmega328P microcontroller. Using simplified C/C++, users create sketches to control inputs like moisture sensors and outputs like motors. The IDE's library manager offers pre-built libraries (e.g., Servo, Wire) for seamless hardware integration. A serial monitor aids real-time debugging. The Arduino CLI and platforms like Platform IO support advanced workflows. Code is uploaded via USB with a bootloader. Extensive community resources, tutorials, and third-party libraries enhance functionality for applications like flood detection, making the Uno ideal for prototyping, education, and automation projects.

6.1.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is an open- source software platform used for writing, compiling, and uploading code to Arduino boards. It serves as a bridge between the user and the Arduino hardware, allowing for easy programming of microcontrollers such as the Arduino Uno, Nano, and Mega. The IDE is available for Windows, macOS, and Linux, making it accessible across various operating systems.

At its core, the Arduino IDE uses a simplified version of C++, which is beginner-friendly and ideal for those new to programming microcontrollers. The interface of the Arduino IDE is intuitive, with essential features like the editor for writing code, a console for displaying messages and errors, and a toolbar with buttons for compiling, uploading, and saving sketches (programs). Sketches are

written in the editor window and can include functions like setup()—which runs once when the program starts— and loop(), which runs continuously.

The IDE includes a built-in library manager that allows users to import libraries for specific tasks like controlling sensors, displays, motors, and other peripherals. Libraries simplify coding by providing pre- written functions for complex operations, so users don't have to write everything from scratch. For example, libraries for controlling ultrasonic sensors, LCD displays, or servos are readily available, and users can easily integrate them into their projects.

The Arduino IDE also supports a serial monitor, which allows real-time communication between the Arduino board and the user's computer. This feature is helpful for debugging by displaying output from the Arduino board, such as sensor readings or status messages.

Figure 6.1 Arduino IDE

7. IMPLEMENTATION AND DISCUSSIONS

7.1 REALTIME IMPLEMENT

The real-time implementation of the AI-based vehicle detection system is carried out using a simple and effective prototype model. The main objective of this implementation is to demonstrate how the system works in a real-time environment and how it can automatically detect and monitor vehicles without human intervention.

Initially, all the required components such as Arduino UNO, IR sensor, LED, buzzer, breadboard, and jumper wires are collected. The components are connected properly according to the circuit diagram. The Arduino UNO is connected to a laptop using a USB cable, which provides power as well as supports programming through the Arduino IDE.

After completing the connections, the program is written in the Arduino IDE and uploaded to the Arduino board. The program includes instructions for reading the sensor data, detecting vehicles, and controlling output devices. Once the code is uploaded, the system is ready for testing.

To demonstrate real-time working, a small model is created to simulate a road environment. A simple pathway is arranged using cardboard or any flat surface, and small toy vehicles are used to represent real vehicles. The IR sensor is placed at a suitable position where it can easily detect the movement of vehicles passing through the path.

When the system is powered ON, the Arduino starts executing the program. At the same time, the IR sensor begins to emit infrared rays continuously. When no vehicle is present in front of the sensor, the infrared rays are not interrupted, and the system remains in its normal state.

When a vehicle passes in front of the sensor, the infrared rays are blocked or reflected back to the receiver. This change is detected by the IR sensor, which generates an electrical signal. This signal is sent to the Arduino UNO for processing.

The Arduino continuously checks the input signal from the sensor. When it receives the signal indicating the presence of a vehicle, it processes the data and performs the required action. One of the main actions is increasing the vehicle count. Each time a vehicle is detected, the count value is increased by one.

At the same time, the Arduino activates the output devices such as LED or buzzer. The LED glows or the buzzer produces sound whenever a vehicle is detected. This provides a clear indication that the system is working properly. It also helps the user to visually and audibly understand the detection process.

The output can also be monitored using the serial monitor in the Arduino IDE. The serial monitor displays the number of vehicles detected in real time. This helps in analyzing traffic conditions and understanding vehicle movement patterns.

The system works continuously in a loop. After detecting one vehicle, it returns to its initial state and waits for the next vehicle. This continuous operation ensures that all vehicles passing through the detection area are counted without missing any.

Proper placement of the IR sensor is important for accurate detection. The sensor should be positioned in such a way that it can clearly detect the movement of vehicles without any obstruction. If the sensor is not placed correctly, it may give incorrect readings or fail to detect vehicles.

The real-time implementation shows that the system is simple, efficient, and reliable. It can detect vehicles quickly and provide immediate output. The response time of the system is very fast, which makes it suitable for real-time applications.

One important advantage of this system is that it reduces the need for manual monitoring. The system works automatically and provides accurate results. It also helps in saving time and effort.

However, during implementation, some challenges may be faced. For example, external light sources or environmental conditions may affect the performance of the IR sensor. Proper adjustments and calibration are required to overcome such issues.

The system can be further improved by adding more sensors for multiple lane detection. It can also be integrated with display units to show vehicle count more clearly. Advanced features such as wireless communication and cloud storage can also be added for better monitoring.
Overall, the real-time implementation of this project successfully demonstrates how a simple system can be used for vehicle detection and monitoring. It provides a clear understanding of the working process and shows how automation can be applied in traffic management systems.

7.2 HARDWARE SETUP

- The hardware setup of the vehicle detection system is simple and easy to assemble. It mainly consists of components such as Arduino UNO, IR sensor, LED, buzzer, breadboard, jumper wires, and a USB power supply. All these components are connected properly to form a complete working system.
- Initially, the Arduino UNO is placed on a stable surface, and it acts as the main control unit of the system. All input and output components are connected to the Arduino. The Arduino is powered using a USB cable connected to a laptop or charger, which provides the required voltage for the system.
- The IR sensor is used as the input device in the hardware setup. It is connected to the Arduino using three pins: VCC, GND, and Output. The VCC pin of the sensor is connected to the 5V supply of the Arduino, and the GND pin is connected to the ground. The output pin of the sensor is connected to a digital input pin of the Arduino, usually pin number 2 or 3.
- The placement of the IR sensor is very important. It should be positioned in such a way that it can easily detect vehicles passing in front of it. The sensor should be placed at an appropriate height and angle to ensure accurate detection. Proper alignment helps in reducing errors and improving system performance.
- The LED is used as a visual indicator in the system. It is connected to one of the digital output pins of the Arduino, usually pin number 13. A resistor is connected in series with the LED to control the current flow and prevent damage. The other end of the LED is connected to the ground.

- A buzzer can also be connected as an output device. It is connected to another digital output pin of the Arduino. When a vehicle is detected, the Arduino sends a signal to the buzzer, which produces sound. This provides an audio indication along with the visual indication from the LED.

- The breadboard is used to make all the connections without soldering. It helps in arranging the components neatly and allows easy modification of the circuit. Jumper wires are used to connect different components on the breadboard to the Arduino. These wires make the connections flexible and easy to handle.

- Proper wiring is very important in the hardware setup. All the connections must be checked carefully before powering the system. Loose or incorrect connections may lead to malfunction or improper working of the system.

- The power supply is provided through the USB cable connected to the Arduino. The Arduino distributes power to all connected components such as the IR sensor and LED. It is important to ensure that the correct voltage is supplied to avoid damage to components.

- Once all the connections are completed, the system is tested to ensure proper functioning. The Arduino program is uploaded, and the sensor is checked for detection. When a vehicle passes in front of the sensor, the LED glows or the buzzer sounds, indicating successful detection.

- The hardware setup is designed to be simple and user-friendly. It does not require complex tools or advanced skills. This makes it suitable for students and beginners who are learning basic electronics and Arduino projects.

- The setup can also be modified or expanded based on requirements. Additional sensors can be added to improve detection accuracy. More output devices can be included to enhance system functionality. The system can also be integrated with display units for better visualization.

- Overall, the hardware setup of this project is simple, reliable, and easy to implement. It provides a clear understanding of how different components are connected and how they work together to perform vehicle detection.

8. CONCLUSION

The AI-based vehicle type detection and monitoring system is an effective and innovative solution for improving modern traffic management. With the rapid increase in the number of vehicles on roads, traditional methods of traffic monitoring are becoming less efficient and more challenging to manage. This project addresses these issues by introducing an automated system that can detect and monitor vehicles in real time.

The system uses components such as Arduino UNO, IR sensors, and output devices like LED or buzzer to detect the presence of vehicles. It reduces the need for manual intervention and provides quick and accurate results. The use of sensors ensures that the system responds instantly whenever a vehicle passes, making it suitable for real-time applications.

One of the major advantages of this system is its simplicity and cost- effectiveness. The hardware components used are easily available and affordable, making the system suitable for small-scale as well as large-scale implementations. The use of a breadboard and jumper wires also makes it easy to assemble, modify, and test the circuit without complexity.

Furthermore, the system can be enhanced by integrating advanced technologies such as cameras, artificial intelligence, and wireless communication modules. These improvements can enable features like vehicle classification, traffic density analysis, and remote monitoring through cloud platforms. This makes the system scalable and adaptable for future smart city applications.

The project also highlights the importance of proper system design and integration of hardware and software components. A well-designed system ensures better performance, reliability, and accuracy. It also allows easy maintenance and future.

REFERENCES

1. Jindal, B., & Singh, A. (2024). Deep learning-based vehicle detection and classification using CNN. International Journal of Computer Vision and Applications.
2. Khemchandani, H. (2024). Intelligent traffic monitoring using machine learning algorithms. IEEE Transactions on Intelligent Transportation Systems.
3. Riyazussal, M. (2024). Dynamic vehicle classification based on motion analysis. Journal of Transportation Engineering.
4. Zafari, M. (2024). Edge computing in intelligent traffic systems. IEEE Access.
5. Redmon, J., & Farhadi, A. (2016). YOLO: Unified real-time object detection. CVPR.
6. Liu, W., et al. (2016). SSD: Single Shot MultiBox Detector. ECCV.
7. Sivakumar, R., & Trivedi, M. (2013). Vision-based vehicle detection systems. IEEE Transactions on ITS.
8. Dalal, N., & Triggs, B. (2005). Histogram of Oriented Gradients. CVPR.
9. Girshick, R. (2014). R-CNN object detection. CVPR.
10. He, K., et al. (2016). Deep residual learning. CVPR.
11. Krizhevsky, A., et al. (2012). ImageNet classification using CNN. NIPS.
12. OpenCV Contributors. (2023). OpenCV Library
13. Espressif Systems. (2022). ESP8266 Wi-Fi Module Datasheet.
14. Viola, P., & Jones, M. (2001). Rapid object detection using Haar features. CVPR.
15. Teichmann, M. (2018). Real-time semantic segmentation for autonomous vehicles.
16. Geiger, A., et al. (2013). KITTI Vision Benchmark Suite. CVPR.
17. World Health Organization. (2018). Global status report on road safety.
18. Ministry of Road Transport and Highways. (2022). Road Accident Report.
19. Jindal, B., & Singh, A. (2024). Deep learning-based vehicle detection and classification using CNN. International Journal of Computer Vision and Applications.
20. Khemchandani, H. (2024). Intelligent traffic monitoring using machine learning algorithms. IEEE Transactions on Intelligent Transportation Systems.
21. Riyazussal, M. (2024). Dynamic vehicle classification based on motion analysis. Journal of Transportation Engineering.
22. Zafari, M. (2024). Edge computing in intelligent traffic systems. IEEE Access.
23. Redmon, J., & Farhadi, A. (2016). YOLO: Unified real-time object detection. CVPR.
24. Liu, W., et al. (2016). SSD: Single Shot MultiBox Detector. ECCV.
25. Sivakumar, R., & Trivedi, M. (2013). Vision-based vehicle detection systems. IEEE Transactions on ITS.
26. Dalal, N., & Triggs, B. (2005). Histogram of Oriented Gradients. CVPR.
27. Girshick, R. (2014). R-CNN object detection. CVPR.
28. He, K., et al. (2016). Deep residual learning. CVPR.
29. Krizhevsky, A., et al. (2012). ImageNet classification using CNN. NIPS.
30. OpenCV Contributors. (2023). OpenCV Library
31. Espressif Systems. (2022). ESP8266 Wi-Fi Module Datasheet.
32. Viola, P., & Jones, M. (2001). Rapid object detection using Haar features. CVPR.

© 2026 The Author(s). Published by IJCOPE Journal. Website: https://ijcope.org/

33. Teichmann, M. (2018). Real-time semantic segmentation for autonomous vehicles.
34. Geiger, A., et al. (2013). KITTI Vision Benchmark Suite. CVPR.
35. World Health Organization. (2018). Global status report on road safety.
36. Ministry of Road Transport and Highways. (2022). Road Accident Report.