International Journal of Research Publication and Reviews, Vol 6, Issue 6, pp 4803-4809 June 2025

WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Traffic Light Negotiation Using Image Processing

Thallapelli bharath[1], Dr. K. Santhi Sree[2]

[1](*Post Graduate Student, M. Tech (SE)* Department of Information Technology, Jawaharlal Nehru Technological University Hyderabad, Email: thallapellibharath310@gmail.com)

[2](Professor, Department of Information Technology, Jawaharlal Nehru Technological University Hyderabad, Email: drksanthisree@gmail.com)

DOI : https://doi.org/10.5281/zenodo.15631376

ABSTRACT

In urban environments, increasing traffic congestion demands smarter and more responsive control systems. This project proposes a traffic signal management system that dynamically adjusts signal timings based on real-time vehicle density using advanced image processing and object detection. We employed the YOLOv8n model, known for its speed and accuracy, to detect and count vehicles from live video feeds captured via IP webcams. By analyzing lane-wise traffic density through edge detection and bounding box regression, the system intelligently assigns green light durations to the busiest lanes, significantly reducing overall wait times and improving traffic flow. Our approach offers faster response compared to traditional fixed-timer systems and demonstrates high accuracy even under variable traffic conditions. Experimental simulations using Pygame visualize signal switching based on real-time vehicle counts, validating the system's practical effectiveness in smart city scenarios.

Keywords — YOLOv8, Image Processing, Traffic Light Automation, Real-Time Vehicle Detection, Smart Traffic Management.

I. INTRODUCTION

In many rapidly growing cities, including Hyderabad, traffic congestion has become a major challenge. Every day, commuters face long wait times at intersections, especially during rush hours when the volume of vehicles fluctuates unpredictably. Traditional traffic management systems, which operate on fixed-timer traffic lights, are unable to adapt to these dynamic conditions. Fixed timers allot the same duration to each direction, regardless of whether the road is crowded or empty, leading to unnecessary delays, increased fuel consumption, and greater environmental pollution.

Motivated by these inefficiencies, this project introduces a dynamic, AI-based traffic signal system that adjusts signal timings in real-time based on actual vehicle density. By using live camera feeds and applying advanced image processing techniques, we detect and count vehicles at each junction. A lightweight, fast object detection model — YOLOv8n — is utilized to ensure quick and accurate traffic analysis. Based on the number of vehicles detected in each lane, the system intelligently allocates green light durations, giving priority to the busiest roads while reducing waiting times for less congested ones. This approach not only improves traffic flow but also contributes to lower emissions and better urban mobility.

Through this project, we aim to move beyond rigid, outdated systems toward smarter and more sustainable traffic management solutions that can adapt in real time to the ever-changing demands of modern cities.

II. LITERATURE REVIEW

[1] Traffic congestion in urban areas often results from fixed traffic light systems unable to adapt to real-time conditions. To address this, a dynamic traffic light negotiation system using image processing and object detection technologies is proposed. Cameras at intersections capture live traffic data, which is analysed using advanced algorithms like YOLO, SSD, and Faster R-CNN to detect vehicles, estimate lane density, and optimize signal timing. YOLO’s speed and SSD’s balance between accuracy and efficiency make them ideal for real-time applications, while Faster R-CNN offers high precision but is computationally intensive. By leveraging these deep learning models and datasets like Microsoft COCO, the system aims to enhance traffic flow, reduce delays, and contribute to smarter urban traffic management solutions.

[2] Traffic congestion has become a pressing issue in urban areas due to the rapid increase in vehicle numbers and insufficient infrastructure. Traditional traffic control systems, which rely on fixed-timer signals or manual interventions, often fail to adapt to dynamic traffic conditions. As a result, these systems lead to delays, increased pollution, and higher fuel consumption. To address these challenges, smart traffic light systems based on image processing have emerged as an innovative and efficient solution. These systems utilize cameras to capture real-time traffic data and analyse it using advanced algorithms like Fast R-CNN, fuzzy logic controllers, and density-based methods. By determining vehicle density in different lanes, the

system dynamically allocates green light durations to ensure smoother traffic flow. For instance, lanes with higher traffic density receive longer green light times, while empty or less congested lanes have reduced waiting periods. This not only minimizes idle times but also prevents unnecessary delays.

[3] This paper presents an innovative solution to urban traffic congestion using real-time image processing to dynamically manage traffic signals. A camera captures live traffic footage, which is analysed in MATLAB to estimate vehicle density; this data determines the duration of green signals for each lane, prioritizing heavily congested roads. The system eliminates the need for traditional timers or electronic sensors, making it cost-effective and adaptable. It achieved high accuracy in detecting vehicles under optimal conditions, though challenges like low camera resolution and misinterpretations need addressing through improved hardware and dynamic thresholds. By minimizing human intervention and offering scalability for intersections with more than four lanes, the system enhances traffic flow, reduces pollution, and ensures safety, proving to be a practical and efficient solution for modern traffic management.

[4] The paper, "Traffic Management System using Machine Learning Algorithm," introduces a machine learning-driven solution to tackle urban traffic congestion. Using the YOLO (You Only Look Once) algorithm, the system performs real-time object detection, vehicle tracking, and counting from video inputs to assess traffic density and flow. By leveraging convolutional neural networks (CNN), it processes video frames to identify vehicles, predict congestion levels, and recommend alternative routes.

This approach aims to reduce traffic delays, prevent bottlenecks, and enhance road safety. A key feature of the system is its ability to handle wrong-way vehicle detection by analysing the vehicle's movement direction relative to the camera. When congestion is detected—based on a predefined vehicle count threshold—the system can suggest detours to alleviate traffic. The YOLO algorithm's high-speed processing and accuracy enable quick decision-making, making it ideal for real-world applications.

The paper emphasizes the benefits of this automated and proactive traffic management system, including minimized manual intervention and improved efficiency. It highlights the system's scalability for future integration with smart city technologies, where roads equipped with sensors can further enhance data collection and traffic prediction. Overall, the proposed system provides a practical, accurate, and efficient method for modern traffic management, contributing to sustainable and connected urban environments.

[5] The paper "Smart Traffic-Management using Machine Learning and IoT" discusses an advanced traffic management system integrating Machine Learning (ML) and the Internet of Things (IoT) to optimize traffic flow, enhance safety, and reduce congestion. IoT devices like sensors and cameras collect real-time data on traffic density, vehicle movement, and environmental conditions, while ML algorithms analyse this data to predict and manage traffic patterns dynamically. Techniques like object detection and clustering methods (e.g., DBSCAN) are employed to identify traffic violations, detect anomalies, and adapt signal timing in response to changing traffic conditions. This fusion enables proactive decision making, improves commute times, and minimizes traffic snarls. By automating responses and leveraging real-time insights, the system ensures a responsive, efficient, and sustainable approach to urban traffic management, supporting the development of smarter cities. The integration demonstrates significant potential for scalable, data-driven transportation solutions that can evolve with urban mobility challenges.

III. EXISTING SYSTEM

In the past, traffic control relied on manual management by traffic police stationed at intersections. These officers would observe the flow of vehicles and make real-time decisions to direct traffic accordingly. However, this approach had several limitations. Human decision-making was prone to delays, inconsistencies, and fatigue, making it inefficient. Additionally, a single officer could only effectively manage one junction at a time, which led to congestion in busy areas. Safety was also a major concern, as standing in the middle of a busy road put officers at risk of accidents. To overcome these challenges, manual traffic control was gradually replaced by automated traffic signals operating on fixed time cycles. The fixed time allocation technique currently in use follows a predetermined timing system for traffic lights. Each lane is given a set duration, such as 30 seconds, during which vehicles can pass before the signal changes. This system operates uniformly, assigning equal priority to all lanes regardless of actual traffic conditions. While this method is more structured than manual control, it does not account for real-time fluctuations in vehicle density. As a result, some lanes may remain green even when empty, while others, burdened with heavy traffic, receive the same duration, leading to inefficiencies.

This rigid approach creates several issues. Low-traffic lanes waste valuable green-light time, while high-traffic lanes suffer from congestion due to insufficient signal durations. Vehicles in other lanes are forced to wait unnecessarily, increasing travel times and driver frustration. Since fixed timings do not adapt to varying traffic patterns throughout the day, the system is particularly ineffective during rush hours. Additionally, prolonged vehicle idling at red lights leads to excessive fuel consumption and increased emissions, contributing to environmental pollution. These drawbacks highlight the need for a more dynamic and responsive traffic management system.

IV. PROPOSED SYSTEM

To address the limitations of static traffic signal systems, we propose a real-time, adaptive traffic control solution powered by computer vision and deep learning. This system dynamically adjusts signal timings based on the actual number of vehicles in each lane, which helps reduce unnecessary delays and manage congestion more effectively.

At the core of the system is an IP webcam placed at traffic intersections. The camera captures continuous video streams, which are broken down into individual frames for analysis. These frames are first processed using OpenCV to enhance visual clarity—for example, by reducing noise or improving contrast. Once preprocessed, each frame is fed into the YOLOv8n object detection model.

YOLOv8n, known for its lightweight design and fast inference speed, identifies vehicles in the scene by drawing bounding boxes around them and assigning each box a confidence score. We configured the model to focus on commonly seen vehicle types, including cars, motorcycles, and buses. After detecting the vehicles, the system counts how many are present in each lane to estimate traffic density.

Based on this count, green light durations are dynamically assigned. Lanes with more vehicles are given longer green cycles, while less congested lanes receive shorter ones. This real-time adjustment ensures efficient traffic flow, particularly during peak hours when conditions change rapidly.

To test the logic of our system, we built a simulation environment using Pygame. The simulation mimics real-world traffic behavior and visualizes how signal timings change based on incoming traffic data. It allowed us to evaluate the performance of our approach before any real-world deployment.

Challenges such as low-resolution camera feeds and overlapping vehicles were addressed through targeted preprocessing and careful tuning of YOLOv8n's detection thresholds. Overall, the system is designed to be lightweight, low-cost, and easy to integrate into existing infrastructure, making it a strong candidate for smart city traffic management.

ARCHITECTURE

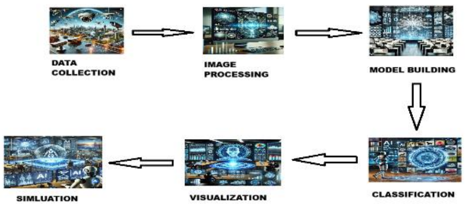


Figure No 1: Architecture

V. MODEL

In our project, we chose the YOLOv8n model—part of the YOLO (You Only Look Once) family—for real-time vehicle detection. The "n" version (nano) is optimized for speed and efficiency, making it particularly well-suited for systems where quick decision-making is essential, such as live traffic control.

YOLOv8n analyzes each video frame in a single pass, which means it can detect all objects in one shot rather than scanning the image multiple times. This single-shot detection approach greatly reduces processing time, allowing us to handle video input in real time. The model outputs bounding boxes for each detected vehicle along with class labels and confidence scores.

We trained and tuned the model to accurately detect cars, bikes, and buses, and adjusted detection thresholds to handle challenges like shadows, poor lighting, and vehicle overlap. One of the key advantages of YOLOv8n is its updated backbone network, which improves feature extraction, especially for small and partially visible vehicles.

By combining YOLOv8n with OpenCV preprocessing and real-time signal logic, we created a system that balances performance with practical deployment needs. This setup makes it possible to implement intelligent traffic control without requiring expensive hardware or complex installations.

Key Features:

Enhanced Backbone:

➢Utilizes a modernized backbone network for better feature extraction.

➢ Improves accuracy on diverse datasets.

Improved Head:

➢ Advanced head architecture for precise bounding box predictions.

➢ Enhances detection of small and overlapping objects.

Flexibility:

➢ Supports object detection, instance segmentation, and image classification.

YOLO ALGORITHM

The YOLO (You Only Look Once) algorithm is designed to detect objects quickly and accurately in a single glance, which makes it highly effective for real-time applications like traffic monitoring. Unlike traditional methods that scan the image multiple times, YOLO processes the entire image in one go, making it extremely fast.

YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

RESIDUAL BLOCKS

At the heart of YOLO's architecture are residual blocks, which help the neural network learn better by allowing information to skip through layers. This means the model can go deeper without losing important features, which improves the ability to detect objects more accurately—even in complex or cluttered scenes like busy intersections.
First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.

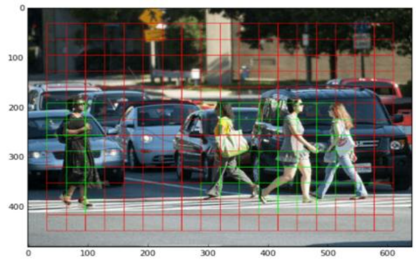


Figure No 2: Grid View

In the image above, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

BOUNDING BOX REGRESSION

Once objects are identified in the image, YOLO draws a rectangular box—called a bounding box—around each one. These boxes are defined by a few key attributes: their width, height, center coordinates (x, y), and the object's class (like car, bus, or bike). The algorithm doesn't just guess these values; it learns to predict them using a process called bounding box regression, which refines the box size and position to best fit the object.

➢ A bounding box is an outline that highlights an object in an image. Every bounding box in the image consists of the following attributes:
➢ Width (bw)
➢ Height (bh)
➢ Class (for example, person, car, traffic light, etc.)- This is represented by the letter
➢ Bounding box center (bx,by)

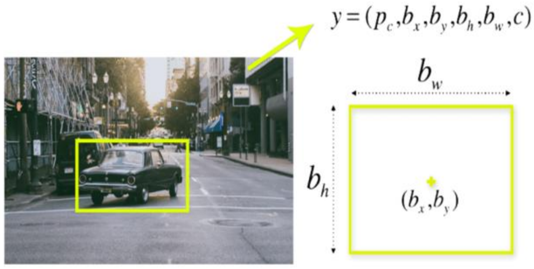


Figure No 3: Bounding box

INTERSECTION OVER UNION (IOU)

To evaluate how good these bounding boxes are, YOLO uses a metric called Intersection over Union (IoU). Think of it as comparing the predicted box with the actual, ground-truth box: IoU calculates how much the two overlap. A perfect match gives a score of 1.0, and the lower the overlap, the lower the score. This helps the model focus on the most accurate predictions and ignore false positives.
Together, these techniques make YOLO highly efficient for tasks like vehicle detection in live traffic video feeds, where quick and reliable decisions are crucial.
IoU is calculated as the ratio of the area of overlap between the predicted bounding box and the ground truth bounding box to the area of their union:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Figure No 4: IOU

VI. RESULTS

- Monitoring traffic using IP webcam for detecting the vehicles.

Figure No 5: IP WEBCAM

- Vehicle Detection and Counting.

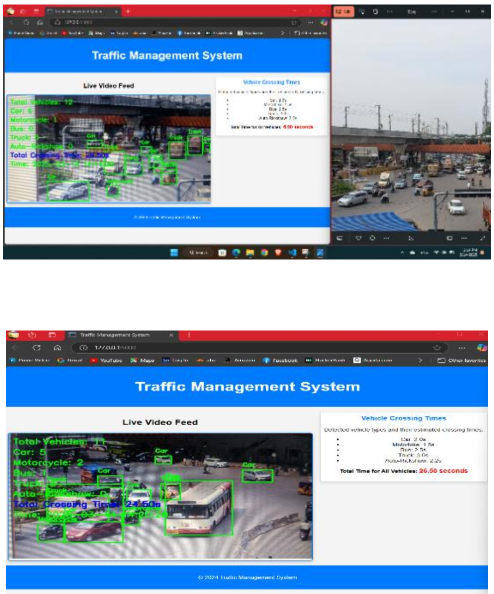


Figure No 6: Vehicle Detections

- Simulation of Traffic signals.

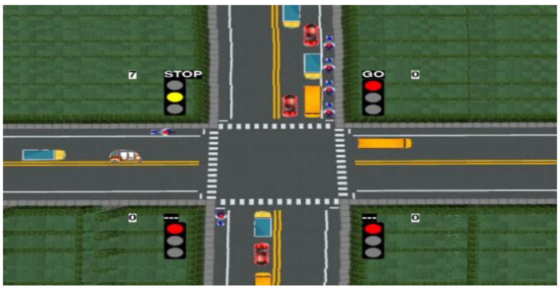
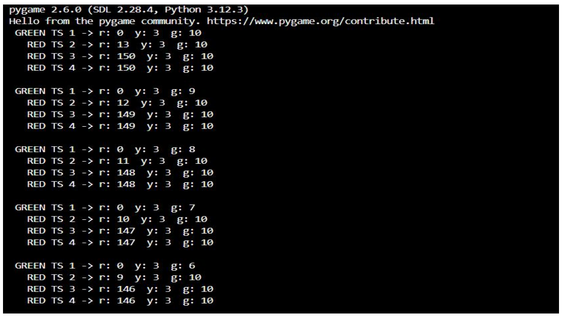


Figure No 7: simulation showing yellow light

Figure No 8: simulation showing green light

Figure No 9: simulation showing traffic



```
pygame 2.6.0 (SDL 2.28.4, Python 3.12.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
 GREEN TS 1 -> r: 0  y: 3  g: 10
   RED TS 2 -> r: 13  y: 3  g: 10
   RED TS 3 -> r: 150  y: 3  g: 10
   RED TS 4 -> r: 150  y: 3  g: 10

 GREEN TS 1 -> r: 0  y: 3  g: 9
   RED TS 2 -> r: 12  y: 3  g: 10
   RED TS 3 -> r: 149  y: 3  g: 10
   RED TS 4 -> r: 149  y: 3  g: 10

 GREEN TS 1 -> r: 0  y: 3  g: 8
   RED TS 2 -> r: 11  y: 3  g: 10
   RED TS 3 -> r: 148  y: 3  g: 10
   RED TS 4 -> r: 148  y: 3  g: 10

 GREEN TS 1 -> r: 0  y: 3  g: 7
   RED TS 2 -> r: 10  y: 3  g: 10
   RED TS 3 -> r: 147  y: 3  g: 10
   RED TS 4 -> r: 147  y: 3  g: 10

 GREEN TS 1 -> r: 0  y: 3  g: 6
   RED TS 2 -> r: 9  y: 3  g: 10
   RED TS 3 -> r: 146  y: 3  g: 10
   RED TS 4 -> r: 146  y: 3  g: 10
```

Figure No 10: showing times for signals

VII. CONCLUSION

In this project, we successfully developed a real-time adaptive traffic light control system using YOLOv8n for vehicle detection and dynamic signal timing. The system was able to detect vehicles accurately from live IP webcam feeds and allocate green light durations based on lane-wise traffic density. Through simulation testing with Pygame, we demonstrated that dynamic signal control significantly reduced unnecessary wait times compared to fixed-timer systems, leading to smoother traffic flow and more efficient lane clearing. One of the key achievements was optimizing YOLOv8n for low-resolution webcam feeds by applying preprocessing techniques, which maintain detection accuracy even under varying lighting conditions. The integration of fast object detection and real-time simulation proved that low-cost, easily deployable traffic management solutions are feasible without requiring expensive hardware or complex installations. For future work, we plan to extend the system by integrating emergency vehicle detection to prioritize ambulances and fire trucks at intersections. Additionally, we aim to deploy the system on a real-world small-scale testbed with multiple cameras to validate its performance in different traffic and environmental conditions. Incorporating IoT sensors to gather more detailed traffic data and using predictive analytics to anticipate congestion before it builds up are also considered for enhancing the system's intelligence. This project demonstrated the practical application of deep learning and computer vision in improving urban traffic management and sets a foundation for future enhancements toward fully autonomous, adaptive traffic control systems.

VIII. REFERENCES

[1]. S.K. Katiyar and P.V. Arun, Comparative detection techniques in context of object using Machine Learning.

[2]. Mohammed Siddique, Smart Traffic control System on Dhaka's average traffic speed Case using MATLAB.

[3]. Ankush, G Sai Pavan, Harihran M, Jagadish Naik, Kiran S M, Smart Traffic Control System Using Image Processing.

[4]. Dr.A. Ravi, R. Nandhini, K. Bhuvaneshwari, J.Divya, K.Janani .Professor, Department of EEE, A.V.C College of Engineering, Traffic Management System using Machine Learning Algorithm.

[5]. Beela Sushma Gayathri B. Tech Student, Department of IT, GMR Institute of Technology, Rajam-532127, Andhra Pradesh, India, Smart Traffic-Management using Machine Learning and IoT.

[6]. Aditi Yadav, Vaishali More, Neha Shinde, Manjiri Nerurkar, Nitin Sakhare, Department of Computer Engineering, Vishwakarma Institute of Information and Technology, Pune, Adaptive Traffic Management System Using IoT and Machine Learning

[7]. Sayanti Rahishet, Aparajita Sahoo, Aparna Indore, Vaibhav Deshmukh, Pushpa U.S, Department of Electronics and Telecommunications Engineering, Fr. C Rodrigues Institute of Technology, Vashi "Intelligent Traffic Light Control Using Image Processing".

[8]. Banerjee and M. K. Muju, "Implementation of image processing in real time traffic light control," in3rd International Conference on Electronics Computer Technology, April 2011.

[9]. Lei Yang and Xiaoyu Wu, "An improved Prewitt algorithm for edge detection based on noised image,"in 4th International Congress on Image and Signal Processing, Shanghai, China, October 2011.

[10]. H Alvari, P Shakarian and J. K Snyder, "A non-parametric learning approach to identify online human trafficking", 2016 IEEE Conference on Intelligence and Security Informatics (ISI), pp. 133-138, 2016.

[11]. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[12]. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[13]. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," arXiv preprint arXiv:2207.02696, 2022.

[14]. R. Shanmugamani, Deep Learning for Computer Vision, 1st ed., Packt Publishing, 2018.