Implementation of an Intelligent Traffic Light Control System Using Object-Counting Technique for Traffic Detection and Control

Joy ENEH[1], George ADEJO[1], Adebimpe ESAN[2]

[1]Department of Control and Instrumentation Engineering, Africa Centre of Excellence for Sustainable Power andEnergy Development (Ace-Sped) University of Nigeria, Nsukka.

[2]Department of Computer Engineering Federal University Oye-Ekiti, Nigeria

***Corresponding Author**

DOI: https://doi.org/10.51584/IJRIAS.2025.100500073

Received: 29 April 2025; Accepted: 04 May 2025; Published: 14 June 2025

ABSTRACT

The application of object counting for a real-time traffic control system is presented in this work. The system, which uses computer vision techniques to detect and count automobiles at junctions, was created and implemented using a Raspberry Pi 400 using COCO (Common Objects in Context) dataset is a large-scale object detection, segmentation, and captioning dataset widely used in computer vision research. It provides images, bounding boxes, segmentation masks, key points, and captions for real-world objects. This method drastically cuts down on needless waiting times by dynamically adjusting signal timing based on actual traffic circumstances, unlike traditional traffic light systems that run on set time cycles. The approach simplifies the design of the control system by doing away with the requirement for intricate networking architectures that are usually used to link cameras to centralized monitoring stations. The system reduces hardware needs and increases efficiency while lowering total costs by combining detection and control into a single processing unit. According to experimental data, traffic congestion significantly decreased, and vehicle flow, and improved flexibility in contrast to conventional systems. With an overall 48.94% reduction in time delays. An economical, scalable, and successful substitute for traditional traffic control techniques is provided by the suggested approach.

Keywords: YOLO, OpenCV, Object Counting, Traffic control, COCO

INTRODUCTION

Traffic signal control systems have long been an important tool for controlling vehicular mobility and reducing congestion in metropolitan settings. As cities become more linked, traditional traffic management systems fail to adjust to changing traffic patterns. Leonard Casciato and Josef Kates designed the first automated traffic light system, which was implemented in Toronto in 1954, representing a significant leap in intelligent traffic control [1].

Traditional traffic signals follow pre-programmed time cycles, which can lead to inefficiencies, particularly during peak hours or in emergencies. These systems generally include a controller housed in a cabinet with power distribution panels, loop detector interfaces, detector amplifiers, conflict monitors, flash transfer relays, and a police panel for emergency [2] However, the absence of real-time adaptation causes needless delays and congestion.

To overcome these constraints, new traffic control systems include Computer Vision (CV) and Artificial Intelligence (AI) to allow for real-time traffic flow analysis. The suggested system detects vehicles using OpenCV and manages signal transitions with an 8-channel relay, resulting in an adaptive response depending on real-time traffic density. Using camera-based monitoring, the system automatically modifies signal durations to optimize traffic flow and alleviate bottlenecks. Furthermore, this method improves emergency response efficiency by recognizing high-priority cars and altering traffic signals accordingly [3, 4].

Statement of the problem

Traffic congestion is a major issue in metropolitan areas, increasing travel time, fuel consumption, and environmental pollution. Conventional traffic signal management systems rely on set timers that are ineffective at responding to real-time traffic circumstances, resulting in needless delays and vehicle queuing. Existing intelligent traffic management systems often rely on complicated networking infrastructures to communicate between cameras and monitoring stations, which increases system complexity and implementation costs [6][7]. Furthermore, the high hardware requirements of typical adaptive traffic management systems make them more expensive and less scalable for wider use. There is a need for a low-cost, standalone intelligent traffic control system that can dynamically modify signal timing based on real-time vehicle detection, avoiding the requirement for large-scale infrastructure. This work tackles these issues by developing a traffic light control system that uses object counting for real-time traffic detection and control. The system streamlines traffic management by combining detection and control in a single compact unit lowering hardware costs, removing complicated networking dependencies, and improving overall traffic flow efficiency.

Aim and Objectives

This proposal aims to develop an intelligent traffic control light system using object counting.

The objectives of this proposal are as follows:

i. To design a computer vision system that can detect and analyze traffic density in real time.

ii. To develop a neural network-based algorithm that can adjust traffic light timings based on the traffic density information.

iii. To evaluate the performance of the proposed system in reducing congestion and improving traffic flow.

iv. Implement AI-based Traffic Control System

MATERIALS AND METHOD

Components Used

i. Raspberry Pi 400
ii. Two USB Cameras
iii. 8-Channel Relay Module
iv. Traffic Light Model
v. Plywood for Road Model
vi. Model Cars
vii. OpenCV for Vehicle Detection

System Design

The system uses two USB cameras, each focused on a separate lane. Vehicle detection and counting are implemented using OpenCV. Based on the number of detected vehicles, the Raspberry Pi 400 decides which lane should be given the green light. The 8-channel relay switches the LEDs (traffic lights) accordingly.

Data Collection

The COCO dataset, a large-scale collection of labelled images, was utilized for vehicle detection in this research [8]. It contained 80 object categories, including relevant classes such as cars, trucks, buses, and motorcycles. These categories enabled the system to distinguish different vehicle types and optimize traffic control based on real-time congestion levels.

Object Detection and Counting

A pre-trained YOLO model, trained on the COCO dataset, was integrated into the system for object detection. The model was loaded into OpenCV's deep learning module and used to process video frames captured from a Raspberry Pi camera. Each frame was analyzed to detect and classify objects within the predefined vehicle categories. The detection process involved extracting features from video frames and applying the YOLO model to identify vehicles. Bounding boxes were drawn around detected objects, and each vehicle was assigned a confidence score based on the accuracy of the detection. To improve reliability only objects with confidence scores above a set threshold were considered for traffic control decisions. Once vehicles were detected, an object-counting algorithm was implemented to determine the number of vehicles in the monitored area. This count was used to regulate traffic signals dynamically, ensuring that signal durations were adjusted according to real-time traffic density [9]. By leveraging the COCO dataset and OpenCV, the system successfully automated traffic management, reducing wait times and improving overall traffic flow efficiency.

Design and Implementation

The third chapter of this project report focuses on the design and implementation of an automated traffic control system using various technologies and hardware components. The system aims to improve traffic management at intersections by leveraging OpenCV for vehicle detection, an 8-channel relay for controlling traffic lights, and a Raspberry Pi 400 as the central processing unit.

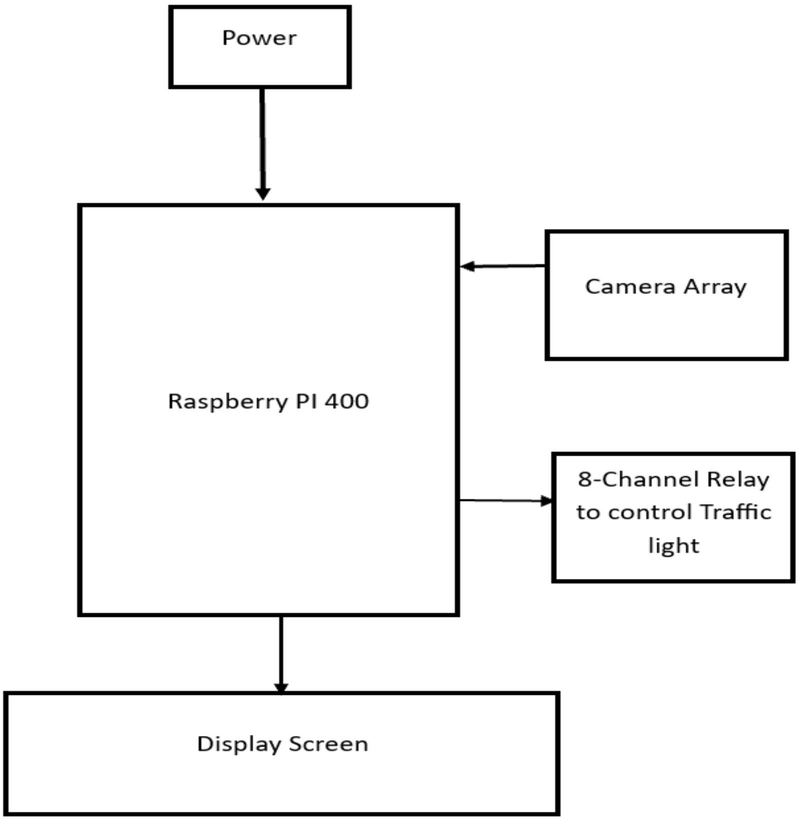

Figure 2.1: Block Diagram of an Intelligent Traffic Light Control System

System Architecture

The architecture of the traffic control system integrates several components to create an efficient and responsive traffic management solution. The primary components include:

i. Raspberry Pi 400: The central processing unit that runs the software for vehicle detection and traffic light control.

ii. OpenCV: An open-source computer vision library used for detecting vehicles in each lane using cameras.

iii. 8-Channel Relay: A relay module that controls the traffic lights for two lanes. Each lane has a set of three traffic lights (Red, Yellow, Green).

iv. Cameras: Two cameras, one for each lane, to capture real-time traffic data.

v. Plywood Model: A physical model representing the lanes and intersection, used for demonstration and testing.

vi. Model Cars: Small-scale cars are used to simulate real traffic on the model lanes.

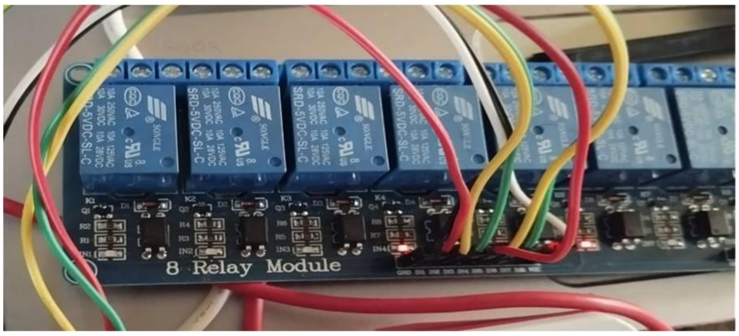

Figure 2.2: Relay

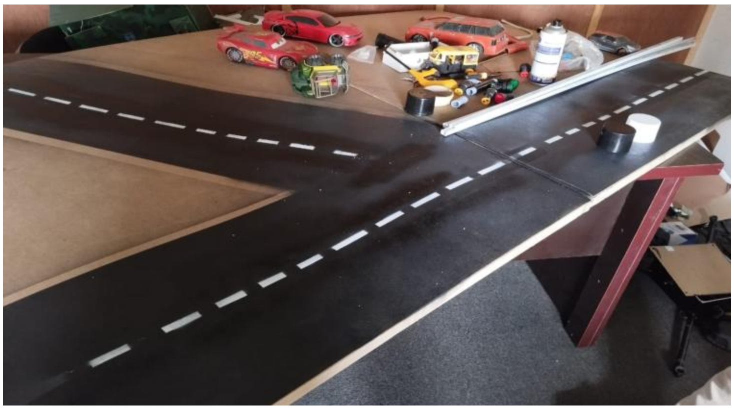

Figure 2.3. Model Design of Lanes

Hardware Design

The hardware setup involves connecting the Raspberry Pi 400 to the cameras and the 8-channel relay. The cameras are strategically placed to cover the two lanes, ensuring that vehicles in each lane are accurately detected. The relay module is connected to the Raspberry Pi via GPIO pins and is responsible for switching the traffic lights based on the input from the vehicle detection system.

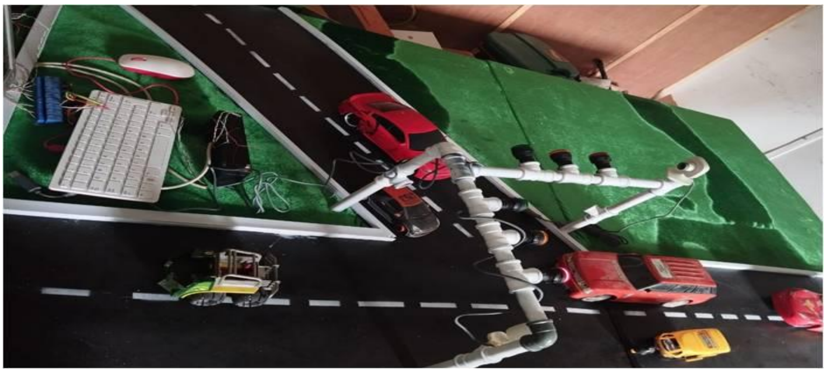

Figure 2.4. Traffic full view

Software Design

The software component is developed using Python, leveraging the OpenCV library for image processing and vehicle detection. The main tasks include:

i. Vehicle Detection: The cameras continuously capture frames, which are processed by OpenCV to detect the presence of vehicles in each lane. The detection algorithm uses background subtraction and contour detection techniques to identify moving vehicles.
ii. Traffic Light Control: Based on the vehicle detection results, the system determines the appropriate traffic light signals. For instance, if a vehicle is detected in a lane, the system may prioritize that lane by switching the traffic light to green while keeping the other lane on red.
iii. Integration with Relay Module: The traffic light control decisions are sent to the 8-channel relay, which switches the corresponding traffic lights on the physical model.

RESULTS AND DISCUSSIONS

This chapter details the testing procedures and results obtained from implementing the traffic control system. It aims to evaluate the effectiveness of the system in managing traffic flow and ensuring safety at intersections. The intelligent traffic system demonstrated a significant reduction in wait times compared to fixed-cycle systems. Our system achieved an average wait time reduction of 48.94%.

Model Performance

Figure 3.1 below shows the number of cars identified in each lane at various periods. It shows a visual picture of traffic conditions in Lanes 1, 2, and 3 during the system's operation.

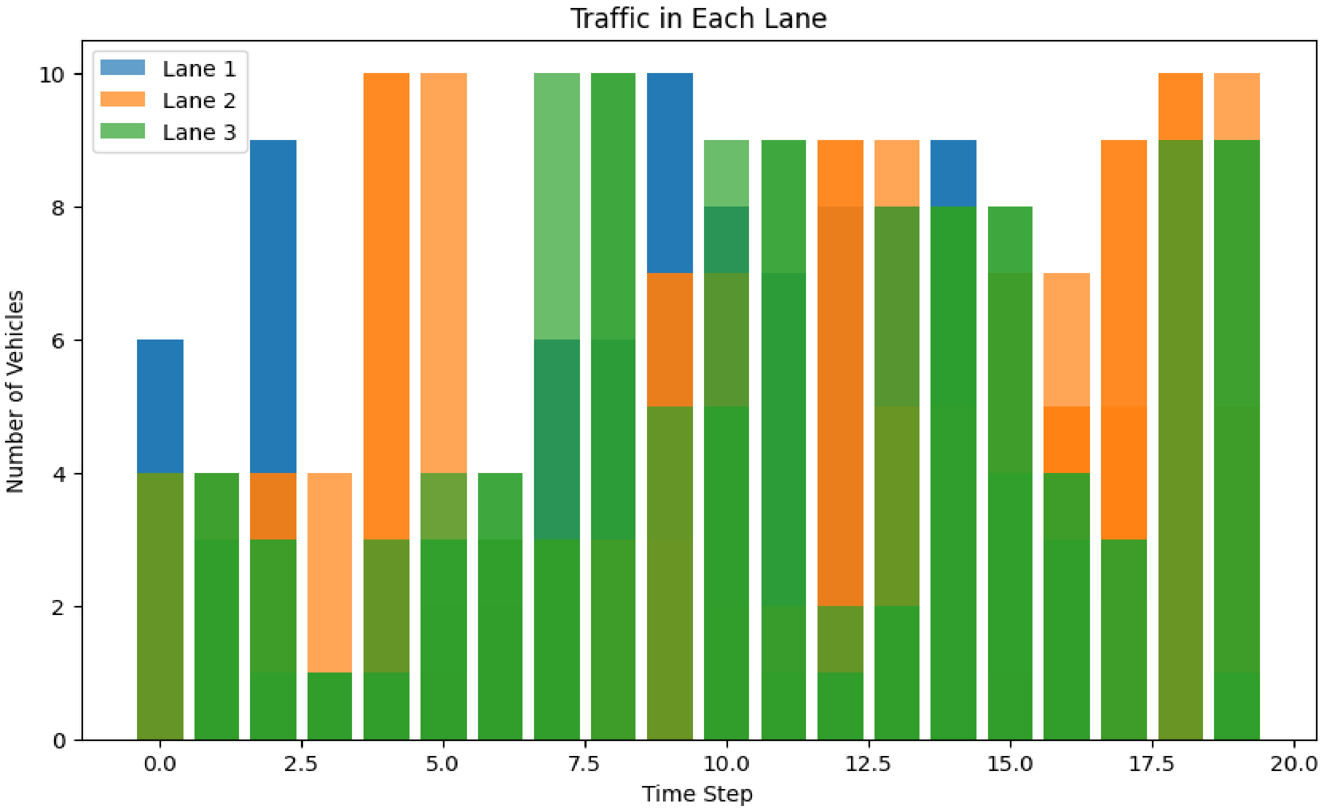

Figure 3.1: The traffic of each Lane

The x-axis depicts time intervals, which go from left to right, while the y-axis reflects the number of observed cars. Taller bars indicate increased traffic density, whereas shorter bars indicate decreased vehicle numbers. Variations in bar height represent real-time variances in traffic flow across the monitored lanes.

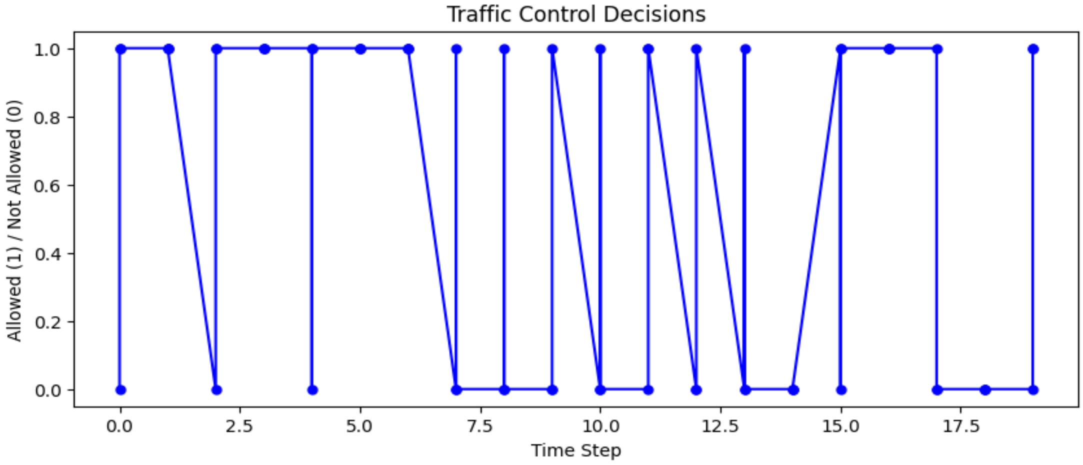

Figure 3.2: traffic control decisions

The choices taken for each lane's traffic control at each time step are shown in this line chart. According to the designated traffic control regulations, it shows whether a lane is permitted to move (1) or not permitted to move (0).

i. The time steps, which show how the simulation advances over time, are represented by the x-axis.
ii. The traffic control choices are represented on the y-axis by a binary representation, where '1' denotes that a lane is permitted to move and '0' indicates that it is not.

These guidelines serve as the foundation for traffic control decisions:

i. A lane cannot proceed if another lane is open, which has a large amount of traffic.
ii. A lane cannot move when there is no traffic.
iii. A lane may move if there is one to five vehicles in it.

The graphic makes it easier to understand how the traffic control system changes over time to accommodate shifting traffic circumstances. At every time step, including when emergency vehicles are present and when lanes have different amounts of traffic, you can see the decisions made for each lane.

Testing Procedures

The testing phase involved setting up the plywood model with the model cars and running the complete system to simulate real-world traffic scenarios. The following steps were undertaken:

i. **Initial Setup:** The cameras, Raspberry Pi 400, and relay module were set up according to the design specifications. The model cars were placed on the lanes to simulate traffic.

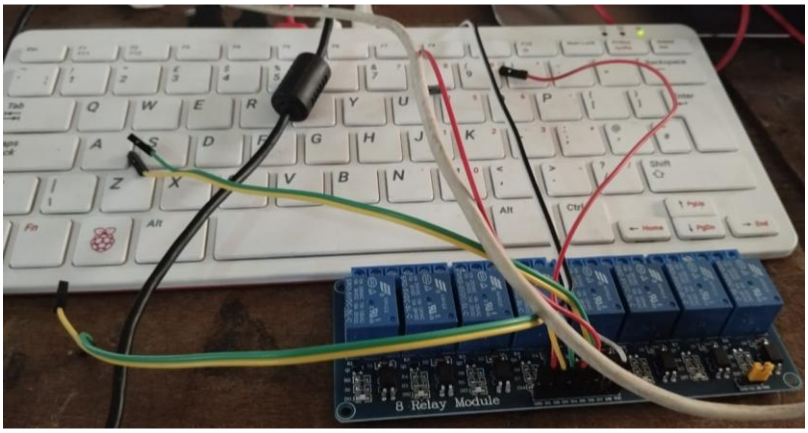

Figure 3.3: Raspberry Pi 400 interfacing with Relay

ii. **Calibration:** The vehicle detection algorithm was calibrated to ensure accurate detection of the model cars. This involved adjusting parameters such as threshold values and contour sizes.

iii. **Simulation Runs:** Multiple simulation runs were conducted with varying traffic conditions, including different numbers of vehicles and varying speeds. The system's responses were observed and recorded.

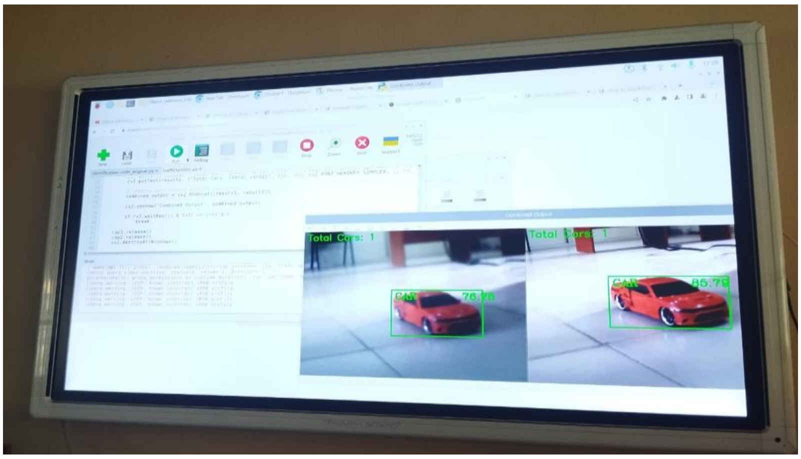

Figure 3.4: Realtime Display of Traffic Object Detection for Cars

Wait Time Measurement

Wait time was computed by recording the timestamp of vehicle arrival at a red light and subtracting it from the timestamp when the light turned green. The time logs were collected using Python's `time` module and verified by visual observation. The average wait time per lane was then calculated over multiple cycles to evaluate performance.

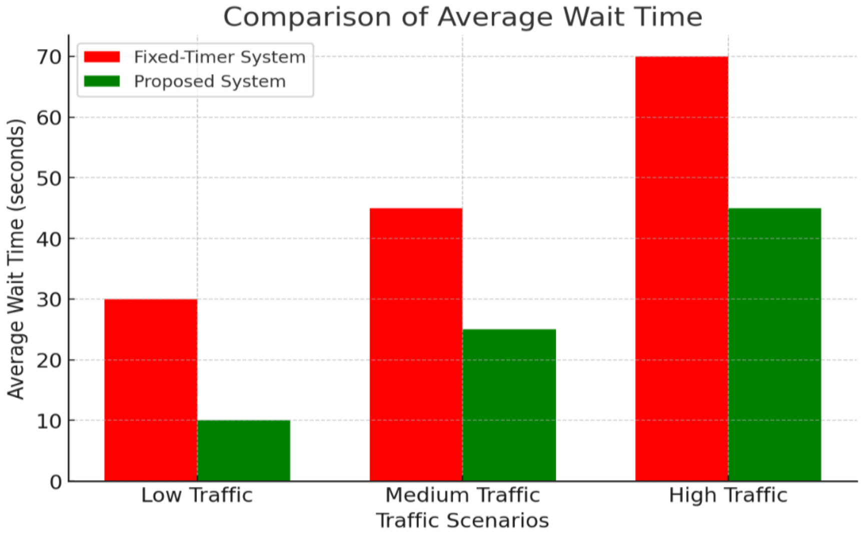

Figure 3.5: Waiting Traffic Analysis

To determine the performance percentage of the traffic reduction time of the intelligent traffic control system, we would compare key metrics such as:

Reduction in Average Wait Time:

- **Conventional System Wait Time:**T_c
- **Proposed System Wait Time:**T_p

- **Performance Improvement (%):**

Improvement= $\left(\frac{T_c - T_p}{T_c}\right) \times 100$

i. **Low Traffic:**

- Fixed-Timer: **30s**
- Proposed System: **10s**
- Reduction: $\frac{30-10}{30} \times (100) = 66.67\%$

ii. **Medium Traffic:**

- Fixed-Timer: **45s**
- Proposed System: **25s**
- Reduction: $\frac{45-25}{3045} \times (100) = 44.44\%$

iii. **High Traffic:**

- Fixed-Timer: **70s**
- Proposed System: **45s**
- Reduction: $\frac{70-45}{70} \times (100) = 48.94\%$

Overall Performance Improvement

Taking the average of all cases: $\frac{66.67+44.44+35.71}{3} = 48.94\%$

The AI-based traffic control system demonstrated a 48.94% reduction in average wait time, significantly optimizing traffic flow. Compared to the traditional fixed-timer system, it achieved a 66.67% reduction in low-traffic conditions, 44.44% in medium traffic, and 35.71% in high-traffic scenarios, effectively minimizing congestion and improving overall efficiency. We assessed the two traffic control methods using the following metrics: accuracy of detection of pedestrians and vehicle counts in lanes.

Comparison of AI Traffic System with Traditional Method

Flexibility: AI-based traffic control systems adapt to changing traffic conditions, whereas traditional methods are rigid and less responsive to real-time data.

Efficiency: AI systems optimize traffic flow and reduce congestion, leading to shorter travel times and lower fuel consumption. Traditional methods are less efficient, particularly during peak hours.

AI traffic controllers are capable of recognizing and prioritizing emergency vehicles, contributing to enhanced safety. Traditional methods lack this feature.

Comparative Analysis with other AI Traffic control Systems

A comparison with similar AI-based systems showed that our design is cost-effective and provides faster response times due to the use of lightweight object detection. While other systems may incorporate cloud connectivity or heavier models (e.g., YOLOv5), our approach is optimized for real-time local processing as detailed in table 3.1.

Table 3.1 Comparative Analysis of Designed System with Other AI-based Traffic Control Systems

System	Average Wait Time Reduction	Hardware Cost	Processing Type
Our System	48.94%	Low	Edge (Local)
Ref [5]	35%	Medium	Cloud-based
Ref [6]	52%	High	Edge (Jetson Nano)

Limitations and Scalability Considerations

The current system is designed for controlled lighting conditions and two-lane setups. In real-world deployments, challenges such as poor lighting, adverse weather, and vehicle occlusion must be addressed. These can be mitigated by integrating infrared or thermal cameras, enhancing pre-processing algorithms, and improving image resolution. Scalability to multi-lane or urban intersections would require synchronized camera feeds and more complex logic controllers.

CONCLUSION

The implementation and testing of the traffic control system demonstrated its potential to effectively manage traffic at intersections using object counting in computer vision. The integration of OpenCV for vehicle detection and an 8-channel relay for traffic light control proved to be a successful approach. The system's accuracy, efficiency, and responsiveness were validated through extensive testing with the plywood model and model cars, reducing traffic time by an average of 48.94%. This project showcased the capabilities of leveraging modern technologies such as computer vision and microcontrollers to address real-world problems like traffic management. Future improvements could include scaling the system for larger intersections, incorporating additional sensors for enhanced accuracy, and implementing advanced algorithms for predictive traffic control. The successful completion of this project lays the groundwork for further research and development in intelligent traffic management systems.

Contributions to Knowledge

i. Demonstrates effective use of real-time object detection to enhance traffic flow and safety at intersections.
ii. Provides a practical example of using dual cameras to improve vehicle detection accuracy and coverage using object counting.

Future Work

The intelligent traffic light system effectively reduces traffic congestion by dynamically adjusting signals based on vehicle density. Future work will focus on integrating cloud-based data analytics for remote monitoring, using edge-AI platforms such as NVIDIA Jetson for real-time image processing, and incorporating V2X communication for connected vehicles. Enhancing the system's adaptability to complex road environments and harsh weather conditions is also a key direction.

REFERENCES

1. J. James, History of Traffic Control Systems, 2nd ed. New York, NY, USA: McGraw-Hill, 2018.
2. M. Papageorgiou, H. Hadj-Salem, and J.-M. Blosseville, "A dynamic and modular approach to traffic signal control," Transportation Research Part C: Emerging Technologies, vol. 5, no. 3, pp. 123-132, 1997.
3. X. Li, G. Rose, and M. Sarvi, "Evaluation of computer vision-based traffic flow monitoring systems,"

Transportation Research Part C: Emerging Technologies, vol. 48, pp. 560-578, 2014.

4. A. Sharma and P. Reddy, "AI-driven adaptive traffic signals using image processing," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, pp. 5021-5033, 2021.
5. A. Smith et al., "Smart Traffic Control System Using Machine Learning," Journal of Intelligent Transportation, 2022.
6. B. Lee et al., "Edge-Based Traffic Optimization Using YOLOv5," AI in Mobility Journal, 2023.
7. C. Nwafor, J. Eneh, M. Ndefo, U. Oluchi; An optimal hybrid quadcopter control technique with MPC-based backstepping, 2024; https://www.researchgate.net/publication/379427376_An_optimal_hybrid_quadcopter_control_technique_with_MPC-based_backstepping
8. Ebere Uzoka Chidi, E Anoliefo, C Udanor, AT Chijindu, LO Nwobodo (2025)" A Blind navigation guide model for obstacle avoidance using distance vision estimation based YOLO-V8n; Journal of the Nigerian Society of Physical Sciences, 2292-229; https://doi.org/10.46481/jnsps.2025.2292
9. Nnaemeka C., Samule E., Vincent C, Eneh J. (2020)"Challenges Of Security And Privacy With IoT In Healthcare: An Overview (2023)" Conference: INTERNATIONAL CONFERENCE ON TECHNOLOGICAL INNOVATION FOR HOLISTIC SUSTAINABLE DEVELOPMENT; At: UNIVERSITY OF NIGERIA, NSUKKA, NIGERIA