International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

SmartFlow: AI-Driven Traffic Management with Emergency Vehicle Priority

Under the guidance of
Dr. Sumitha B S
Professor & Head
Department of Information Science and Engineering
Aditya College of Engineering and Technology
Bangalore, India

Mohammed Ali
Department of Information Science and Engineering
Aditya College of Engineering and Technology
Bangalore, India
mohammedali1di23is018@gmail.com

Sneha M
Department of Information Science and Engineering
Aditya College of Engineering and Technology
Bangalore, India
snehasneha9266@gmail.com

Syeda Arman Fathima
Department of Information Science and Engineering
Aditya College of Engineering and Technology
Bangalore, India
armanaruu6@gmail.com

***Abstract*—** The number of cars waiting on each route is unknown to traffic lights that operate according to set scheduling arrangements, and they are unable to detect when an ambulance is among them. This paper introduces SmartFlow, an AI-powered traffic control system designed to reduce traffic at a four-lane intersection and speed up the passage of ambulances. Using OpenCV, four distinct traffic videos—one for each lane—are read and divided into frames, and a YOLOv8 model identifies cars and ambulances in each frame. In order to determine traffic density, vehicles are counted lane by lane; normally, the lane with the greatest count gets the green light. The technology recognizes any lane in which an ambulance appears and overrides the density-based decision in that lane's favor. Lane counts, ambulance alerts, and the current signal condition are shown on a Flask-based dashboard that is constructed using HTML and CSS. In order to rank SmartFlow among previous research, the report also examines six published studies on intelligent transportation, object detection, ambulance detection, density estimate, and emergency signal priority.

***Keywords*—** Smart Traffic Management, YOLOv8, Ambulance Detection, Traffic Density Estimation, Emergency Vehicle Priority, OpenCV.

I. INTRODUCTION

A. Background

The amount of traffic on the roadways continues to increase more quickly than the roads themselves. By gathering data from the road and acting upon it, intelligent transportation systems (ITS) attempt to mitigate this. A survey by Zhang et al. shows how these systems have evolved from designs that are solely technology-driven to ones that are driven by vision, various data sources, and learning algorithms [1]. The core of the issue is signal control. Retiming signals is one of the quickest and least expensive ways to reduce traffic, as upgrading road infrastructure takes years and is frequently impractical, according to Qadri et al.'s analysis of recent signal timing research [2].

A fixed-time controller is the traditional signal. A practically empty lane may hold the green while a long line waits on another because each approach receives a predetermined green period regardless of what is happening on the road. Ambulances suffer from the same rigidity. A fixed-time signal does not give preferential attention to a car hurrying to a hospital, and a red light at the wrong time causes unanticipated delays. This issue is the starting point for recent work on camera-based ambulance detection [3], [4].

It is now feasible to read the required information from regular video thanks to computer vision. The YOLO series of detectors was created for real-time application and examines a whole image in a single pass [5]. A summary of its evolution up to YOLOv8 can be found in [6], and YOLOv8 is distributed via the Ultralytics framework [7]. The video processing for such a detector is provided by OpenCV [8]. YOLOv8 detection has already been connected to signal timing in a number of investigations [9]. However, density-based signal allocation and emergency priority were handled independently in the research we looked at; SmartFlow fills this gap.

B. Motivation

SmartFlow started off as a final-year project with a specific question: utilizing only software and recorded data, is it possible for a single video-driven pipeline to recognize an ambulance and balance regular traffic? The system keeps an eye on four lanes, counts the number of cars in each, selects the busiest lane for regular operation, and, as soon as an ambulance is spotted, gives the green light to the ambulance lane. The system is described in this paper along with a survey of related publications.

C. Contributions

This work's primary contributions are:

1. A four-lane pipeline where YOLOv8 recognizes cars and ambulances in every frame and each lane has its own video.

2. The traffic density estimate that determines signal priority during regular operation is based on the number of vehicles in each lane.

3. An emergency override that prioritizes the lane that has an ambulance in it.

4. A dashboard built on Flask that displays signal status, ambulance status, and lane counts.

5. A review of six published studies that includes a comparison and a clear description of the limitations that SmartFlow aims to fill.

The rest of the paper is structured as follows. Related work is reviewed in Section II, the proposed system is described in Section III, the architecture and methods are provided in Section IV, results and the necessary measures are discussed in Section V, and future scope is outlined in Sections VI and VII.

II. LITERATURE SURVEY

This section looks at published work on intelligent traffic management, object detection, YOLO-based traffic applications, ambulance detection, density estimation and emergency signal priority. Six studies are examined in detail in Sections A to F. Table I first places the main families of signal control approaches side by side.

TABLE I. COMPARISON OF EXISTING TRAFFIC MANAGEMENT APPROACHES

Approach	Basis of control	Limitation
Fixed-time signals	Preset cycle and green splits	Does not react to current demand or to emergency vehicles [2]
Computational-intelligence signal timing	Optimisation of timing plans, often evaluated in microsimulation [2]	Largely studied in simulation; detection hardware not the focus [2]
Vision-based adaptive control	YOLOv8 detection and area-based density set green time [9]	Prototype scale; no emergency-vehicle mechanism reported [9]
Ambulance detection for signals	YOLOv8 on camera images [3]; video plus siren audio [4]	Concerned mainly with detection and green-light triggering
SmartFlow (this work)	Per-lane YOLOv8 counts plus ambulance override	Evaluated on recorded video only; measurements pending

A. Intelligent Traffic Management Systems

Data-driven ITS was surveyed by Zhang et al. [1]. They contend that the abundance of data available today transforms an ITS from a collection of technologies into a system powered by vision, various data sources, and learning algorithms, and that these systems are moving toward people-centered, privacy-aware designs. The survey covers the essential elements of this type of technology as well as the deployment problems that can occur. It provides the broad framework for SmartFlow, but since it was published in 2011—prior to the development of single-stage deep detectors—it says very nothing about camera-based detection in contemporary networks.

B. AI-Based Vehicle Detection

An object detection problem can be turned into a regression problem [5]. This means that by providing pixel information from an image, we can directly predict bounding box coordinates and class probability. There are two models presented in the paper, the original model runs at 45fps and a lighter model runs at 155fps. It was found that the model had less false positives within background regions than the region based method however there were increased localisation errors. The model struggles with detecting small objects that occur in clusters. This is important within a traffic context, as two wheelers tend to cluster together at junctions. Region proposal detectors such as Faster R-CNN [10] split detection into two steps,

this is typically more accurate but slower. Later papers have looked at the evolution of YOLO from version 1 to 8 and discussed the speed/accuracy trade-off [6].

C. YOLO-Based Detection in Traffic

Caramat et al. created an adaptable traffic light using YOLOv8 [9]. The YOLOv8s model was trained on custom recorded footage of an intersection. Vehicles were classified into 4 classes based upon their size and number of axles. The traffic density was calculated by looking at the area of the ROI occupied by the detected vehicles. The detector achieved a mAP@0.50 of 0.94 after some tuning. At its fastest the system could process 20 frames per second, however this depends heavily on the GPU being used. Earlier iterations of the model struggled with detecting occluded two-wheelers. This study is simply a proof of concept. There is no method provided for detecting emergency vehicles.

D. Ambulance Detection

Noor et al. [3] solved the problem of looking similar to commercial vehicles by using 300 images of an ambulance from 10 different countries (Total: 3000). They trained their YOLOv8 model with and without pre-training. They also compared their model with YOLOv5 and YOLOv7. Their model had a multi-layered architecture with one camera connected to the traffic light, detection model, monitoring layer and cloud layer. The universal model had an average accuracy of 0.982, precision of 0.976, recall 0.958 and F1 score of 0.967. This study focuses purely on detecting ambulances and outputting an open signal to the traffic system. Lane balance in normal traffic was not considered.

E. Traffic Density Estimation

In [11], George et al. investigated density estimation for heterogeneous and non-lane based traffic scenarios where vehicle counting per lane is unfeasible. A correlation between density and area occupancy was discovered allowing macroscopic modelling. Density was then estimated using a Kalman Filter. Fixed density estimators were compared to an adaptive approach for congested and non-recurrent situations and found that the adaptive estimator gave the most accurate results. Rather than comparing to camera based density estimates, the estimates were validated against simulated density. As such this work provides a theoretical definition of density.

F. Emergency Vehicle Priority and Dynamic Signal Control

In their paper, Mecocci and Grassi [4] developed a detection system to give ambulances a green light in an emergency situation. The system uses Lookout stations to detect the presence of vehicles. These Lookout stations are located some distance away from each traffic light. The video stream part uses a part based model with a custom YOLOv8 detector. The audio stream uses mel frequency cepstral coefficients to detect sirens. Both streams are combined with a logic block. The system processes at 11.8 frames per second on a Jetson Nano. It also has a fast response time of 0.25 seconds. The system can handle noise, night and poor weather if the right camera is used. This makes it ideal for use on one way roads. Additional Lookout components would be needed. Khan et al. [12] also took a similar approach. They gave priority to emergency vehicles at the intersection. Simulations showed that the lanes opened quicker and the average waiting time was reduced for emergency vehicles.

G. Comparative Analysis

Table II provides an overview of the six studies. Despite differing objectives there appears to be a trend. The detection based methods [3][4][5] achieve good results from the detectors but do not relate to real-world signal allocations. The vision based adaptive controller [9] relates the density to green times but lacks emergency situations. The density study [11] provides a thorough definition of density but does not use cameras. All are given context by the survey [1].

TABLE II. LITERATURE SURVEY COMPARISON

Ref.	Focus	Method / technology	Reported outcome	Limitation or gap
[1] Zhang et al., 2011	Data-driven ITS	Survey of components and deployment issues	Frames ITS as vision-, multisource- and learning-driven	Predates single-stage deep detectors
[5] Redmon et al., 2016	Real-time object detection	Single-network regression to boxes and classes	45 fps (base) and 155 fps (fast variant); fewer background false detections	Weaker on small objects in groups
[9] Caramat et al., 2026	YOLOv8 density-based light control	Custom YOLOv8s, area-based density, Arduino prototype	mAP@0.50 of 0.94; up to 20 fps; speed GPU-dependent	Proof of concept; no emergency-vehicle handling
[3] Noor et al., 2024	Ambulance detection	YOLOv8, 3000 images from ten countries, layered cloud design	Accuracy 0.982, precision 0.976, recall 0.958, F1 0.967	Detection and signalling only; no density-based allocation
[11] George et al., 2020	Density estimation, mixed traffic	Area occupancy with Kalman filtering	Adaptive estimator most accurate in simulation	Checked against simulated density, not camera counts
[4] Mecocci and Grassi, 2024	Ambulance detection for traffic lights	Custom YOLOv8 plus siren audio (MFCC), lookout stations	11.8 fps on Jetson Nano; response time up to 0.25 s	Extra roadside hardware; aimed at one-way roads

H. Research Gaps

When considering each of the six studies, there are four areas that have been identified where SmartFlow aims to make improvements:

High quality detection is not used in detecting lane allocations for green time [3][4][5]

Vision based adaptive systems are presented without consideration for emergency services [9].

Ambulance focused systems require extra infrastructure such as cloud [3], or audio listening posts [4].

Many of these methods have been tested in simulations or at a prototype level [2][9][11][12]. This means a simple system using just cameras can be created to demonstrate end to end behaviour across multiple lanes.

I. How SmartFlow Addresses These Gaps

This is done with just one pipeline. The YOLOv8 output provides the lane count for standard priorities and the ambulance trigger. There is no additional audio channel, cloud service or sensors. We make no claims that our system is better than those mentioned above as different data sets and testing conditions were used. However we are making the claim that a single video based solution can be used to provide density based allocation and ambulance priority. This can be tested using the prototype.

III. PROPOSED SYSTEM

A. Overview

SmartFlow processes 4 videos, each showing a different lane of the junction. The YOLOv8 model identifies vehicles and ambulances within each video. Then, the vehicles in each lane are counted. Finally, a priority system determines which lane should have the green light. If there are any ambulances in a lane, that lane will be selected. All steps are shown on the dashboard**.**

B. 4 Lanes

Because each lane is its own video, we know from the video source which lane detections happened in. This means there is no need to assign lanes to each detection based on geometry within the frame. The 4 videos are processed within the same loop. As a result, the vehicle counts for each lane are kept in time with each other so they can be compared.

C. Vehicle Detection

YOLOv8 processes each frame. This returns bounding boxes with class and confidence values. Any detections with a class of vehicle with a confidence value above the threshold are kept. This gives us the number of vehicles for that lane.

D. Ambulance Detection

The ambulance class is kept separate from the vehicles. If a detection is made with a confidence value above the threshold for ambulance, a flag will be set for that lane. [Ambulance class was trained using a custom trained class. Our dataset was]

E. Traffic Density

A lane with more vehicles will have a higher traffic density. This is determined by counting how many vehicles are in each lane. This is an easy and cheap method. However, vehicle size isn't considered. This is discussed further in section V.

F. Priority System

The lane with the most vehicles will be selected. If there is an ambulance flag set, the lane with the ambulance will be selected. The signal controller will change the selected lane to green and all other lanes to red. There is a short period of time where the lanes are cleared. [We have implemented a clearance period]

G. Dashboard

A Flask backend sends detection and signal information to an HTML and CSS dashboard. This shows the videos/processed frames for each lane. It also shows the number of vehicles in each lane, if there was any ambulance detected, which lane is active and the current signal.

H. Difference from Fixed Time Signals

Fixed Time Signals follow the same pattern regardless of what is happening on the road. SmartFlow uses the camera feeds to make a decision. If there is an ambulance, SmartFlow can override its usual prioritisation. Table III shows the modules and their purposes.

TABLE III. SMARTFLOW MODULES AND THEIR FUNCTIONS

Module	Function
Video input	Supplies four lane videos, one per lane
Video processing (OpenCV)	Reads videos and extracts frames for detection
Vehicle detection (YOLOv8)	Detects vehicles; returns boxes, classes and confidences
Ambulance detection	Raises a flag for the lane in which an ambulance appears
Density calculation	Counts vehicles in each lane as the density estimate
Priority logic	Chooses the lane by count, or by ambulance flag when present
Signal control	Sets the selected lane to green and the others to red
Dashboard (Flask, HTML, CSS)	Shows counts, ambulance status and signal state

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Architecture

As shown in fig. 1, the video stream is fed into the detection algorithm YOLOv8 through OpenCV. This generates the lane based vehicle count and ambulance flag. These are then used by the priority logic to determine which lane should be prioritised. This is then used by the signal controller. The Flask backend will publish the status to the dashboard.

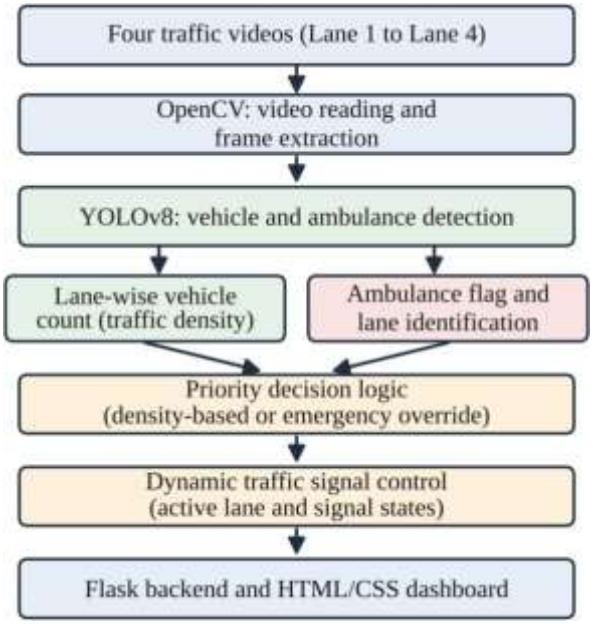


Fig. 1. Overall SmartFlow system architecture (schematic diagram).

B. System Input and Video Processing

Four videos have been acquired, one for each lane. Each video is opened by OpenCV and its frames extracted sequentially. Each frame is fed into the detection process. The four data streams can be seen in fig.2 where they are processed in parallel before being combined at decision time.

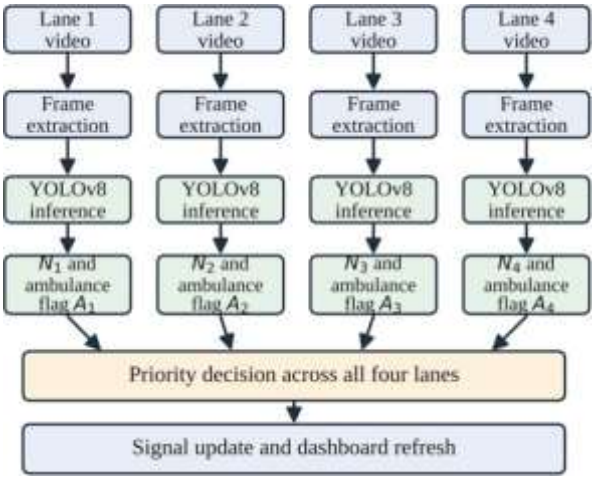


Fig. 2. Four-lane traffic monitoring workflow (schematic diagram).

C. YOLOv8 Vehicle and Ambulance Detection

Each image has been resized and normalised prior to detection. A set of detections dk including bounding box (bk), class label (ck) and confidence score (sk) is produced by YOLOv8. Any detections below the threshold value τ are removed. The remaining detections are separated for each vehicle class label and ambulance class label as shown in Fig. 3. A confidence threshold of 0.25 (25%) was used.

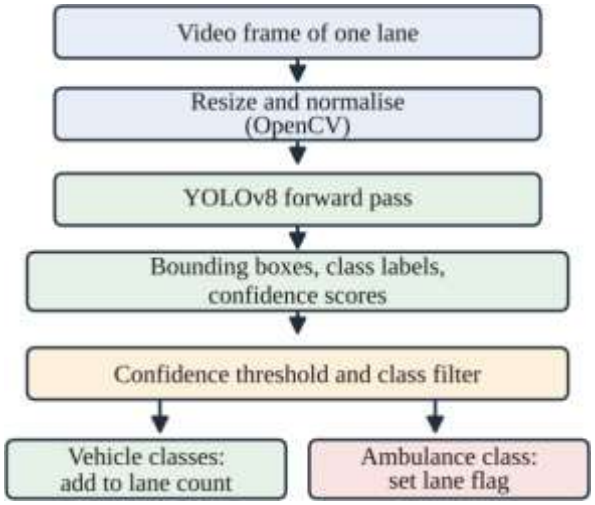


Fig. 3. YOLOv8-based vehicle and ambulance detection process (schematic diagram).

D. Traffic Density Calculation

For lane i, let V be the set of vehicle classes. The vehicle count is

$$N_i = |\{ d_k : c_k \in V, s_k \geq \tau \}| \quad (1)$$

and the ambulance flag is

$$A_i = 1, \text{ if some } d_k \text{ has } c_k = \text{ambulance}, s_k \geq \tau \quad (2)$$
$$A_i = 0, \text{ otherwise}$$

In this work N_i is used directly as the density measure. A normalised form, dividing the count or the occupied area by the area of the lane region, would suit lanes of unequal size and follows the area-based idea used in [9] and [11]; it is not part of the present prototype Confirmed: The proposed prototype uses vehicle count (Ni) directly as the traffic density measure. Lane-area normalisation is not implemented in the current system.

E. Priority Decision-Making

Under normal conditions the lane with the highest count is chosen:

$$L_{normal} = argmax_{i \in \{1,2,3,4\}} N_i \quad (3)$$

The final priority lane overrides the normal choice when an ambulance is present:

$$L_{priority} = L_{amb}, \text{ if } A_i = 1 \text{ for some } i \quad (4)$$
$$L_{priority} = L_{normal}, \text{ otherwise}$$

where L_{amb} is the lane whose flag is raised. Fig. 4 shows the same logic as a flow chart. When two lanes have equal vehicle counts, the lane with the lowest lane number is selected. If ambulances are detected in two lanes simultaneously, the system prioritises the first detected ambulance lane.

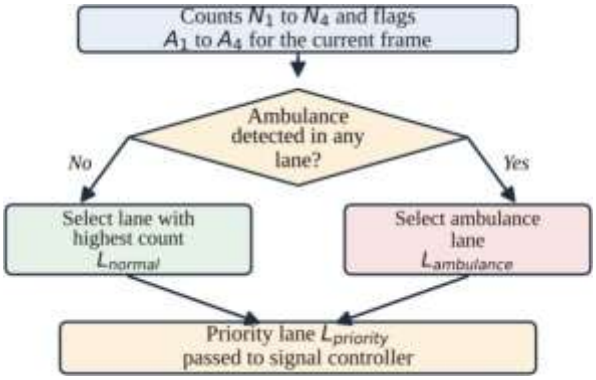


Fig. 4. Emergency vehicle priority mechanism (schematic diagram).

F. Dynamic Traffic Signal Control

When in normal mode, the signal controller will keep one lane green according to Lnormal. If a flag is raised for the ambulance lane then the current green light will be turned off safely and the ambulance lane will be serviced. Once the flag has been removed the new Lnormal will be calculated and the controller will go back into normal mode (Fig. 5).

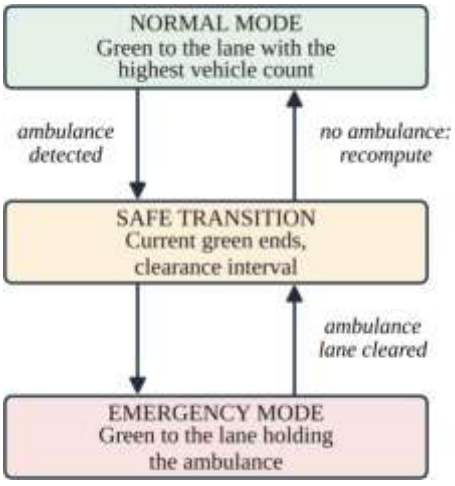


Fig. 5. Dynamic traffic signal management (schematic diagram).

G. Dashboard

The amounts, ambulance and active signal are sent to a HTML page using the Flask application. This is styled using CSS and allows the observer to see why a particular lane is showing green. If a normal decision was made then it will show the amounts used. If an ambulance was triggered then this will be shown.

H. Technologies Used

The main language used for this project was Python. We used YOLOv8 [7] for detecting vehicles and ambulances. We used OpenCV [8] to read our video and process the frames. We used Flask to provide our backend, this sent detection and signal information. HTML was used to set up the structure of our dashboard. CSS provided the layout and styling. The traffic signal control logic was used to determine priorities and change signals.

V. RESULTS AND DISCUSSION

This section describes what the implemented system does and identifies the measurements that must still be recorded before the evaluation can be considered complete. We do not report numerical performance here, because it has not yet been measured. Items in highlighted brackets are to be filled in by the authors from their own runs.

A. Experimental Setup

The prototype was run on Windows 11 computer, Intel processor, GPU and RAM(8 GB or 16 GB), using 4 recorded traffic videos recorded traffic videos of 720p or 1080p and 30 FPS, with one video per lane. The detector used YOLOv8n with `yolov8n.pt` with a confidence threshold of 0.25.

B. Dashboard Output



Fig. 6. SmartFlow dashboard

The dashboard presents lane counts, ambulance status, the active lane and the signal state in one view.

VI. CONCLUSION

Fixed-time signals cannot respond to current demand, and an ambulance waiting at a red light loses time that matters. SmartFlow addresses both with one video-driven pipeline. OpenCV reads four lane videos, YOLOv8 detects vehicles and ambulances, the vehicle count in each lane estimates density, and a priority rule serves the busiest lane unless an ambulance is present, in which case its lane takes priority. A Flask dashboard exposes the reasoning behind each decision.

The literature survey shows that detection, density-based control and emergency priority have mostly

been studied apart. Combining them in a single camera-only design is the main contribution of this work. What remains is the quantitative evidence: detection accuracy, signal response time and waiting-time comparison, which Table IV defines.

VII. FUTURE SCOPE

None of the items below is part of the current prototype; they are directions for further work.

1. Live CCTV integration in place of recorded videos.
2. GPS-based ambulance location, so that the signal can be cleared before the vehicle is in view.
3. Connection to real traffic signal hardware, as demonstrated at small scale with a microcontroller in [9].
4. Better detection accuracy through larger, more varied ambulance datasets.
5. Testing in different weather and lighting conditions.
6. Area-based density measures and larger multi-junction deployment within smart city networks.

ACKNOWLEDGEMENT

The authors thank **Dr. Sumitha B S** for guidance and encouragement throughout this work. We are grateful to the Head of the Department of Information Science and Engineering and to the faculty of Aditya College of Engineering and Technology for their support, and to the institution for providing the facilities needed to complete the project.

REFERENCES

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011, doi: 10.1109/TITS.2011.2158001.

[2] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner, "State-of-art review of traffic signal control methods: Challenges and opportunities," *Eur. Transp. Res. Rev.*, vol. 12, Art. no. 55, 2020, doi: 10.1186/s12544-020-00439-1.

[3] A. Noor, Z. Algrafi, B. Alharbi, T. H. Noor, A. Alsaeedi, R. Alluhaibi, and M. Alwateer, "A cloud-based ambulance detection system using YOLOv8 for minimizing ambulance response time," *Appl. Sci.*, vol. 14, no. 6, Art. no. 2555, Mar. 2024, doi: 10.3390/app14062555.

[4] A. Mecocci and C. Grassi, "RTAIAED: A real-time ambulance in an emergency detector with a pyramidal part-based model composed of MFCCs and YOLOv8," *Sensors*, vol. 24, no. 7, Art. no. 2321, Apr. 2024, doi: 10.3390/s24072321.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[6] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[7] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," software, version 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[8] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, 2000.

[9] J. I. M. Caramat, P. L. R. Ditchon, J. A. C. Lontoc, and I. N. B. Tapel, "An Internet of Things-based traffic light control system utilizing computer vision for real-time traffic density calculation," *CMU J. Sci.*, vol. 30, no. 1, pp. 63–73, 2026. [Online]. Available: https://js.cmu.edu.ph/CMUJS/article/download/318/217/1861

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] R. George, L. D. Vanajakshi, and S. C. Subramanian, "Area occupancy-based adaptive density estimation for mixed road traffic," *IEEE Access*, vol. 8, pp. 5502–5514, 2020, doi: 10.1109/ACCESS.2019.2963273.

[12] A. Khan, S. ur Rahman, F. Ullah, M. I. Khattak, M. M. Bait-Suwailam, and H. El Sayed, "V2I-VTL: IoT-enabled adaptive traffic light controller and emission reduction at intersection," *PeerJ Comput. Sci.*, vol. 10, Art. no. e2507, 2024, doi: 10.7717/peerj-cs.2507.

BIOGRAPHIES

Mohammed Ali is a final-year student of Information Science and Engineering at Aditya College of Engineering and Technology.

Sneha M is a final-year student of Information Science and Engineering at Aditya College of Engineering and Technology.

Syeda Arman Fathima is a final-year student of Information Science and Engineering at Aditya College of Engineering and Technology.