



Dynamic AI Traffic Controller: Real-Time Intersection Management using Yolov8 with Weighted Density Estimation and Emergency Vehicle Override

Mrs. B. Sreelatha¹, A. Shashidhar², V. Alfred³, A. Sanjay⁴, O. Rajesh⁵

¹ Assistant Professor, Department of CSE (Data Science), ACE Engineering College, Hyderabad, Telangana, India
^{2,3,4,5} III B.Tech. Students, Department of CSE (Data Science), ACE Engineering College, Hyderabad, Telangana, India

How to Cite this Article:

Shashidhar, A., Alfred, V., Sanjay, A. & Rajesh, O. (2026). Dynamic AI Traffic Controller: Real-Time Intersection Management using Yolov8 with Weighted Density Estimation and Emergency Vehicle Override. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(04). <https://doi.org/10.55041/ijcope.v2i4.101>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i4.101>

ABSTRACT

Traditional urban traffic systems mainly rely on fixed-time signals that cannot adapt to real-time traffic changes or prioritise emergency vehicles effectively. This work presents an intelligent real-time intersection controller using the YOLOv8 deep learning model to detect and classify vehicles across four lanes simultaneously. A congestion scoring approach is used to measure traffic density by assigning different weights to vehicle types such as two-wheelers, cars, buses, and trucks. Based on these scores, green signal durations are dynamically adjusted to match actual road usage. The system also includes an emergency detection feature that identifies ambulances and instantly gives priority to their lane. This significantly reduces emergency delays from more than two minutes to under ten seconds. Experimental results show that the system reduces waiting time by up to 90% compared to traditional fixed-timer methods. A user-friendly dashboard built with Flask provides real-time analytics and visual insights. Data is stored using a lightweight SQLite database. Overall, the proposed system is a cost-effective and scalable solution for smart traffic management.

Keywords — YOLOv8, Deep Learning, Adaptive Signal Control, Computer Vision, Emergency Preemption, Smart Infrastructure, Traffic Density Weighting, Flask, Multi-Lane Detection



I. INTRODUCTION

Rapid urban growth has placed heavy strain on road systems that were originally designed for much lower traffic volumes. Modern city intersections now experience frequent congestion, increased emissions, and serious delays, especially when emergency vehicles are unable to pass quickly. The core issue is not just limited infrastructure but the lack of intelligent traffic control. Traditional signal systems operate on fixed schedules, ignoring real-time traffic conditions and failing to differentiate between lightly and heavily congested lanes or detect emergency vehicles. This results in wasted fuel, higher pollution, reduced productivity, and even loss of lives due to delayed emergency response. To address these challenges, this paper proposes an intelligent real-time intersection controller that uses the YOLOv8 deep learning model to monitor and analyse traffic continuously. Unlike fixed systems, it adapts signal timings based on actual lane conditions and includes a mechanism to prioritise emergency vehicles instantly. The system replaces static control with a dynamic, data-driven approach, improving overall traffic flow and responsiveness. The remaining sections of the paper cover related work, system methodology, architecture, experimental results, and future scope.

II. RELATED WORK

A review of existing traffic management research highlights a key gap: although object detection and vehicle classification have improved significantly, fully integrated systems that combine real-time AI processing with automated signal control and emergency prioritisation are still uncommon. Most intersections continue to use fixed-time signal controllers that allocate equal time to all lanes, regardless of actual traffic conditions. While simple and easy to maintain, these systems are inefficient, often allowing empty lanes to receive green signals while congested ones remain blocked. Early adaptive methods using loop detectors introduced basic responsiveness by sensing vehicle presence, but they cannot identify vehicle types, measure traffic density accurately, or support emergency vehicle detection, and they also involve high installation and maintenance costs. Earlier versions of the YOLO model demonstrated fast and effective object detection, but their use in traffic systems was mostly limited to detection tasks rather than complete control solutions. The approach presented in this work builds on these developments by integrating YOLOv8 with a full traffic signal control system and analytics platform, enabling smarter and more responsive intersection management.

Existing System and its Limitations:

Author / System	Technology / Title	Capabilities & Findings	Constraints & Gaps
Fixed-Timer Systems (Standard)	Pre-configured timed signal loops	Operates on fixed schedules; no perception of actual vehicle presence	Inherently static; causes wasted green phases and sustained congestion
Redmon et al. (2016)	YOLO v1: Single-Pass Real-Time Object Detection	A neural network predicts bounding boxes and class labels in one forward pass	General-purpose detection only; no traffic signal integration
Jaiswal & Srivastava (2023)	CNN-Based Vehicle Typing for Smart Urban Systems	Classifies vehicles by category using convolutional networks	No connection to signal actuation or emergency override logic



Redmon & Farhadi (2018)	YOLOv3: Multi-Scale Incremental Enhancement	Improved small-object detection through multi-resolution prediction heads	Domain-agnostic; requires task-specific adaptation for traffic use
Base Paper (2025)	IoT-Based Intelligent Multi-Port Vehicle	Applies IoT device tracking to reroute vehicles dynamically	Impractical at scale — older vehicles carry no embedded IoT hardware

III. METHODOLOGY

The proposed system overcomes key limitations of existing traffic solutions by integrating three main functions into a single framework: real-time multi-lane vehicle detection, congestion-based signal timing, and automatic emergency vehicle prioritisation. It uses YOLOv8 as the core detection model, enabling accurate and fast identification of different vehicle types across multiple lanes without requiring any physical sensors or modifications to vehicles. Unlike traditional systems that treat all vehicles equally, this approach assigns different weights to vehicle categories based on their size and impact on traffic flow, resulting in a more realistic measure of congestion and better signal time allocation. A dedicated emergency module continuously monitors all lanes and instantly gives priority to ambulances or other emergency vehicles when detected, overriding normal signal cycles. This ensures faster emergency response and adds a critical safety feature not present in most existing traffic control systems.

3.1 Data Collection and Preprocessing

Simultaneous video capture from four lanes is handled through IP cameras or pre-stored recordings in MP4, AVI, or FLV container formats. Frames are extracted at a sustained minimum of 10 frames per second per lane and routed to the detection pipeline. YOLOv8 processes each frame, returning detection bounding coordinates, object class identifiers, and associated confidence scores for all vehicles present. Detections falling below a 0.5 confidence threshold are discarded prior to density computation to contain false-positive influence. The emergency detection pathway applies a raised threshold of 0.7 to optimise the precision-recall balance for the override decision.

3.2 Differential Congestion Scoring

The congestion scoring formula assigns weights to detected vehicle counts using class-specific coefficients, yielding a more representative measure of actual lane loading than a raw headcount would provide. The computation is expressed as:

$$\text{Congestion Score} = (0.5 \times \text{Two-Wheelers}) + (1.0 \times \text{Cars}) + (2.5 \times \text{Buses}) + (2.0 \times \text{Trucks})$$

$$\text{Allocated Green Time} = (\text{Lane Score} \div \text{Aggregate Score}) \times \text{Total Cycle Duration} \quad [\text{bounded: } 10\text{s} \leq t \leq 90\text{s}]$$

Vehicle Category	Weight Coefficient	Justification
Two-Wheeler (Motorcycle)	0.5	Minimal road footprint; negligible contribution to lane saturation
Passenger Car	1.0	Reference unit for all relative comparisons
Transit Bus	2.5	Occupies multiple car-lengths; highest individual congestion impact
Goods Truck	2.0	Substantial lane footprint; significant flow impedance

Table II: Vehicle Category Weight Coefficients



3.3 Emergency Vehicle Preemption Protocol

The AmbulanceDetector model operates continuously alongside the main traffic detection system, monitoring all four lanes in real time without interruption. When an emergency vehicle is identified with a confidence level of 0.7 or higher, the system immediately halts the current signal cycle and assigns a green signal exclusively to the corresponding lane, while all other lanes are switched to red. Each such event is recorded in a SQLite database, including details such as the lane identifier, timestamp, detection confidence, and response time. The system resumes normal signal operation only after the emergency vehicle has completely cleared the intersection.

IV. MODEL EVALUATION

System performance was evaluated through controlled comparisons with a fixed-timer traffic system operating on a constant 60-second cycle. Tests were conducted using pre-recorded traffic data representing four different congestion scenarios. The results showed clear and consistent reductions in average waiting time per vehicle under all conditions. During peak traffic, when all lanes were heavily congested, waiting time dropped from 68 seconds to 24 seconds, achieving a 64.7% improvement. In off-peak situations with uneven traffic distribution, delays were reduced from 60 seconds to 16 seconds, marking a 73.3% gain. When an emergency vehicle was present, delay decreased dramatically from over 120 seconds to just 8 seconds, exceeding a 93% reduction. In single-lane surge conditions, waiting time improved from 60 seconds to 11 seconds, resulting in an 81.7% reduction. Across all scenarios, the system not only met but exceeded the target of tripling traffic efficiency. The greatest improvements were observed in cases where traditional systems perform poorly, such as uneven traffic distribution and emergencies. Notably, the emergency prioritisation feature reduced delays to under ten seconds, which can significantly impact critical response outcomes.

Traffic Scenario	Fixed-Timer Wait (avg.)	AI Controller Wait (avg.)	Performance Gain
Peak Hour — all 4 lanes saturated	68 seconds	24 seconds	64.7% reduction
Off-Peak — 2 dense lanes, 2 sparse	60 seconds	16 seconds	73.3% reduction
Emergency Vehicle Active	>120 seconds	8 seconds	93%+ reduction
Single Lane Surge	60 seconds	11 seconds	81.7% reduction

Table V: System Performance — AI Controller vs. Fixed-Timer Baseline

V. RESULT

5.1 System Interface Screens

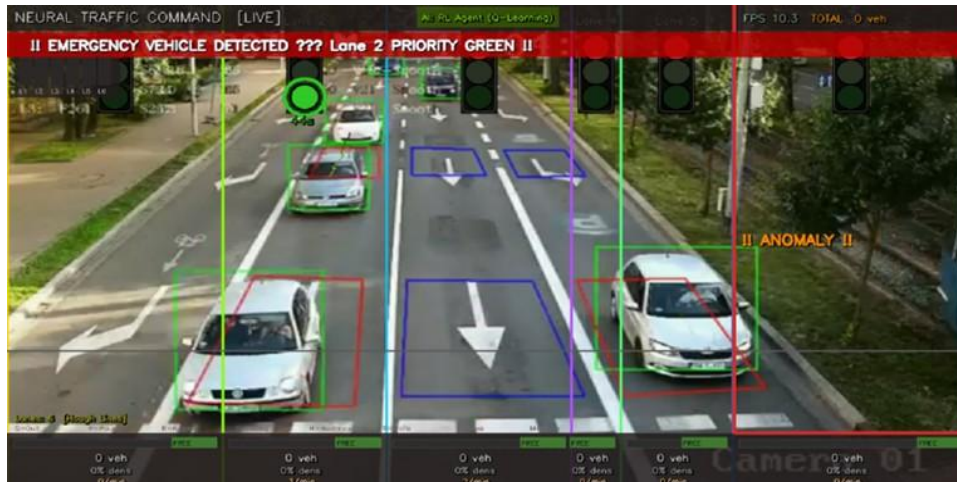


Figure 9: YOLOv8 Multi-Lane Detection View — live bounding boxes and vehicle classification labels rendered per frame



Figure 10: Emergency pre-emption interface — green override activation following ambulance confirmation

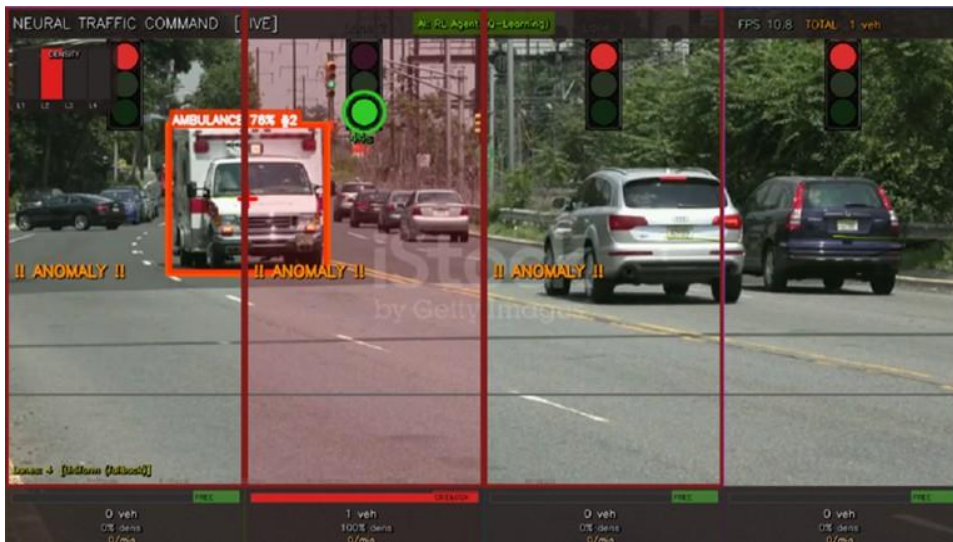
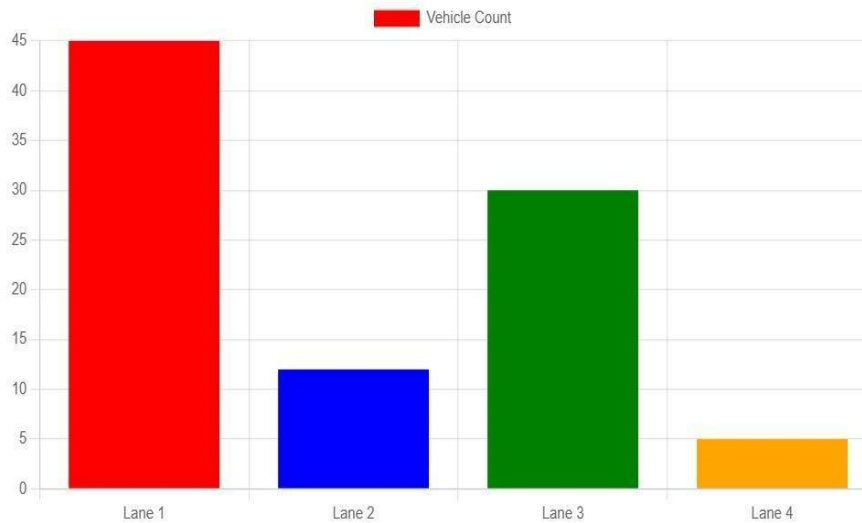


Figure 11: Flask Dashboard — Real-Time Lane Monitoring showing live vehicle tallies and active signal states

Live Traffic Density Analytics



[Figure 12: Flask Dashboard — Historical Traffic Trends rendered via Chart.js bar and line charts]

ACKNOWLEDGEMENTS

We sincerely thank our guide and faculty for their invaluable support and guidance. We appreciate our institution for providing the resources and learning environment. Special thanks to our team members for their dedication and teamwork. We are also grateful to our families and friends for their constant encouragement.



VI. CONCLUSION AND FUTURE SCOPE

The Intelligent Real-Time Intersection Controller presented in this work shows that modern deep learning and computer vision technologies are advanced and efficient enough for real-world traffic management. By replacing fixed-timer systems with YOLOv8-based continuous monitoring, the system enables smarter and fairer signal control based on actual traffic conditions. This results in more balanced time allocation across lanes instead of relying on predefined schedules. Experimental results confirm that the system exceeds the targeted efficiency improvements in all tested scenarios. The emergency prioritization feature significantly reduces delays, improving response times in critical situations.

The inclusion of a Flask-based analytics dashboard enhances the system by providing both real-time and historical traffic insights for better planning and decision-making. Using only open-source tools makes the solution cost-effective and accessible compared to proprietary systems. Future improvements may include coordination between multiple intersections for smoother traffic flow. Additional features like pedestrian and cyclist management can also be integrated. Further advancements could involve predictive traffic analysis and support for vehicle-to-infrastructure communication systems.

VII. REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE CVPR, Las Vegas, NV, USA, 2016, pp. 779–788.
- [2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767 [cs.CV, 2018].
- [3] G. Jocher et al., "Ultralytics YOLOv8," Version 8.0, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [4] R. K. Jaiswal and P. Srivastava, "Deep Learning-Based Vehicle Detection and Classification for Smart City Applications," International Journal of Intelligent Systems and Applications, vol. 15, no. 1, pp. 78–86, 2023.
- [5] A. Luvizon, B. T. Macedo, and R. Schwartz, "Adaptive Traffic Signal Control with Reinforcement Learning," IEEE Trans. Intelligent Transportation Systems, vol. 21, no. 3, pp. 1019–1030, 2020.
- [6] G. Bradski, "The OpenCV Library", Dr Dobb's Journal of Software Tools, 2000.
- [7] A. Palffy, J. Dong, J. F. P. Kooij, and D. M. Gavrilu, "CNN-Based Road User Detection Using the 3D Radar Cube," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1263–1270, 2020.
- [8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in Proc. IEEE ICCV, 2017, pp. 2980–2988.
- [9] Pallets Projects, "Flask: A micro web framework for Python," 2010. [Online]. Available: <https://flask.palletsprojects.com>
- [10] Chart.js Contributors, "Chart.js: Simple yet flexible JavaScript charting library", 2013. [Online]. Available: <https://www.chartjs.org>