AI - powered Density Based Traffic Control System

S Anupama[1], Neeraj Pradeesh[2], Harinandana P[3], Harinandana V A[4], Ms. Silja Varghese[5]

[1]Bachelor of Technology in CSE, NCERC
[2]Bachelor of Technology in CSE, NCERC
[3]Bachelor of Technology in CSE, NCERC
[4]Bachelor of Technology in CSE, NCERC
[5]Assistant Professor, Dept. of CSE, NCERC

---***---

Abstract - In urban areas, unmanaged traffic congestion leads to delays, fuel wastage, and increased pollution. This paper proposes an AI-based traffic control system that dynamically adjusts traffic signal durations based on real-time vehicle density. The system is implemented as a mini working model, and for this purpose, a YOLO object detection model is trained specifically on images of toy vehicles to detect and count them from a USB webcam feed. The processed data is used to determine traffic density, and the ESP32 microcontroller is interfaced to control signal lights accordingly. The system achieves a cost-effective and efficient solution for intelligent traffic management, especially in a scaled-down prototype model using toy vehicles.

Key Words: YOLO, ESP32, Dynamic Traffic Control, Object Detection, Smart City, Real-Time Detection

1. INTRODUCTION

Managing urban traffic efficiently remains a significant challenge due to the ever-increasing number of vehicles on the road. With the rise in private vehicle ownership, population growth, and urban expansion, existing traffic systems are often unable to cope with the dynamic and unpredictable nature of traffic flow. As a result, cities frequently experience bottlenecks, prolonged waiting times at signals, fuel wastage, and increased air and noise pollution. In many developing countries, the problem is further intensified due to lack of infrastructure upgrades and inadequate traffic enforcement.

Traditional traffic light systems generally rely on pre-set timers that do not consider actual traffic density, making them inefficient during off-peak hours or during congestion. These fixed-timing systems treat all lanes equally, regardless of the number of vehicles waiting, resulting in unfair green signal durations and increased driver frustration. These limitations necessitate the development of intelligent traffic management systems capable of adapting to real-time traffic conditions. Such systems are envisioned to dynamically alter traffic light durations based on current road usage, thus promoting smooth traffic flow, reducing idle time, and minimizing vehicle emissions.

Recent advancements in artificial intelligence (AI), computer vision, and microcontroller technology have enabled the design of smart systems that can monitor traffic, analyze vehicle flow, and adjust signal timings accordingly. One such advancement is the use of object detection algorithms like YOLO (You Only Look Once), which allow real-time recognition and counting of vehicles in video feeds. YOLO's speed and accuracy make it an ideal candidate for real-time applications such as traffic management.

In this project, we propose a low-cost, AI-powered traffic signal control system that has been developed and implemented as a mini working model. The system simulates real-life traffic scenarios using toy vehicles on a scaled-down road setup. A USB webcam captures overhead video of the model, and OpenCV is used for image handling and frame preprocessing. [2] The YOLOv8 model, trained to detect toy versions of cars, trucks, buses, and bikes, analyzes each frame to determine the number and type of vehicles. The training dataset was annotated using the labelImg tool to ensure precise bounding boxes and class labels.

Using a custom weighted vehicle count method, the system classifies traffic density levels and determines optimal green light durations for each lane. For example, heavier vehicles like buses and trucks are assigned higher weights compared to smaller vehicles like bikes. This decision is sent from a Python program running on a laptop to an ESP32 microcontroller via serial communication. The ESP32 then controls LEDs representing traffic signals, simulating a real-world adaptive signal control system. The programming for the microcontroller is done in C using the Arduino IDE, while the AI model and logic are implemented in Python using PyCharm.

The prototype demonstrates how AI and embedded systems can be leveraged to build smart city solutions, offering both educational value and practical insight into future traffic control implementations. Though implemented using a small-scale model, the approach and architecture are easily scalable and can be adapted to real-world traffic control infrastructures.

2. LITERATURE SURVEY

[2.1] Density Based Smart Traffic Control System Using Canny Edge Detection[1]

Authors: Harsha, P., Rahatekar, R., Rama Vaishnavi, T., Reddy, S., & Sreeya, S. (2024)

Focus: Proposed a dynamic traffic signal management system using image processing techniques. The system uses the Canny edge detection algorithm to estimate vehicle density from real-time camera feeds. Based on the detected density, green light durations are adjusted adaptively for each lane. Additionally, YOLO object detection is integrated to identify emergency vehicles and prioritize their passage. The study demonstrated that this approach significantly improves traffic flow efficiency compared to traditional fixed-timer systems.

[2.2] Adaptive Traffic Signal Timer[6]

Author: Mihir M. Gandhi

Focus: This project focuses on developing an AI-powered traffic signal system that adjusts green light durations dynamically based on real-time traffic density. Using the YOLO object detection algorithm, it detects and counts vehicles from live camera feeds to determine lane-wise density. The system then calculates optimal green signal timings to improve traffic flow, reduce idle time, and minimize fuel consumption. A simulation using Pygame is also included to visualize and evaluate the performance of the adaptive system under different traffic conditions. The project demonstrates how computer vision and automation can be effectively applied in intelligent traffic management.

[2.3] Dynamic Traffic Management and Prioritization for Emergency Vehicles Using IoT-enabled Density Control[3]

Authors: Palani Pavan Teja, S. Saravanan, Sura Deepika, Surepally Manasa, A. Vijaykumar, M. Prameela

Focus: This paper presents an IoT-based dynamic traffic management system that adjusts signal timings based on real-time traffic density. The system utilizes sensors and IoT technology to monitor traffic flow and prioritize emergency vehicles, ensuring timely passage through intersections. By dynamically controlling traffic signals, the system aims to reduce congestion, enhance emergency response times, and improve overall traffic efficiency.

3. PROPOSED SYSTEM

The proposed system is designed to address the challenges of traffic congestion by dynamically adjusting traffic signal timings based on real-time vehicle density. The system employs AI-based object detection for accurate vehicle counting and integrates this with an embedded microcontroller for traffic signal control.

The core components of the proposed system are as follows:

Real-Time Vehicle Detection: A USB webcam captures live footage of a miniature road setup. The captured footage is processed by a YOLOv8 model, which has been trained on toy vehicle images. The model detects and classifies vehicles (car, truck, bike, and bus) in real-time.

Traffic Density Calculation: The system uses the vehicle count to calculate the traffic density for each lane. Different vehicle types are assigned different weights (e.g., car = 2, bus = 3, truck = 3, bike = 1), allowing for an accurate calculation of the overall traffic density. Based on this density, the system classifies the traffic situation into categories such as Low, Medium, High, or Extreme.

Dynamic Signal Control: The Python script calculates the green signal duration for each lane based on the traffic density. A longer green signal duration is allocated to lanes with higher vehicle density to optimize traffic flow. The calculated green time is sent to the ESP32 microcontroller via USB serial communication.

Microcontroller and Traffic Light Operation: The ESP32 receives the green signal time and controls the traffic light modules, switching them between red, yellow, and green states based on the dynamic green time. The traffic light system simulates a real-world traffic junction with three lanes.

The system, implemented as a mini working model with toy vehicles, demonstrates the feasibility and effectiveness of a smart, AI-driven traffic control system. By adjusting signal timings in real-time based on actual traffic conditions, this system aims to reduce wait times, improve traffic flow, and present a cost-effective solution to traffic management.

4. MODULE DESCRIPTION

The system is divided into the following five major modules:

Traffic Monitoring Module: This module captures real-time traffic footage using a USB webcam mounted above the miniature road setup. It serves as the input source for vehicle detection.

Image Processing Module: The captured video frames are processed using OpenCV, which handles real-time image acquisition and preprocessing tasks. These processed frames are then passed to a YOLOv8 object detection model that has been specifically trained to recognize toy vehicles such as cars, buses, trucks, and bikes. The module detects the number and type of vehicles in each lane.

Decision Making Module: Based on the detected vehicles, this module calculates traffic density using a weighted formula where different vehicle types contribute differently to the total density. It then determines the appropriate green light duration for each lane.

Communication Module: This module handles the serial communication between the laptop (which processes the image data) and the ESP32 microcontroller. It transmits the calculated green signal duration to the hardware controller.

Traffic Control Module: The ESP32 receives the timing data and controls the LEDs (representing traffic lights) accordingly. Each lane's signal is managed to allow optimized traffic flow based on density.

5. TECHNOLOGIES USED

The system leverages a combination of software and hardware components to achieve efficient traffic control:

Software

Coding platforms:

PyCharm: The integrated development environment (IDE) used for Python development, including YOLO model training and control logic implementation.

Arduino IDE: The IDE used for programming the ESP32 microcontroller, handling the communication and control of the traffic light modules.

Programming Languages:

Python: The programming language used to develop the control logic, including YOLO model inference, traffic density calculation, and serial communication with the ESP32.

C/C++: The programming language used for ESP32 microcontroller programming to control the traffic light signals.

Libraries and Frameworks:

YOLOv8 (You Only Look Once): A state-of-the-art object detection model used for real-time detection and classification of toy vehicles. YOLOv8 is fast and accurate, making it ideal for traffic monitoring applications.

OpenCV: A computer vision library used for image processing. It handles video feed input, image preprocessing, and visualization, feeding the processed images to the YOLOv8 model for detection.

labelImg: A graphical image annotation tool used to create and annotate a custom dataset of toy vehicle images for training the YOLOv8 model. It simplifies the process of labeling images for object detection tasks.

NumPy: A Python library for numerical operations, used in the traffic density calculation and processing.

PySerial: A Python library used for serial communication between the laptop and the ESP32 for controlling traffic signals.

Hardware

ESP32 microcontroller: A versatile microcontroller used to control the traffic signal lights. It receives the green signal duration from the Python script and adjusts the traffic lights (red, yellow, and green) for three lanes based on the calculated vehicle density (Fig.5.1).

USB Webcam: Captures real-time footage of the miniature road setup, providing input to the object detection system for vehicle counting and density calculation.

Traffic Light Modules: Controlled by the ESP32, these modules simulate the operation of traffic lights for three lanes of the miniature setup (Fig:5.2).

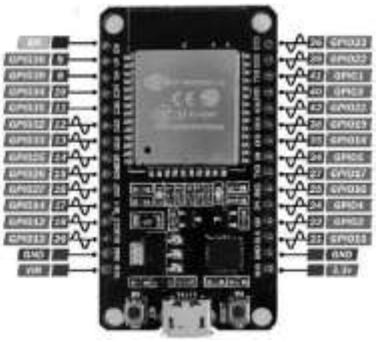

Fig-5.1: ESP32 Pin Diagram

Fig-5.2: Traffic Light Module

6. SYSTEM DESIGN

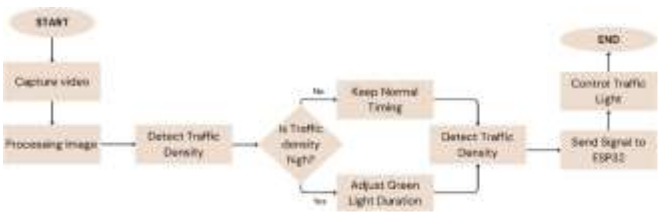

Fig-6.1: Flowchart

7. RESULTS

(i) YOLO training and testing results

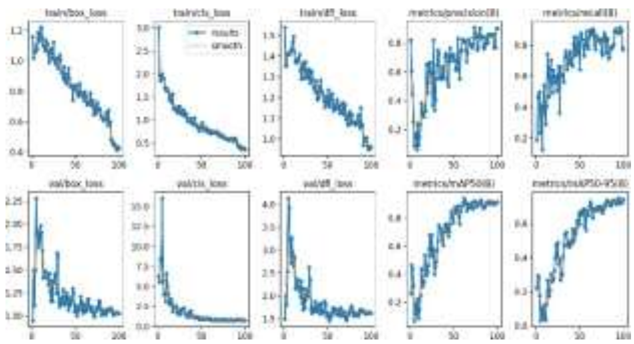

Fig-7.1

Fig-7.2

Fig-7.3

Fig-7.4

(ii) Implementation Results

Fig-7.5: Trained YOLO model detecting vehicles

Fig-7.6: Initializing

Fig-7.7: Dynamic allocation of green signal duration

Fig-7.8: Traffic lights functioning based on dynamic allocation of time

8. DISCUSSION AND ANALYSIS

8.1. ADVANTAGES

Smarter Traffic Flow Management: Dynamically adapts signal timing based on real-time traffic conditions, ensuring smoother traffic flow.

Reduced Waiting Time: Minimizes unnecessary delays at signals by prioritizing lanes with higher traffic density.

Reduced Fuel Consumption & Emissions: Shorter idle times lead to lower fuel use and reduced vehicular emissions, supporting sustainability.

Reduced Need for Manual Intervention: Fully automated decision-making removes dependency on human operators or fixed schedules.

Supports Smart City Initiatives: Aligns with modern urban planning strategies aimed at creating intelligent and efficient city infrastructure.

8.2. FUTURE WORKS

Smart City Integration: Expand the system's capability to integrate with city-wide traffic management and smart infrastructure.

Traffic Violation Detection: Incorporate AI models to identify red light violations or illegal turns.

Emergency Vehicle Prioritization: Add recognition features for ambulances, fire trucks, etc., to provide them priority signals.

Mobile App Integration: Enable users and authorities to monitor and control the system remotely through a dedicated mobile application.

Smart Pedestrian Crosswalk: Extend the system to detect pedestrian presence and provide dedicated crossing time based on foot traffic density.

9. CONCLUSION

In this project, we successfully developed an AI-based density traffic control system using a combination of computer vision, embedded systems, and machine learning. The mini working model demonstrates how smart traffic signals can be implemented using a YOLOv8 object detection model trained specifically on toy vehicles, effectively simulating real-life traffic conditions. By integrating OpenCV for image processing and ESP32 for traffic signal control, the system dynamically adapts signal timing based on traffic density, significantly improving traffic flow efficiency even in a scaled-down environment.

The modular and cost-effective design of our system highlights its potential for educational use and real-world adaptation, particularly in smart city initiatives. This project serves as a foundation for further advancements in intelligent traffic management, with possibilities for expanding features like emergency vehicle prioritization and city-wide IoT integration. Overall, the proposed solution exemplifies how AI and automation can transform traditional infrastructure into responsive, data-driven systems that improve urban mobility and reduce environmental impact.

In conclusion, the system sets a foundation for intelligent traffic control solutions and demonstrates the power of merging AI with embedded systems. With the increasing push towards smart cities and sustainable infrastructure, such innovations are likely to play a crucial role in shaping the urban transport systems of tomorrow.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our Mentor Ms. Silja Varghese for her invaluable guidance, encouragement, and continuous support throughout the development of this project. Her expertise and insightful feedback played a crucial role in enhancing our understanding and improving the project's quality. We are also thankful to Nehru College of Engineering and Research Centre, Thiruvilwamala, for providing the necessary resources and a conducive learning environment, enabling us to explore and apply our technical knowledge effectively.

Finally, we deeply appreciate the unwavering support and encouragement from our family and friends, whose motivation kept us inspired throughout this journey. This project has been a significant learning experience, and we are truly grateful to everyone who contributed to its successful completion.

REFERENCES

[1] "Density Based Smart Traffic Control System Using Canny Edge Detection"- Harsha, P Rahatekar, R Rama Vaishnavi, T Reddy, S Sreeya Turkish Journal of Computer and Mathematics Education;Gurgaon Vol. 15, Iss 3,3,(2024): 247 261.

DOI:10.61841/turcomat.v15i3.14798

[2] International Conference on Green Energy, Computing and Intelligent Technology 2024 (GEn-CITy 2024) "Traffic control system modelling to control traffic lights using smart camera" - Silvester D. H. Permana, Grandy Damarnegara, Ketut B. Y. Bintor

DOI: https://doi.org/10.1049/icp.2025.0256

[3] "Dynamic Traffic Management and Prioritization for Emergency Vehicles using IoT-enabled Density Control"- Palani Pavan Teja; S. Saravanan; Sura Deepika; Surepally Manasa; A Vijaykumar; M. Prameela.

DOI: 10.1109/IConSCEPT61884.2024.10627923

[4] "Self-Adaptive Traffic Density Based Management System" - Prutha Pawade; Vrinda Parkhi; Aditya Kadlag; Sakshi Patil; Manish Zure; Vivek Nikam.

DOI: 10.1109/ICBDS61829.2024.10837297

[5] "Fusing crowdsourced data to an adaptive wireless traffic signal control system architecture" - Amit Agarwal, Deorishabh Sahu, Anuj Nautiyal, Manan Gupta, Pratham Agarwal.

DOI: https://doi.org/10.1016/j.iot.2024.101169

[6] "Adaptive Traffic Signal Timer"

GitHub: https://github.com/mihir-m-gandhi/Adaptive-Traffic-Signal-Timer.git.

YouTube:https://youtu.be/OssY5pzOyo0?si=0kFj5Ns3ffTPadK0