Dynamic Traffic Management

A

Project Report

Submitted for the Partial Fulfilment

of B.Tech. Degree in

COMPUTER SCIENCE & ENGINEERING

by

Akash Singh (Roll No. 2100520100111)
Varnit Saxena (Roll No. 2100520100161)
Yuvraj Singh (Roll No. 2100520100164)

Under the supervision of

Dr. Divakar Singh Yadav

Er. Ahmad Hasan

Department of Computer Science and Engineering

Institute of Engineering & Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh

June,2025

Declaration

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for requirement of any other degree.

Date :-

Submitted By :-

1. **Name :-** Akash Singh
   Roll No. :- 2100520100111
   Branch :- CSE – Self Finance
   Signature

2. **Name :-** Varnit Saxena
   Roll No. :- 2100520100161
   Branch :- CSE – Self Finance
   Signature

3. **Name :-** Yuvraj Singh
   Roll No. :- 2100520100164
   Branch :- CSE – Self Finance
   Signature

Certificate

This is to certify that the project report entitled **"Dynamic Traffic Management"** presented by **Akash Singh, Varnit Saxena and Yuvraj Singh** in the partial fulfilment for the award of degree of Bachelor of Technology in Computer Science and Engineering, is a record of the original work carried out by them under my supervision and guidance in the Department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow.

It is also certified that this project has not been submitted at any other Institute for the award of any other degrees to the best of my knowledge.

Dr. Divakar Singh Yadav
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Er. Ahmad Hasan
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgement

It gives us immense pleasure to express our deepest gratitude to all those who have supported and guided us throughout the development of this project, titled **"Dynamic Traffic Management"**.

First and foremost, we would like to extend our sincere thanks to our respected **Supervisor, Dr. Divakar Singh Yadav**, for his invaluable guidance, continuous encouragement and insightful feedback at every stage of the project. His mentorship and deep technical knowledge have been instrumental in shaping our ideas into a practical and scalable solution.

We are also profoundly grateful to our **Co-Supervisor, Er. Ahmed Hasan**, for his constant support, constructive suggestions and motivation throughout the course of this work. His input played a key role in refining our methodology and improving the overall quality of our project.

We would also like to thank our esteemed **Head of Department, Dr. Girish Chandra**, for providing us with the necessary resources, infrastructure and a conducive academic environment that enabled us to carry out this project smoothly. His encouragement towards innovation and research excellence has been truly inspiring.

Lastly, we extend our heartfelt thanks to all faculty members, peers and well-wishers who supported us directly or indirectly. Their assistance and moral support contributed significantly to the successful completion of our project.

Table of Contents

List of Figures

Abstract

This thesis presents the development of a **Dynamic Traffic Management**, an AI-powered solution designed to regulate urban traffic flow intelligently based on real-time vehicular behaviour. The system leverages deep learning-based computer vision techniques, specifically the YOLOv8 object detection model integrated with the DeepSORT tracking algorithm, to identify, track and categorize incoming vehicles from recorded video feeds. The key vehicle types monitored include cars, motorcycles, buses and trucks, each contributing differently to traffic congestion. Unlike traditional timer-based traffic signal systems, this project introduces a dynamic approach wherein the green light duration for each lane is allocated proportionally to the average incoming speed of the vehicles in that lane. Vehicles are assigned unique IDs for accurate tracking and their speeds are calculated using centroid displacement across frames. The average speed is then computed for all vehicles in a lane to determine the congestion level and automatically adjust the signal timing. Slower lanes receive longer green durations to enable smoother traffic clearance, especially in lanes congested with slower vehicles like auto-rickshaws and heavy trucks.

The system produces annotated output videos with tracking overlays and generates a structured CSV report containing vehicle IDs, types and average speeds. Developed using Python and OpenCV, the backend employs optimized video processing pipelines to reduce CPU load and utilize GPU acceleration for inference. Experimental results on simulated cross-section video data demonstrate the system's effectiveness in managing lane-wise traffic adaptively and reducing overall congestion.

This project illustrates the potential of integrating real-time object detection, tracking and speed analytics for building intelligent, data-driven urban traffic systems that prioritize both efficiency and fairness in signal distribution.

Building on the foundation of real-time vehicle detection and speed-based signal adjustment, the system also incorporates scalability features to adapt to varying traffic environments. By leveraging modular components, it can be extended to multi-intersection networks, where data from several junctions are aggregated to optimize traffic flow across a larger urban area. Additionally, the architecture supports integration with existing traffic management infrastructures, allowing seamless upgrades without requiring complete system overhauls. This flexibility makes it suitable for deployment in diverse city layouts and traffic conditions, promoting smarter urban mobility solutions.

Furthermore, the project emphasizes robustness and reliability by implementing advanced filtering techniques to handle occlusions, lighting changes, and weather variations that commonly affect video feed quality. The use of DeepSORT ensures consistent tracking despite temporary loss of visual cues, while periodic calibration routines maintain accuracy in speed estimation. These features contribute to the system's resilience, ensuring continuous and dependable performance in real-world scenarios. Overall, this approach demonstrates how intelligent vision-based analytics can empower municipalities to enhance road safety, minimize delays, and support sustainable transportation initiatives.

1 Introduction

1.1 Background

Urbanization and the increasing number of vehicles on the road have led to significant challenges in traffic management, especially at busy intersections. Traditional traffic signal systems often operate on fixed-timer mechanisms, lacking the flexibility to adapt to fluctuating traffic patterns throughout the day. As a result, they contribute to unnecessary delays, longer travel times, and increased fuel consumption. This inefficiency is further exacerbated during peak hours, where imbalanced lane congestion causes bottlenecks and frustration among commuters.

To address these growing concerns, intelligent traffic systems have emerged as a viable alternative, combining real-time data analysis with adaptive signal control. Among these, computer vision-based approaches offer a promising solution due to their ability to analyze traffic using surveillance footage without the need for physical sensors or infrastructure changes. By utilizing video data from traffic cameras, it becomes possible to detect, classify, and monitor vehicles dynamically, enabling the system to respond to real-time conditions on the road.

The concept of dynamic traffic management focuses on allocating green signal durations not based on rigid schedules, but on actual lane activity such as vehicle density or flow rate. However, counting vehicles alone does not always reflect congestion accurately, especially when a lane is filled with slow-moving traffic like auto-rickshaws or trucks. Hence, measuring the average incoming speed of vehicles offers a more precise and fair indicator of how congested a lane truly is. A lower average speed typically signifies heavier congestion and justifies longer green light allocation, while faster-moving lanes require shorter durations.

This project builds on that principle, introducing a system that combines real-time vehicle detection and tracking using deep learning models such as YOLOv8 and DeepSORT. It not only recognizes key vehicle types but also estimates their movement speeds and aggregates this data to adjust signal timings intelligently. Through this approach, the system aims to reduce idle time, improve traffic flow efficiency, and adapt to varying road conditions without manual intervention. The outcome is a scalable, camera-based solution suitable for modern cities aiming to adopt smart infrastructure without extensive hardware investments.

Conventional traffic management systems struggle to cope with the dynamic and unpredictable nature of urban traffic, particularly at busy intersections where fixed-timer signals lead to inefficiencies and increased commuter frustration. These systems fail to account for real-time variations in traffic volume and vehicle behavior, often resulting in uneven lane utilization, longer waiting times, and higher fuel consumption. The inability to distinguish between different vehicle types or factor in their speeds further limits the effectiveness of existing approaches. Addressing these challenges requires a smart, adaptive solution capable of analyzing traffic patterns in real-time using existing infrastructure. This project identifies the core problem as the lack of intelligent, responsive traffic control mechanisms and proposes a computer vision-based system that leverages deep learning for accurate vehicle detection, tracking, and speed estimation, ultimately enabling smarter green light allocation and improved traffic flow efficiency.

1.2 Problem Statement

In many rapidly growing urban and semi-urban areas, managing traffic at busy intersections has become increasingly difficult due to rising vehicle density and unpredictable traffic patterns. Traditional signal systems often operate on fixed timers that fail to adjust to real-time road conditions, resulting in longer waiting times, inefficient lane utilization, and higher fuel consumption. These outdated systems contribute to commuter frustration and hinder overall traffic flow, especially during peak hours when congestion is at its worst.

Moreover, current traffic control methods tend to rely on simplistic metrics such as vehicle count, which do not accurately represent the true congestion level of a lane. For instance, a lane filled with slow-moving vehicles like auto-rickshaws or heavy trucks may experience greater delays despite having fewer vehicles than a faster-moving lane. This lack of contextual understanding leads to unfair signal timing and worsens traffic bottlenecks.

Another significant limitation lies in the reliance on expensive physical infrastructure—such as inductive loop sensors or radar systems—to gather traffic data. These installations are not only costly but also difficult to maintain and scale, especially in cities with constrained budgets or aging infrastructure. As a result, many regions are unable to implement smart traffic solutions despite having a clear need for them.

This project addresses these pressing issues by proposing a computer vision-based intelligent traffic system that adapts signal timing according to real-time traffic conditions. The system tackles the key challenges through:

1. A speed-based congestion metric that considers both vehicle count and movement speed for fairer signal allocation.
2. Vehicle classification using deep learning to account for slower vehicles that impact lane efficiency more significantly.
3. Real-time video processing from standard traffic cameras, eliminating the need for additional sensor hardware.
4. Frame-skipping and resolution scaling techniques to reduce system load and enable deployment even on low-power edge devices.
5. Scalable and cost-effective architecture that leverages existing surveillance infrastructure for broader implementation across city intersections.

By integrating these capabilities, the system offers an adaptive, data-driven approach to traffic management that improves intersection efficiency while remaining accessible and affordable for a wide range of urban settings.

1.3 Objectives

The primary objective of the "Dynamic Traffic Management" project is to enhance the efficiency and responsiveness of traffic signal control by leveraging real-time video analysis and deep learning. This approach aims to overcome the limitations of traditional fixed-timer systems, which are often unresponsive to fluctuating traffic conditions and contribute to congestion, delays, and commuter dissatisfaction. To address these issues, the project sets out to achieve the following specific objectives:

1. To develop an intelligent traffic signal control system that dynamically adjusts green light durations based on real-time traffic conditions rather than preset timers, ensuring adaptive response to congestion levels.
2. To utilize computer vision techniques with pre-trained models such as YOLOv8 for accurate detection and classification of vehicles, including cars, buses, trucks, and motorcycles, from standard CCTV footage.
3. To implement DeepSORT-based tracking for assigning unique IDs to each vehicle, enabling continuous monitoring of vehicle flow without repeated counting of the same vehicle.
4. To measure the average incoming speed of vehicles in each lane and use it as a core metric for determining congestion severity and optimizing signal timing accordingly.
5. To reduce system load and increase processing efficiency by incorporating load optimization techniques such as frame skipping, resolution scaling, and GPU acceleration where available.

	Project Objectives	**Project Outcomes/Deliverables**
1.	Optimize traffic signals using machine learning.	A system that dynamically adjusts signal timings based on traffic.
2.	Detect vehicles using video processing techniques.	Real-time vehicle counting using YOLO.
3.	Simulate real-world traffic scenarios for testing.	A simulation environment to validate system performance.
4.	Ensure fair rotation across all traffic lanes dynamically.	Cyclic signal allocation algorithm ensuring fairness for all.
5.	Reduce environmental impact by minimizing idle time at signals.	A system that lowers fuel consumption and emissions through optimized signal timings.

TABLE 1: Project Objectives and Corresponding Outcomes

1.4 Scope of Project

This project focuses on the design, development, and testing of a prototype intelligent traffic signal management system that dynamically adjusts signal timing based on real-time vehicle movement. The primary emphasis is on demonstrating the feasibility of using computer vision and deep learning for traffic monitoring, vehicle tracking, and adaptive signal control, rather than deploying a full-scale city-wide infrastructure solution. The scope is clearly defined by the following boundaries and limitations:

1. **Traffic Scenario Coverage:** The system will be tested on pre-recorded video footage representing single-lane traffic at intersections. It will simulate dynamic signal behaviour for one lane at a time. Multi-lane, multi-junction, or real-time integration with multiple live feeds is beyond the current scope.

2. **Vehicle Categories:** The system will focus on detecting and classifying four primary vehicle types—cars, buses, trucks, and motorcycles—using a pretrained YOLOv8 model. Auto-rickshaws, bicycles, and other vehicle types may be detected but are not explicitly handled in the prototype.

3. **Signal Control Simulation:** While the system dynamically calculates green light durations based on average incoming vehicle speed, actual hardware-based signal integration is not included. The green signal behaviour will be simulated digitally within the program.

4. **Data Source and Environment:** The system will operate on pre-recorded videos captured from static traffic surveillance cameras. Real-time video streaming, cloud- based deployment, or direct roadside deployment are not included in this phase.

5. **Speed Estimation Method:** Vehicle speed will be estimated using object tracking across frames and positional displacement over time. Calibration against real-world GPS or sensor data is not part of the current scope and all speed calculations are approximate.

6. **Hardware and Optimization:** The system will be developed and tested on a local machine with GPU acceleration if available. Scalability testing on edge devices, embedded boards, or cloud servers will not be covered in this version.

7. **Reporting and Analytics:** The prototype will generate CSV reports containing details such as vehicle ID, type, and estimated average speed. Advanced analytics dashboards, visualizations, or API integration for traffic departments are outside the scope.

The scope of this project centres on developing a functional prototype that applies modern computer vision and deep learning techniques to regulate traffic dynamically. By analysing real-time vehicle movement through video feeds, the system adjusts green signal durations in response to actual congestion rather than relying on static timers. This allows for smarter signal distribution and reduced idle time at intersections. This prototype leverages advanced tools like YOLOv8 for vehicle detection and DeepSORT for tracking, enabling precise identification of traffic patterns.

1.5 Organisation of Thesis

The remainder of this thesis is structured as follows:

Chapter 2: Literature Review
Reviews prior research and developments in intelligent traffic systems, object detection models (YOLO), multi-object tracking (DeepSORT), speed estimation methods, signal timing algorithms and real-time video analytics.

Chapter 3: Methodology
Explains the system architecture and core functional modules including vehicle detection and classification, tracking and speed estimation, dynamic green time allocation and cyclic signal control. It also outlines the technologies used (YOLOv8, DeepSORT, OpenCV), the data flow from video input to traffic signal control and CSV-based data logging.

Chapter 4: Project Interface
Presents the output video frame overlay with bounding boxes, lane-wise detection and real time countdown timers.

Chapter 5: Limitations
Discusses the constraints of the current system, such as the reliance on static camera angles, bounding box over-scaling, absence of real-time GPS input, lack of emergency vehicle prioritization, hardware dependency and computational load in multi-lane processing.

Chapter 6: Future Scope
Proposes potential enhancements including custom-trained YOLO models (with auto-rickshaw class), multi-camera integration.

Chapter 7: Monetization Strategies
Explores scalable revenue models such as government licensing, B2B analytics subscriptions, violation-based fine automation, traffic data sales, franchise deployment, smart ad integration on digital signboards and white-labelling for regional partners.

Chapter 8: References
Lists all cited sources and research materials.

2 Literature Review

This chapter reviews existing technologies and research relevant to the development of a dynamic traffic signal management system based on computer vision. The discussion is organized around the key technological components of the system, including traditional and adaptive traffic control systems, vehicle detection models, object tracking algorithms, speed estimation techniques, camera-based traffic monitoring and data-driven signal timing methods.

2.1 Traditional and Adaptive Traffic Control Systems

Conventional traffic signals are predominantly time-based, using static cycles to manage flow at intersections. However, such systems often fail to respond to varying traffic volumes, leading to inefficiencies during peak hours and underutilization during off-peak periods. According to Gupta [1], adaptive signal control systems (ASCS) represent an improvement by adjusting signal timing based on real-time traffic data. Commercial solutions like SCATS (Sydney Coordinated Adaptive Traffic System) and SCOOT (Split Cycle Offset Optimization Technique) dynamically optimize signal phases using sensor inputs and predefined logic.

Despite their effectiveness, these systems require expensive hardware installations such as inductive loops and radar sensors, limiting scalability and adoption in cost-sensitive regions. The proposed project seeks to overcome this limitation by using standard surveillance cameras for input, eliminating the need for embedded road infrastructure.

2.2 Deep Learning for Vehicle Detection

Recent advancements in deep learning have led to highly accurate object detection models such as YOLO (You Only Look Once), SSD and Faster R-CNN. Among them, YOLOv8 stands out for its real-time performance and high accuracy in detecting multiple object classes, including vehicles. Bochkovskiy [2] emphasize YOLO's capability to deliver frame-by-frame object detection at high speeds, making it ideal for traffic surveillance applications.

YOLOv8 has been trained on large datasets like COCO, which includes common vehicle types such as cars, trucks, buses and motorcycles. This enables rapid deployment in traffic environments without requiring custom training for basic vehicle classification. The model's single-stage architecture ensures low latency, making it suitable for real-time traffic video analysis.

2.3 Object Tracking with DeepSORT

While object detection identifies vehicles in individual frames, tracking is essential to maintain continuity across frames and prevent double counting. The DeepSORT algorithm combines Kalman filtering with appearance-based features to associate object identities over time, even under occlusion or erratic motion. Wojke [3] demonstrate that DeepSORT enhances robustness in multi-object tracking compared to earlier methods like SORT and optical flow.

In the proposed system, DeepSORT assigns a unique ID to each detected vehicle, enabling the estimation of per-vehicle speed and flow rate. This tracking capability forms the backbone for congestion assessment and signal control logic.

2.4 Speed Estimation Using Video Analytics

Traditional speed measurement relies on radar, LIDAR, or GPS data, which are either costly or impractical in large-scale deployments. Video-based methods, in contrast, offer a passive and infrastructure-light alternative. According to Sundar [4], vehicle speed can be estimated by measuring the displacement of bounding box centroids across frames, given a known frame rate and calibration factor.

Although precise speed estimation may require calibration using reference distances in the video frame, approximate speeds can still yield reliable congestion indicators. This is particularly useful in lanes occupied by slow-moving vehicles like auto-rickshaws or heavy trucks, where vehicle count alone fails to capture true congestion levels.

2.5 Camera-Based Traffic Monitoring

Camera-based monitoring systems are gaining popularity due to their affordability and integration with existing surveillance infrastructure. As noted by Wang [5], cameras offer high resolution, multi-lane coverage and facilitate real-time visual inspection along with automated processing.

Unlike physical sensors, cameras are non-intrusive and easy to maintain. Moreover, combining camera footage with AI models allows for scalable and adaptive traffic management without large infrastructure investments. This aligns with the project's objective of developing a cost-effective solution deployable in developing cities or smart urban zones.

2.6 Data-Driven Signal Timing Algorithms

Recent research advocates for traffic light control based on real-time vehicle flow characteristics rather than rigid time cycles. Studies by Kulkarni [6] propose adaptive algorithms that prioritize lanes with higher congestion levels or lower average vehicle speeds. Machine learning models trained on historical traffic data have also shown promise in predicting traffic volume and optimizing signal schedules dynamically.

However, this project takes a rule-based approach focused on average incoming speed as the primary metric. It simplifies the implementation while still enabling dynamic signal [7] timing based on real-time traffic activity, a middle ground between full AI optimization and basic timer-based control.

3 Methodology

This chapter presents a detailed account of the methodology adopted for the design, development and deployment of the Dynamic Traffic Management System. The objective is to build an AI-based smart traffic controller that uses computer vision to detect and track vehicles, calculate their average incoming speeds and allocate green signal time dynamically to ensure smoother traffic flow. The system uses pre-recorded lane videos, modern object detection algorithms and intelligent traffic signal logic to achieve fair and adaptive management of road congestion.

3.1 Training of Object Detection Model

The heart of the system lies in accurate detection of vehicles on road lanes. For this, the YOLOv8x model is used.

1. **Model Selection and Setup:** The YOLOv8x (You Only Look Once – Version 8 Extra Large) model by Ultralytics was selected due to its high precision in detecting multiple classes with real-time speed. The pre-trained weights on the COCO dataset allow recognition of standard vehicle types like cars, buses, trucks and motorcycles.
2. **Custom Class Inclusion (Auto-Rickshaws):** Since auto-rickshaws are common in Indian roads and are not part of default YOLO classes, a future extension involves training the model on a custom dataset (using tools like Roboflow) to classify "auto" as a separate class.
3. **Model Deployment:** The YOLOv8 model is loaded and deployed in Python using the Ultralytics library. The model is optimized to run on GPU if available, otherwise it falls back on CPU, ensuring flexibility in system configuration.
4. **Detection Thresholds:** The confidence threshold and Intersection Over Union (IoU) values are adjusted to balance between detection sensitivity and accuracy.

3.2 Vehicle Tracking and Classification

After detection, the system tracks the movement of each vehicle across frames to avoid duplicate counts and to classify movement patterns.

1. **DeepSORT Tracking Algorithm:** DeepSORT (Simple Online and Realtime Tracking) assigns a unique ID to each vehicle. It uses Kalman Filters and the Hungarian Algorithm to maintain tracking even when occlusions occur (e.g., one vehicle hides behind another temporarily).
2. **Vehicle ID Assignment:** Each vehicle is identified by a unique ID, which persists across frames. This prevents the same vehicle from being counted multiple times as it moves through the video.

3. **Bounding Box Management:** Each detected vehicle is displayed with a bounding box and its class label (e.g., "Car", "Truck"). The boxes are dynamically adjusted to remain accurate and tight-fitting.

4. **Class-Based Categorization:** The system tracks vehicle type to allow category-specific analysis (e.g., autos tend to be slower and may affect congestion differently). Each vehicle entry includes type, ID and confidence score.

3.3 Speed Estimation

The system calculates the incoming speed of traffic using motion tracking and frame rate information.

1. **Centroid Displacement:** The position of each vehicle's centroid (centre point of its bounding box) is calculated in each frame. The pixel distance covered between consecutive frames is recorded.

2. **Time Calculation:** The time difference between the two frames (based on video FPS) is used to estimate speed in pixels/second. This is converted to real-world units (like km/h or m/s) based on a pre-defined pixel-to-meter ratio, which may be calibrated using reference points in the camera view.

3. **Smoothing and Averaging:** To handle fluctuations due to detection noise or jitter, a moving average of speed is taken for each vehicle over multiple frames. Then, the average speed per lane is calculated by aggregating across all tracked vehicles.

4. **Handling Static or Slow Vehicles:** Vehicles with negligible movement (e.g., stuck in traffic) are identified by minimal centroid shift and are assigned near-zero speed, which helps detect congestion accurately.

3.4 Time Allocation Logic for Green Signal

The most critical output of the system is the dynamic calculation of how much green time to allocate to each lane.

1. **Green Time Formula:** The system uses a proportional model where:

$$\text{Green Time} = \min\left(60,\ 10 + \text{int}(\text{total\_vehicle\_count} \times 0.2 + \text{avg\_speed} \times K)\right)$$

Here, *avg_speed* is the calculated average incoming speed for the lane and K is a scaling constant. The minimum and maximum green time thresholds are kept to ensure fairness and safety.

2. **Cyclic Signal Control:** A cyclic algorithm is followed to ensure all four lanes get turns in a round-robin fashion. Even if a lane has very low traffic, it gets at least the minimum green duration.
3. **Fairness and Starvation Prevention:** This design ensures that no lane is starved. Additionally, lanes with higher congestion receive more green time, which helps in faster clearance and smoother flow.
4. **Future Enhancements:** In future iterations, reinforcement learning models can be trained to optimize signal timing dynamically based on real-time traffic behaviour and historical patterns.

3.5 Real-Time Visualization and Output

To make the system user-friendly and demonstrable, a graphical interface is included.

1. **Live Detection Video Feed:** The GUI shows the input video with bounding boxes, labels and vehicle IDs overlaid in real-time.
2. **Signal Timer Display:** A countdown timer is displayed for each lane indicating how long the green signal will remain on.
3. **Vehicle Count and Speed Info:** Stats such as the total number of vehicles detected and average incoming speed are shown on the screen or terminal.
4. **CSV Export for Inference:** At the end of each cycle, the system exports vehicle logs to a CSV file with columns like Vehicle ID, Type and Estimated Speed. This is useful for performance analysis and academic reporting.

3.6 Technologies and Tools Used

The Dynamic Traffic Management System integrates multiple technologies spanning deep learning, computer vision, data processing and visualization. Each tool or library was carefully chosen to fulfill a specific functionality in the system. Below is a breakdown of the technologies and tools used:

1. **Python 3.x**
    - Core programming language used to develop the entire project.
    - Offers flexibility, support for machine learning and image processing libraries and simple syntax for rapid development.
2. **OpenCV (Open Source Computer Vision Library)**

- Used for video handling, frame extraction, drawing bounding boxes and displaying real-time GUI windows.
- Facilitates reading from video files, resizing frames, converting colour spaces and rendering output feeds.

3. **Ultralytics YOLOv8**
   - State-of-the-art object detection framework for identifying vehicles in video frames.
   - YOLOv8x model is used for high precision in real-time applications.
   - Detects multiple vehicle classes like cars, buses, trucks, motorcycles and optionally auto-rickshaws.
4. **DeepSORT (Simple Online and Realtime Tracking)**
   - Tracks detected vehicles across multiple frames.
   - Assigns unique IDs to vehicles, maintaining identity even during partial occlusion or brief disappearance.
   - Ensures accurate vehicle counting and speed estimation.
5. **NumPy**
   - Used for efficient numerical operations such as pixel displacement calculation and centroid tracking [**?**].
   - Supports array manipulation, vectorized operations and mathematical computations for performance optimization.
6. **Pandas**
   - Handles tabular data operations like storing vehicle logs and exporting to CSV.
   - Simplifies data summarization (e.g., average speed calculation per lane) and future data analysis.
7. **CSV Module / File Handling**
   - Used for structured storage of vehicle information (Vehicle ID, Type, Lane, Speed) into .csv files.
   - Enables offline review, analysis, or plotting of traffic trends.

This combination of modern machine learning frameworks, computer vision tools and visualization libraries make the system both powerful and extensible — allowing for future improvements like voice commands, IoT signal integration, or dashboard-based admin control.

3.7 Data Flow Summary

The following step-by-step breakdown describes how data is processed through the system—from input video to signal timing and result output:

1. **Video Feed Acquisition**
   - The system begins by loading pre-recorded video files (one per lane).
   - Each video simulates the live feed from a traffic surveillance camera positioned at an intersection.
2. **Frame Extraction and Preprocessing**
   - Frames are extracted sequentially from the video stream using OpenCV.
   - To reduce computational load, only every nth frame (e.g., every 2nd or 3rd) is selected for analysis.
   - Each frame is resized and normalized to match the input size expected by the YOLOv8 model.
3. **Vehicle Detection via YOLOv8**
   - The YOLOv8x object detection model is applied to each selected frame.
   - It outputs bounding boxes, class labels (e.g., car, bus, motorcycle) and confidence scores for each detected object.
   - Non-vehicle objects (like pedestrians or traffic signs) are ignored based on class filtering.
4. **Vehicle Tracking with DeepSORT**
   - Detected vehicles are passed to the DeepSORT tracker.
   - Each vehicle is assigned a persistent and unique ID across multiple frames.
   - The tracker updates vehicle positions, manages ID consistency and handles re-identification in case of occlusion.
5. **Vehicle Classification and Labelling**
   - Each tracked vehicle retains its class label (e.g., “Car”, “Auto”) from YOLO.
   - Labels, IDs and confidence scores are displayed as overlays on the frame using bounding boxes.
6. **Speed Estimation Per Vehicle**
   - For every tracked vehicle, its centroid (centre of bounding box) is calculated per frame.
   - Speed is computed using the formula:

$$\text{Speed} = \frac{\text{Pixel Distance}}{\text{Time}}$$

- Frame rate (FPS) is used to estimate time elapsed between frames.
- Speed is stored and updated for each vehicle across multiple frames.

7. **Aggregation of Lane-Wise Traffic Data**
   - Vehicles are grouped based on the input video lane they belong to (Lane 1, Lane 2, etc.).
   - For each lane, the system calculates:
     - Total number of unique vehicles
     - Average incoming speed of traffic
     - Vehicle type distribution (cars, trucks, autos, etc.)

8. **Dynamic Signal Time Calculation**
   - Based on the average speed and vehicle density, green signal time is calculated using:

     $$\text{Green Time} = \min(\text{MAX\_TIME}, \text{MIN\_TIME} + \text{int}(\text{Total\_Vehicle\_Count} \times 0.2 + \text{avg\_speed} \times K))$$

   - Each lane receives green signal in a cyclic order and its duration is dynamically adapted to current congestion.

9. **Visualization and Real-Time Output**
   - The processed frame is displayed live using a GUI (OpenCV window), showing:
     - Detected vehicles with bounding boxes
     - Vehicle ID, type and speed
   - Active lane and its signal countdown timer

10. **CSV Logging and Export**
    - At each time step or after processing, details of each unique vehicle are saved in a CSV file with:
      - Vehicle ID
      - Type
      - Lane Number
      - Estimated Speed
    - This data can be analysed later for accuracy checks, traffic studies and reporting.

11. **Signal Rotation and Loop**
    - Once green time elapses for a lane, the signal turns red and the next lane enters green.
    - The cycle continues for all four lanes in a loop.
    - Each cycle dynamically adapts based on traffic state in that lane during its turn.

This detailed data flow ensures the system performs intelligent detection, speed-based congestion analysis and adaptive signal timing — all while keeping processing efficient and visual feedback real-time.

FIGURE 1: Model Training-1

FIGURE 2: Model Training-2

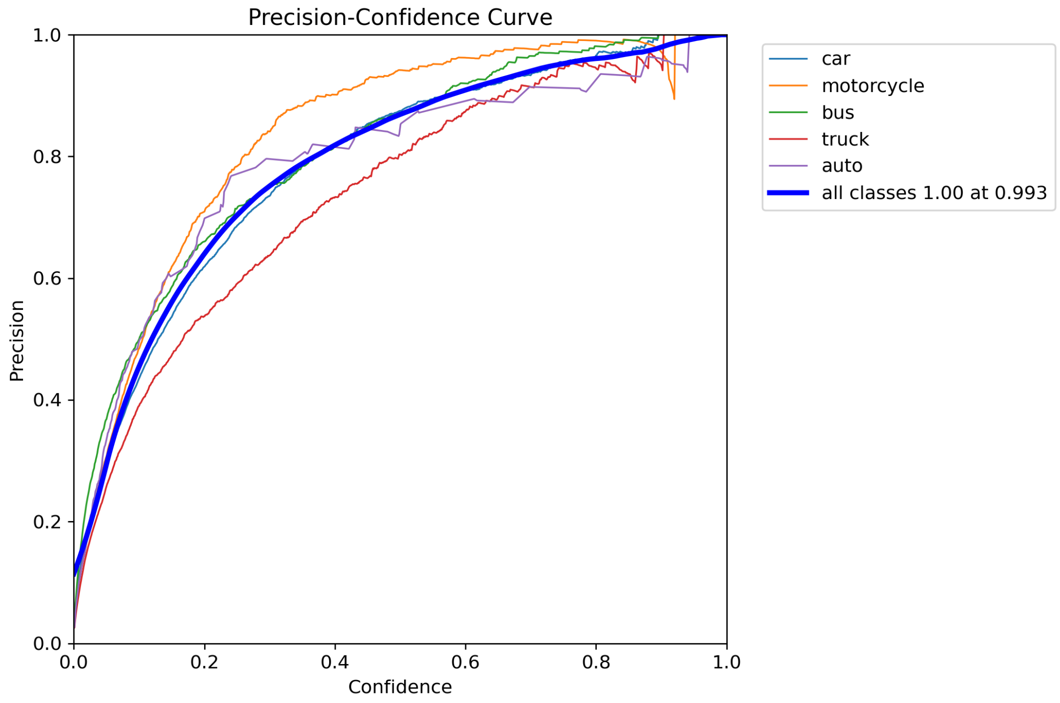

FIGURE 3: Precision-Confidence Curve

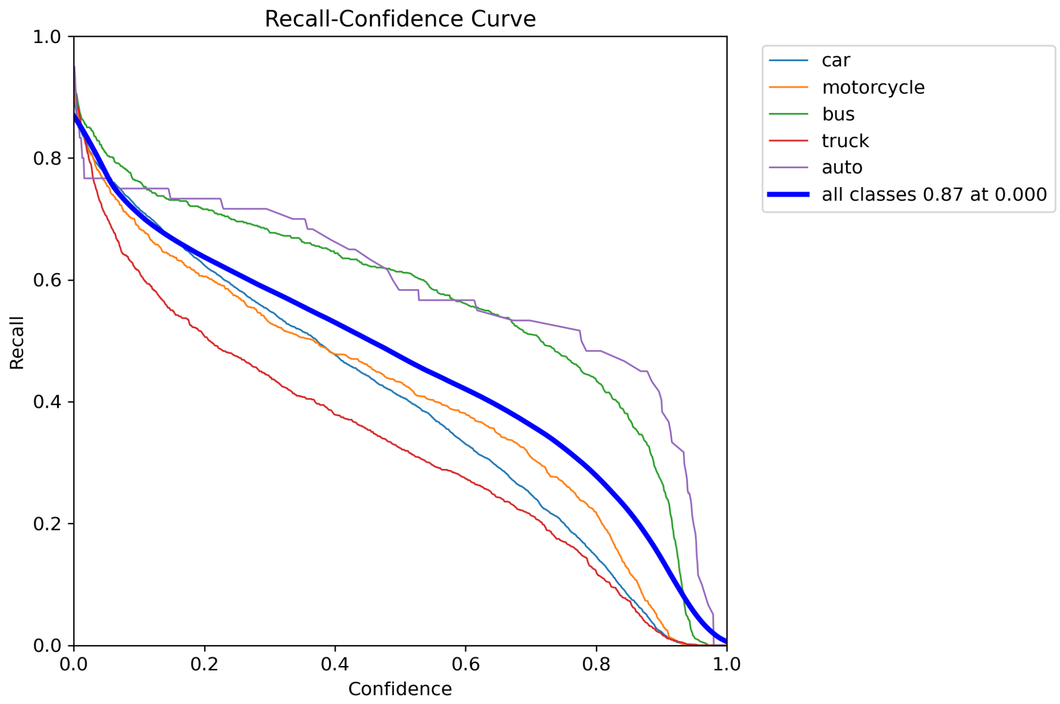

FIGURE 4: Recall-Confidence Curve

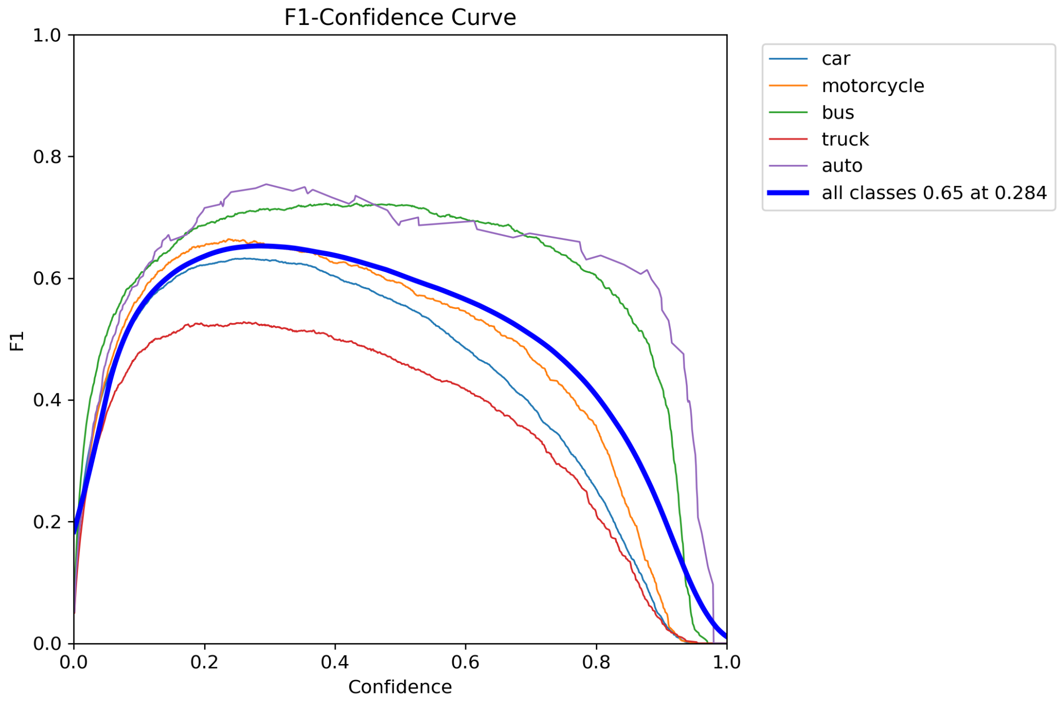

FIGURE 5: F1-Confidence Curve

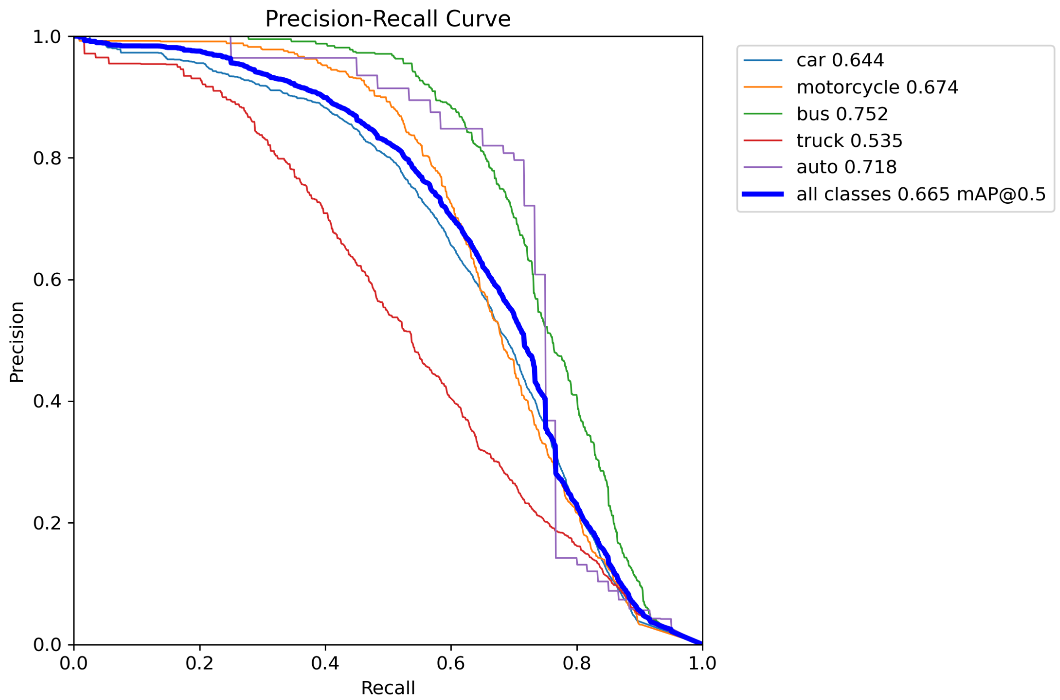

FIGURE 6: Precision-Recall Curve

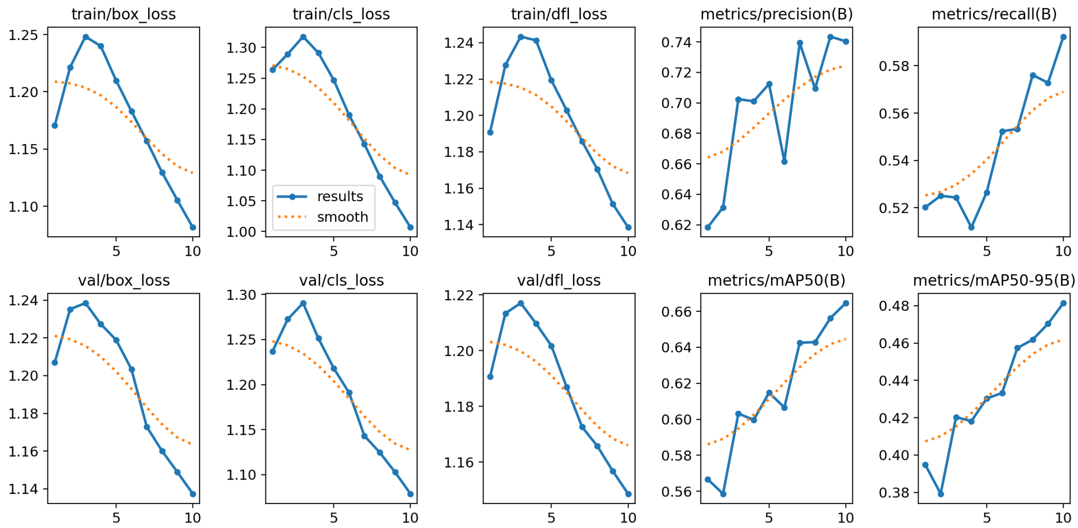

FIGURE 7: Resultant Curves

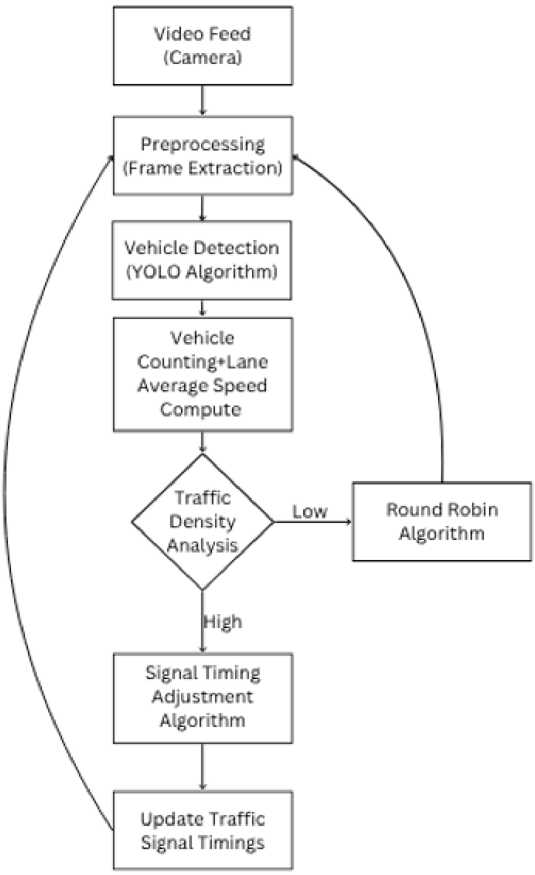

FIGURE 8: Flowchart

4 Project Interface

The output interface of the dynamic traffic management system is thoughtfully designed to provide a clear, interactive, and intuitive visualization of real-time vehicle detection and traffic analytics. As each uploaded video frame is processed, the system overlays bounding boxes around detected vehicles—such as cars, buses, trucks, and motorcycles—each accompanied by a unique vehicle ID and class label. This enables continuous visual tracking and differentiation of individual vehicles across frames.

To emulate a realistic traffic scenario, the interface segments the scene into four distinct lanes by analyzing the spatial positions of detected vehicles within the frame. Each vehicle is assigned to a lane accordingly. The system estimates vehicle speeds based on displacement of their centroids between consecutive frames. These individual speeds are aggregated to calculate the average incoming speed per lane, which forms the basis for dynamically adjusting green light durations. This adaptive approach ensures fair and responsive signal cycles that prioritize lane congestion levels effectively. A standout feature of the interface is the integration of live countdown timers displaying the remaining green light duration for each lane. These timers update in real-time to clearly communicate signal phase progress, enhancing transparency and interpretability for traffic engineers and urban planners monitoring the system. Sample screenshots from video outputs showcase annotated bounding boxes, lane segmentation overlays, vehicle trajectories, and active signal states, offering a comprehensive and visual demonstration of the system's core capabilities in vehicle identification, tracking, and adaptive signal control.

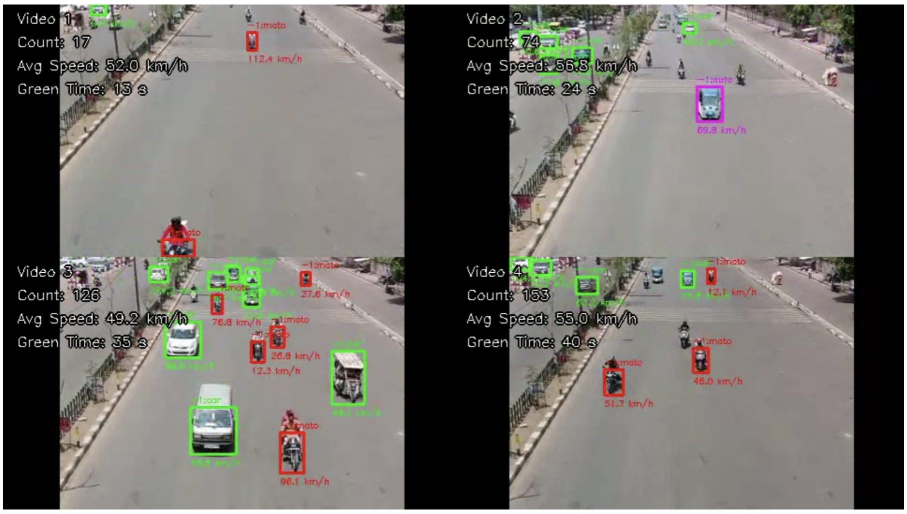

FIGURE 9: Sample Four Lane Output

5 Limitations

Despite the innovation and promise brought by the "Dynamic Traffic Management System"— through its use of real-time video analytics, YOLO-based vehicle detection and DeepSORT tracking—the system has several limitations that arise from technical, infrastructural and operational boundaries. Understanding these constraints is crucial for anticipating deployment challenges, refining the solution in future iterations and ensuring scalability in real-world urban environments.

5.1 Real-Time Detection and Processing Constraints

The system currently processes video input from traffic cameras using deep learning models on local or cloud-based machines. While this enables real-time vehicle detection and speed estimation, there are several constraints:

1. High Computational Load: Running YOLOv8x along with DeepSORT continuously on multiple video streams is computationally intensive. It may cause performance degradation on low-end hardware or without proper GPU support.
2. Frame Skipping Trade-offs: To reduce system load, frames are skipped (e.g., processing every 2nd or 3rd frame), which can lead to missed vehicle detections, especially in fast-moving traffic.
3. Latency in Inference: Delays may occur between detection and action, especially when cloud inference is used or multiple lanes are processed concurrently, affecting signal response time.
4. Limited Parallel Processing: The current architecture may not handle multiple intersections simultaneously unless horizontally scaled or containerized, making large scale deployment harder.

5.2 Limitations in Vehicle Speed Estimation

The system calculates vehicle speed using bounding box displacement across frames. While effective for average traffic flow, this method has limitations:

1. Accuracy Variation: Bounding box centre-point estimation can be affected by object detection jitter, camera angle, or motion blur, leading to noisy speed readings.
2. No Depth or Real-World Scaling: Without camera calibration or reference markers, speed is measured in pixel displacement. Converting this to accurate meters/second requires manual tuning and field calibration.
3. Occlusion and Overlaps: When vehicles overlap or are occluded (e.g., buses covering motorcycles), tracking continuity is lost, affecting both counting and speed estimation.

4. Short-Term Tracking: DeepSORT does not maintain long-term trajectory memory, which limits long-duration observations or trajectory-based analytics.

5.3 Static Camera and Environmental Dependency

The system relies entirely on static CCTV footage. Environmental and installation-related issues can heavily impact performance:

1. Fixed View Limitations: Cameras with fixed positions cannot detect lane changes or side road traffic, limiting holistic traffic awareness.
2. Night-Time Detection: In low-light or night-time conditions, vehicle detection performance may drop significantly unless thermal or IR-based cameras are used.
3. Weather Sensitivity: Heavy rain, fog, or glare can blur frames and reduce detection reliability, especially in outdoor environments without covered enclosures.
4. Camera Misalignment or Tampering: A small shift in camera angle (due to vibration or mishandling) can distort tracking zones and result in poor or misleading analytics.

5.4 Limited Multi-Lane and Intersection Synchronization

Currently, the system processes a single lane from a pre-recorded feed. Future real-world deployment will face synchronization challenges:

1. Single-Lane Processing: The current prototype does not handle simultaneous inputs from all four lanes of a junction, limiting practical applicability.
2. Signal Phasing Complexity: Coordinating signal cycles based on live density from multiple directions (e.g., main road vs. side road) requires complex logic currently not implemented.
3. Pedestrian Crossing Ignorance: The system does not account for pedestrian signals or crosswalks, potentially causing safety issues in busy intersections.
4. Emergency Prioritization: There is no mechanism to identify emergency vehicles (e.g., ambulances, fire trucks) and prioritize them during congestion.

5.5 Dependence on Pre-Processed Video and Manual Configuration

As of now, the model works with offline recorded video. This causes several operational bottlenecks:

1. Lack of Real-Time Testing: Without a live video pipeline, field performance (including latency, adaptability and signal timing adjustment) cannot be validated completely.
2. Manual Lane and Region Setup: Regions of interest (ROIs) for detection, such as counting lines or stop zones, must be manually configured per camera view, reducing scalability.
3. No Self-Learning: The system doesn't learn or adapt from changing traffic patterns over time unless manually retrained.
4. Hardcoded Timing Logic: While green signal time is dynamically calculated, the range (min 5s, max 30s) and scaling factor (K) are currently static and not context aware.

5.6 Limited Admin Control and Reporting Tools

The current version lacks advanced admin and reporting capabilities that would assist traffic authorities:

1. No Central Dashboard: There is no unified dashboard to monitor multiple intersections or view analytics across zones.
2. Minimal Reporting: The system does not provide downloadable reports or visual graphs summarizing traffic patterns, peak times, or congestion zones.
3. No Alert Mechanism: In the case of unusual congestion, signal failure, or camera disconnect, the system does not notify the admin or trigger fallbacks.
4. Lack of Simulation Mode: There's no sandbox or test mode for admins to simulate scenarios (e.g., festival traffic) before deploying changes.

5.7 Data Storage and Security Concerns

Handling vehicle data, speed and location information introduces privacy and data security challenges:

1. Lack of Data Anonymization: While only visual analytics are used, storing full video feeds or snapshots without anonymizing vehicle plates or driver faces can raise privacy concerns.
2. No Role-Based Access: The system does not implement granular access control for different admin users, risking unauthorized configuration changes.
3. Local Storage Overhead: Saving full-length videos and analytics data on local disks without compression or cloud sync can lead to storage exhaustion.
4. Vulnerability to Tampering: Without tamper-proof logging or encryption, traffic records could be altered or deleted in case of a breach.

5.8 Hardware and Infrastructure Dependency

For real-world deployment, the system must rely on well-maintained hardware and connectivity:

1. Need for High-Resolution Cameras: Poor-quality feeds can significantly reduce detection accuracy, necessitating modern surveillance infrastructure.

2. No Redundancy Mechanism: If a camera or node fails, there's currently no failover or backup process to ensure continued detection.

3. Power and Network Dependency: The system assumes uninterrupted power and internet access, which may not be realistic in some semi-urban zones.

4. Maintenance Overhead: Camera cleaning, lens alignment and firmware updates require human intervention, increasing operational cost.

6 Future Scope of the Project

The “Dynamic Traffic Management System” project marks a transformative step toward the intelligent regulation of urban intersections using AI-driven technologies. The current implementation—leveraging YOLOv8x for vehicle detection, DeepSORT for real-time tracking and average-speed-based green signal allocation—lays the groundwork for a smarter, more adaptive and data-driven urban mobility system. However, as cities continue to grow and transportation systems evolve, there are vast opportunities to expand the project’s technical capabilities, scalability, real-time responsiveness and operational robustness.

This section outlines future development opportunities across multiple domains: technological, functional, operational and strategic.

6.1 Adaptive AI Signal Control

The present system uses a fixed logic of averaging vehicle speeds and densities to determine green light durations.[9] This can evolve into a self-learning, adaptive traffic control mechanism:

1. Reinforcement Learning Models: Use deep reinforcement learning to dynamically adjust signal times based on live feedback (e.g., queue lengths, weather, time of day).
2. Congestion Prediction: Train AI models using historical traffic data to predict congestion and adjust signal cycles preemptively.
3. Incident Detection: Integrate anomaly detection to identify accidents or stalled vehicles and adjust signal priorities accordingly.

6.2 Integration with Smart City Infrastructure

Future versions can integrate with broader smart city ecosystems:

1. IoT Integration: Connect the system to road-embedded sensors, smart lights and edge cameras to improve data granularity and responsiveness.
2. Public Transport Priority: Coordinate with buses or emergency vehicles using RFID or GPS data to grant signal priority at intersections.

3. Traffic Dashboard API: Create a central dashboard that integrates data from multiple intersections and offers analytics for municipal control rooms.

6.3 Enhanced Vehicle Classification and Behaviour Tracking

The current system segments vehicles into basic classes. This can be enhanced for deeper insights:

1. Custom YOLO Models: Train YOLOv8 on local traffic datasets including autos, e-rickshaws and animal-driven carts for more relevant classification.
2. Speed Profiling: Use long-range multi-frame tracking to understand average vs. peak speeds of different vehicle types.
3. Dangerous Driving Detection: Detect behaviours like wrong-way driving, speeding, or lane jumping.

6.4 Multi-Intersection and Network-Wide Optimization

The current model works per intersection. Future enhancements could optimize traffic at a network level:

1. Coordinated Signal Timing: Develop algorithms to synchronize signals across multiple junctions for green wave corridors.
2. Load Balancing: Divert traffic dynamically to less congested routes using advisory boards or mobile apps.
3. Emergency Evacuation Mode: Automatically reconfigure signal logic during natural disasters or VIP convoys.

6.5 User Interface and Visualization Enhancements

The real-time display and insights can be enhanced to aid operators and stakeholders:

1. Live Video Dashboard: Show annotated camera feeds with overlaid vehicle IDs, speeds and class labels.

2. Historical Playback: Allow traffic officials to replay historical footage with analytics overlays for forensic or optimization analysis.

3. Mobile App for Operators: Let traffic managers adjust green durations and view alerts on the go.

6.6 Integration with Law Enforcement and Penalty System

The system can evolve to support automatic law enforcement:

1. Red Light Violation Detection: Capture license plates of red-light jumpers and forward to police databases.

2. Speed Monitoring: Impose dynamic speed limits and flag violators using tracking data.

3. Parking & Obstruction Detection: Detect vehicles illegally parked or blocking free lanes.

6.7 Cross-Platform Scalability and Cloud Integration

To manage deployment in large cities, the system must scale efficiently:

1. Edge-Cloud Hybrid: Process detections at the edge for real-time needs and upload metadata to the cloud for training and archival.[10]

2. Distributed Processing: Use microservices to distribute workload across GPUs for high-density intersections.

3. Remote Configuration: Admins can update signal rules and AI models remotely via a secure cloud dashboard.

6.8 Environmental and Sustainability Extensions

Traffic management has a direct impact on emissions and public health:

1. Emission Estimation: Estimate CO_2 savings achieved due to optimized signals reducing idle time.

2. Green Wave Promotion: Promote eco-routes with smoother traffic flow to reduce braking and acceleration.
3. Integration with EV Infrastructure: Prioritize electric vehicle routes or direct them to nearby charging stations.

6.9 Community and Commuter Engagement

Involving citizens can make the system more adaptive and transparent:

1. Commuter Feedback App: Let users report issues, view average wait times and suggest improvements.
2. Real-Time Traffic Map: Display congestion levels on a public dashboard to help commuters make informed decisions.
3. Incentives for Off-Peak Travel: Suggest off-peak routes with discounts (e.g., for delivery drivers or ride-share fleets).

6.10 Legal, Ethical and Data Governance

As AI decisions begin to influence real-world movement, governance becomes vital:

1. Data Privacy Compliance: Ensure adherence to data protection laws (e.g., GDPR, DPDP Act) when collecting video feeds.
2. Bias Mitigation: Audit AI models to ensure fair treatment across vehicle types and avoid systemic prioritization.
3. Transparent Decision Logs: Maintain logs of all signal changes and decisions for audit and accountability.

The “Dynamic Traffic Management System” is not just a project—it is a foundation for an urban intelligence layer that redefines how cities think about movement, efficiency and safety. With the right enhancements, this platform can evolve into a city-wide nervous system for traffic, delivering real-time, AI-enhanced and citizen-friendly mobility.

7 Monetization Strategies and Financial Growth Opportunities

As the "Dynamic Traffic Management System" progresses from a proof-of-concept to a scalable smart city solution, integrating well-structured revenue-generation models will be vital for long-term sustainability, municipal adoption and private sector investment. The system's modular architecture, real-time AI analytics and scalable deployment model lay a solid foundation for various monetization avenues. These strategies are designed to support government partnerships, B2B models and private implementation without compromising public interest or traffic efficiency. Below are the key monetization strategies that can drive the platform's future financial success.

7.1 Government Licensing and Smart City Contracts

The primary monetization stream can come through municipal partnerships and government tenders:

1. Pilot Deployment Contracts: Offer paid pilots to municipal corporations or transport departments in tier-1 and tier-2 cities.
2. Annual Licensing Model: License the system on a per-junction or per-zone basis with annual maintenance and update agreements.
3. Smart City Collaboration: Align with national Smart City missions or urban mobility schemes to access grant-based funding or PPP (Public-Private Partnership) models.

7.2 Subscription-Based Analytics and Reporting

Authorities and traffic departments can subscribe to different tiers of analytics and dashboard capabilities:

1. Basic (Free): Access to real-time lane-by-lane traffic visuals and green signal durations.
2. Pro (Paid): Advanced heatmaps, congestion forecasts, violation tracking and downloadable reports.
3. Enterprise: Integration with central command centres, custom dashboards and API access for third-party platforms.

7.3 Violation Detection and Automated Fines

The system can be enhanced to detect and monetize traffic rule violations:

1. Red-Light Jumping & Over-Speeding: Auto-detect violations and generate fines using license plate recognition (in integration with RTOs).
2. Revenue Sharing: Share a percentage of collected fines with civic authorities for using the system as a detection backend.
3. Subscription by Enforcement Agencies: Law enforcement can subscribe to violation logs and offender data for a fee.

7.4 Data Licensing and Commercial Traffic Analytics

Aggregated and anonymized traffic data has high value for commercial, civic and academic use:

1. Urban Planners: Sell data access for traffic pattern analysis and infrastructure planning.
2. Navigation Companies: Collaborate with apps like Google Maps or MapMyIndia to provide real-time congestion and lane behaviour data.
3. Academic & Research Licensing: Offer traffic datasets to universities and research bodies for AI and transport modeling.

7.5 Hardware as a Service (HaaS) Model

For clients lacking the infrastructure, the system can be bundled as a full-service hardware + software offering:

1. On-Camera AI Processing Units: Rent edge devices (like NVIDIA Jetson) with pre-installed detection models.
2. Maintenance Contracts: Include periodic firmware upgrades, lens cleaning and system health checks.
3. Deployment Kits: Offer one-time purchase or lease options for cameras, mounts, cables and network setup.

7.6 Freemium Features for Small Municipalities

Small towns or municipalities with limited budgets can access a basic free version with options to upgrade:

1. Free Tier: Basic detection and green signal logic with watermark branding.
2. Paid Add-Ons: Real-time dashboards, zone heatmaps, route suggestion modules and downloadable reports.
3. Custom Branding: Cities can pay for white-labeled dashboards or signboards.

8 Conclusion

The Dynamic Traffic Management System developed in this project demonstrates a practical and intelligent approach to controlling urban traffic using computer vision and machine learning. By integrating state-of-the-art object detection (YOLOv8) and tracking algorithms (DeepSORT), the system successfully identifies, classifies, and tracks various types of vehicles—including cars, buses, trucks, motorcycles, and auto-rickshaws—from real-time or recorded video feeds. This capability forms the backbone of a modern, data-driven traffic control mechanism that dynamically adapts to real-time traffic flow.

Unlike traditional traffic signal systems that rely on pre-defined and rigid timing schedules, this system incorporates a feedback-based strategy. It analyzes traffic density and average incoming speed for each lane and adjusts the green signal duration accordingly. The cyclic signal rotation algorithm ensures fairness by guaranteeing that every lane is served, while the density- and speed-based green time adjustment enhances traffic throughput and reduces unnecessary wait times at low-traffic lanes. This dual consideration of fairness and efficiency makes the solution both balanced and responsive to real-world traffic conditions.

The use of open-source tools such as YOLOv8, DeepSORT, OpenCV, and Streamlit ensures that the solution remains cost-effective and easily deployable across different urban settings without high infrastructural costs. Furthermore, the interface provides real-time visualization, exporting of analytical logs in CSV format, and clear feedback for system performance—all essential for operational transparency and administrative review.

Overall, this project presents a scalable and practical solution to the growing problem of urban traffic congestion.With further enhancements and real-world deployment, the proposed system can significantly improve traffic flow management, reduce delays, and contribute to smarter and more sustainable urban infrastructure.

Furthermore, the system's modular architecture and reliance on real-time data make it highly adaptable for future integration with smart city frameworks. As urban mobility continues to evolve, incorporating technologies like GPS-based vehicle tracking, V2I (Vehicle-to Infrastructure) communication, and cloud-based analytics can significantly enhance the performance and scalability of this solution. With minimal hardware requirements and the use of lightweight yet powerful AI models, this traffic control system not only meets the needs of current urban infrastructure but also paves the way for intelligent automation in city planning and traffic enforcement. As such, this project stands as a forward-thinking step toward the realization of intelligent, efficient, and equitable traffic management systems.

9 References

1. Ultralytics. (2024). YOLOv8: Next-Generation Real-Time Object Detection. [Online]. Available: https://docs.ultralytics.com

2. Bewley, A., (2016). Simple Online and Realtime Tracking with a Deep Association Metric (DeepSORT). [Online]. Available: https://github.com/nwojke/deep_sort

3. OpenCV. (2024). OpenCV Library Documentation. [Online]. Available: https://docs.opencv.org

4. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

5. NVIDIA Developer. (2023). CUDA Toolkit Documentation. [Online]. Available: https://developer.nvidia.com/cuda-toolkit

6. Pandas. (2024). Python Data Analysis Library – pandas. [Online]. Available: https://pandas.pydata.org

7. Transport Research Laboratory (TRL). (2021). Evaluation of Traffic Signal Timing Based on Vehicle Density. [Online]. Available: https://trl.co.uk

8. NumPy. (2024). NumPy: The fundamental package for scientific computing with Python. [Online]. Available: https://numpy.org

9. Ramezani, H., & Karim, M. R. (2022). Intelligent Traffic Light Control System Using Deep Learning. Journal of Traffic and Transportation Engineering.

10. JetsonHacks. (2023). Deploying YOLO Models on Edge Devices for Traffic Surveillance. [Online]. Available: https://www.jetsonhacks.com

10 Plagiarism Report

Deepanshu Singh Yadav

Final

project

project

IET Lucknow

Document Details

Submission ID

trn:oid:::1:3273040736

Submission Date

Jun 10, 2025, 1:40 PM GMT+5:30

Download Date

Jun 10, 2025, 1:44 PM GMT+5:30

File Name

Final_Project_Report.pdf

File Size

3.0 MB

40 Pages

8,904 Words

56,962 Characters

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

24 Not Cited or Quoted 2%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources
3% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **24** Not Cited or Quoted 2%
  Matches with neither in-text citation nor quotation marks
- **1** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources
3% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	
	Gongquan Zhang, Fengze Li, Dian Ren, Helai Huang, Zilong Zhou, Fangrong Chan...	**<1%**
2	Publication	
	S. Kannadhasan, R. Nagarajan, Alagar Karthick, V. Kumar Chinnaiyan. "Technolog...	**<1%**
3	Publication	
	D S Yadav, V S Yadav, R C Sarswat. "Maintaining Legal Requirements of Digital Sig...	**<1%**
4	Publication	
	"Fifth Congress on Intelligent Systems", Springer Science and Business Media LLC...	**<1%**
5	Publication	
	Tripti Mishra, N.B. Singh, Nandita Singh. "Restoration of red mud deposits by nat...	**<1%**
6	Publication	
	Sanjay Kumar Yadav, Gurmit Singh, Divakar Singh Yadav. "Mathematical Framew...	**<1%**
7	Publication	
	Ton Duc Thang University	**<1%**
8	Publication	
	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...	**<1%**
9	Publication	
	Kiranmaie P, Abhisatwika Reddy Chada, Ilaiah Kavati. "Wrong-Way Driving Detect...	**<1%**
10	Publication	
	Md. Shahriar Zaman Abid, Busrat Jahan, Abdullah Al Mamun, Md. Jakir Hossen, Sh...	**<1%**

11	Publication	
Eilif Hjelseth, Sujesh F. Sujan, Raimar J. Scherer. "ECPPM 2022 – eWork and eBusin...		<1%
12	Publication	
Hanoi Pedagogical University 2		<1%
13	Publication	
Hu Yang, Bao Guo, Changxin Yan, Zhiqiang Chen, Pu Wang. "A New Mobility Field ...		<1%
14	Publication	
James E. Moore, Ii, Stephen P. Mattingly, C. Arthur MacCarley, Michael G. McNally....		<1%
15	Publication	
Monika Mangla, Ashok Kumar, Vaishali Mehta, Megha Bhushan, Sachi Nandan M...		<1%
16	Publication	
S. Prasad Jones Christydass, Nurhayati Nurhayati, S. Kannadhasan. "Hybrid and A...		<1%