

SmartFlow: Intelligent Traffic Signal Management with YOLO And ViT

Mr. Adithya S Kundar¹, Ms. Ambika S Kundar², Mr. Ankith G Kodiya³, Mr. B G Rohith⁴, Prof. Shruthi D⁵
^{1,2,3,4,5}*Department of Computer Science and Engineering, Shree Devi Institute of Technology*

Abstract—Increased traffic, congestion, and traffic signal problems affects fuel consumption and raise emission rates. There are also constraints like emergency vehicle priority, accident management, and red-light violations. This study proposed Automated Traffic System (ATS) using YOLOv8 and computer vision to manage a four-way road traffic system and developed the prototype and web dashboard. The system aims to automatically adjust signal light changes based on the number of vehicles waiting at an intersection. The framework utilizes YOLOv8 to detect and count vehicles and recognize emergency vehicles such as ambulances and police cars and employs a Vision Transformer (ViT) model to classify accidents. Moreover, the OpenCV library was used to extract key frames from the uploaded videos. The framework counts the number of cars in each frame, and the traffic controller decides which junction will get the green light based on this information within a specified range. The system prioritizes pedestrians, emergency vehicles, and accident scenes over regular traffic. The model also provides a 30 seconds green corridor for emergency vehicles, and when an accident is identified, it initiates an accident management state. This framework provides a REST API backend built using the Flask framework and a web dashboard built using HTML, CSS, and JavaScript to view signals status, vehicle counting, alerts, timers, and statistics. Therefore, the proposed ATS framework can be considered as a prototype for managing a four-way traffic system by using an artificial intelligence approach.

Index Terms—Automated Traffic System, Intelligent Transportation System, YOLOv8, Computer Vision, Vision Transformer, Accident Detection, Emergency Vehicle Detection, Adaptive Traffic Signal Control, Green Corridor, Traffic Management, Flask, Real-Time Traffic Monitoring.

I. INTRODUCTION

Due to rapid urbanization, the growing population, and the increasing number of vehicles, traffic management has become a critical problem in urban areas.

Intersections in metropolitan cities often suffer from congestion, traffic delays, inefficient utilization of traffic signals, and difficulties in providing timely passage to emergency vehicles. Traditional traffic signals generally operate according to a predetermined schedule, providing a particular or predetermined green period to each road without continuously considering the current traffic situation. These approaches may not respond to dynamically changing traffic conditions. Recent research has therefore focused on intelligent traffic-management systems that can monitor traffic conditions and adapt signal decisions according to real-time requirements [1], [2], [6].

Modern developments in computer vision and artificial intelligence provide useful approaches for intelligent traffic monitoring. Camera systems can capture road conditions, while deep-learning-based object detection methods can identify and count vehicles from images and video streams [1], [2], [7]. You Only Look Once (YOLO) is one of the widely used object detection approaches for real-time vehicle monitoring. YOLO-based systems have been applied to detect different vehicle categories and estimate traffic conditions from road scenes [2], [5], [10]. Vehicle detection and counting can provide lane-wise traffic information that can be used by an adaptive traffic controller to modify signal timing according to traffic demand [2], [6].

Accidents are another major concern for traffic management. Accident scenes create hazardous conditions and disrupt the regular traffic flow, prompting traffic controllers to respond accordingly. When implemented in a computer vision system, deep learning-based accident detection can automatically identify such unsafe traffic conditions from images or video content. The proposed system utilizes a Vision Transformer (ViT)-based solution for accident

classification, which means that the scene in any given traffic image or video frame can be categorized as “accident” or “no accident.” For the video processing part, the OpenCV library is used to extract relevant frames from the input videos, which are then processed by the AI models.

Emergency vehicle prioritization is another important requirement in intelligent traffic management. Ambulances, fire trucks, and police vehicles may require priority at signalized intersections to reduce unnecessary delays during emergency situations. Previous studies have investigated computer-vision based emergency vehicle detection and traffic signal pre-emption using YOLO-based detection systems [1], [4], [5]. In these approaches, the detection of an emergency vehicle can be used by the controller to provide priority to the corresponding road. Such an approach can support emergency vehicle movement through intersections by temporarily modifying the normal signal sequence [1], [5].

Accidents and other abnormal traffic events can disturb normal traffic flow and require appropriate traffic-management responses. Recent computer-vision research has explored anomaly detection and traffic monitoring using deep-learning models to identify unusual traffic conditions [10]. YOLO-based systems have also been investigated for vehicle monitoring and anomaly analysis in heterogeneous urban environments [10]. In the proposed system, accident detection is treated as a separate image-classification task using a Vision Transformer (ViT)-based approach. The model classifies a traffic image or selected video frame into accident or no-accident categories. For video inputs, OpenCV is used to extract and process relevant frames before they are provided to the AI models.

The paper presents an Automated Traffic System (ATS) for intelligent four-way traffic control and management. The system considers the North, South, East, and West approaches and combines vehicle detection, emergency vehicle identification, and accident classification. The proposed architecture is motivated by earlier research on four-way AI-based traffic signal control and emergency vehicle prioritization [1], adaptive traffic management using YOLOv8 and traffic density [2], and integrated traffic-monitoring approaches based on YOLO and computer vision [5], [7]. Unlike a conventional fixed-time signal, the proposed controller calculates the green-

light duration according to the detected number of vehicles on the selected lane. The green duration is constrained between 30 and 90 seconds, while a two-second yellow phase is used during lane changes as part of the proposed implementation.

The proposed controller uses a priority-based decision mechanism to handle different traffic events. Pedestrian crossing requests are managed through a pedestrian override, while emergency vehicle detection receives the highest priority in the proposed control logic. Accident conditions are considered separately so that the affected lane can be handled appropriately. Emergency vehicle priority and traffic-signal pre-emption have been investigated in previous studies [1], [4], [5], while adaptive signal management based on traffic conditions have also been explored [2], [6]. The proposed ATS combines these concepts into a single software prototype. The system also provides a manual override option so that an operator can temporarily control the signal during testing or exceptional conditions. intelligence at the edge.

The proposed ATS is intended as a proof of concept for using artificial-intelligence-based visual perception in a closed-loop traffic-management system. The system combines adaptive signal control, emergency vehicle prioritization, accident detection, pedestrian handling, manual control, and real-time visualization within a single prototype. Existing research has separately explored adaptive traffic signals, emergency vehicle priority, anomaly detection, IoT-based traffic management, and intelligent traffic monitoring [1]–[10]. The proposed work brings several of these functions together at the software-prototype level. The system can provide a foundation for future development toward CCTV-based deployment, physical traffic signals, connected intersections, multimodal emergency detection, and more advanced AI-based traffic optimization.

II. LITERATURE REVIEW

Recent research in the area of intelligent transportation systems has been exploring computer vision, deep learning, and adaptive traffic control to manage congestion and monitor traffic in real-time. In [1], a YOLOv8-based approach was studied for vehicle detection in the context of intelligent transportation systems using aerial traffic imaging. The study demonstrated improved precision over the YOLOv5

and highlighted the suitability of YOLOv8 for real-time traffic monitoring while revealing challenges posed by vehicle appearance, lighting conditions, and object size variations. In [2], YOLOv8 was employed in an autonomous UAV platform for vehicle detection and traffic monitoring. Different variants of the YOLOv8 were tested, with YOLOv8n demonstrating the potential for real-time use, offering a good tradeoff between detection performance and processing speed. The study further demonstrated the applicability of deep-learning-based vehicle detection in generating traffic-density information. Vehicle detection and counting using YOLOv8 and OpenCV were studied in [3] in the context of urban traffic management. The researchers developed their own dataset and employed object detection and tracking to classify different vehicle categories. Results showed that YOLOv8 could be applicable in practical traffic surveillance and vehicle-counting scenarios, although with varying performance across different vehicle categories. Research into adaptive traffic signal control has been an active area of research. In [4], an enhanced deep reinforcement learning method was proposed for adaptive urban traffic signal control. This study focused on improving the efficiency of signal-control strategies under varying traffic conditions while addressing issues with training efficiency and robustness. Similarly, [5] proposed a multi-objective deep reinforcement approach to adaptive signal control that considered traffic efficiency, safety, and environmental impact at the intersections. Emergency vehicle detection has been an area of study for computer vision and deep learning methods.

In [6], a Deep ConvNet2D-based system was developed to recognize emergency vehicles and support traffic signal priority. The study demonstrated the potential of emergency vehicle recognition in enabling traffic signal control through interrupting normal signal sequences when an emergency vehicle was identified. Other recent research has explored improved YOLO architectures for traffic detection. In [7], an improved YOLOv8 model was developed to address class imbalance issues in vehicle detection datasets, showing that the distribution of classes in a dataset could have significant impact on detection accuracy.

In [8], a comparison of YOLOv8 and YOLOv10 showed the importance of model selection for

detecting different vehicle categories and complex traffic objects. Overall, the literature demonstrates strong progress in vehicle detection, adaptive signal control, emergency vehicle recognition, and intelligent traffic monitoring. However, many studies focus primarily on individual components rather than integrating vehicle-density estimation, emergency priority, accident classification, pedestrian control, and real-time visualization into a single intersection-management framework. The proposed Automated Traffic System (ATS) addresses this integration gap by combining YOLOv8, Vision Transformer-based accident classification, OpenCV keyframe processing, dynamic signal timing, emergency Green Corridor management, pedestrian override, and a Flask-based real-time dashboard.

III. METHODOLOGY

The proposed methodology for the ATS is classified into five phases, namely; traffic data acquisition, AI-based traffic analysis, traffic-state generation, priority-based signal decision-making, and monitoring. The system is designed for the four-way traffic intersection, including the lanes North, South, East, and West.

In the first step, traffic images or videos are given as input to the system for all lanes. The given videos are processed and analyzed using the OpenCV library, YOLOv8, and Vision Transformer. Initially, the OpenCV library is used to extract keyframes from the given videos as input to the AI-based traffic analysis module. Then these frames are given as input to the YOLOv8 model to detect, count the number of vehicles, and identify emergency vehicles, such as ambulances, fire brigade, and police. In parallel, few frames are given as input to the Vision transformer to classify whether the given frame is an accident or not, which classifies the traffic state of the lanes.

Finally, the traffic controller makes decisions based on the priority. Detection of pedestrians has been assigned the highest priority, followed by the detection of emergency vehicles and accidents. If none of these are detected, the system follows the regular dynamic procedure to assign the duration of green light to each of the lanes based on the number of vehicles at that time.

$$G = \min(90 \cdot \max(30 \cdot 30 + 2V))$$

where G represents green duration and V represents the detected vehicle count. Emergency detection initiates a 30-seconds green corridor while accident detection initiates an accident response procedure.

Finally, the Flask backend updates the traffic state using REST APIs, and the web dashboard displays signal status, vehicle counts, alerts, timers, and traffic statistics to the operator.

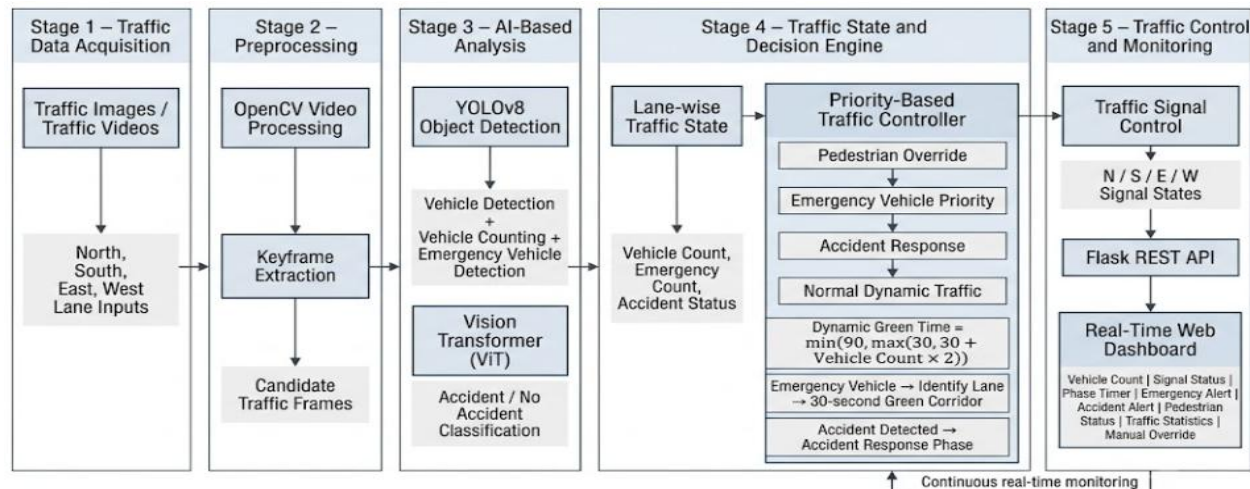


Fig 1: Methodology

Figure 1 illustrates the complete process of the proposed ATS. Initially, we acquire traffic images or videos from all four lanes of the targeted intersection. We utilize OpenCV for preprocessing as well as extracting key frames from the acquired videos. We subject the resulting frames to individual analysis using YOLOv8 and ViT.

From the analyzed data, we extract the relevant vehicle, emergency, and accident information and merge it with the corresponding lane traffic state. Based on the priority, the priority-based controller decides on the appropriate traffic control response including pedestrian override, emergency Green Corridor, accident reaction, or normal adaptive signal control.

The final control state is then relayed by the Flask backend to the real-time web dashboard as illustrated. This way, the pipeline connects the visual perception of the traffic to the automated traffic light control and decision-making process.

IV. IMPLEMENTATION

The Automated Traffic System (ATS) is an AI-powered traffic management system that uses computer vision, deep learning, adaptive traffic signal

control, and web development. It is designed to monitor the status of a four-way crossroad with north, south, east, and west lanes. The video feed from the camera is analyzed to detect vehicles, emergency vehicles, and accidents. This information is used to update the priority-based traffic controller to control the traffic signals.

YOLOv8 is used to detect vehicles and emergency vehicles, and a Vision Transformer (ViT) is used for accident classification. OpenCV is used for video processing and keyframe extraction.

The backend is built with Flask and contains the AI models for processing and analyzing the traffic data, updating the traffic status, REST API, and signal-control algorithm.

A web dashboard is built with JavaScript to view the vehicle detection, traffic status, alerts, timers, and pedestrian status in real-time. The web dashboard is also used to view the traffic.

A. Traffic Dataset and Input Preprocessing

The proposed system accepts traffic images and video files corresponding to the four lanes of the intersection.

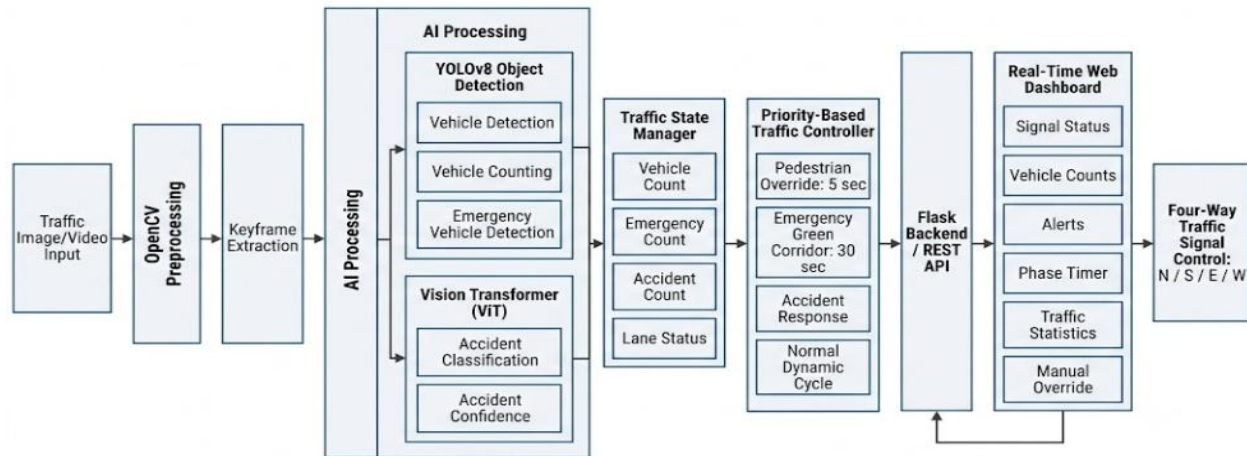


Fig. 2: Implementation Architecture of the Proposed ATS

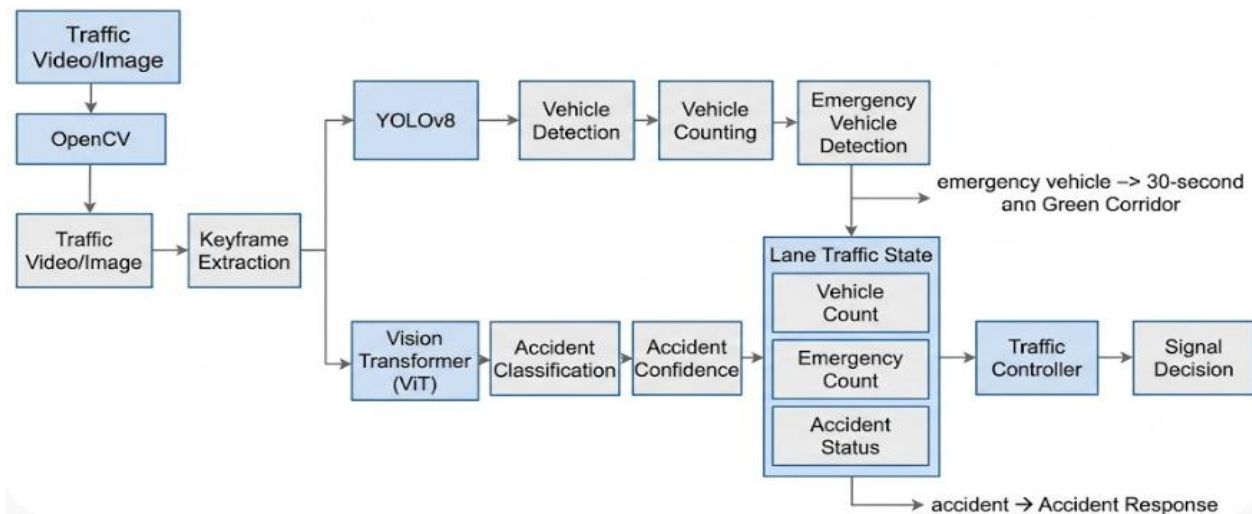


Fig. 3: Real-Time ATS Dashboard

The input media is processed before being supplied to the AI models.

- **Input Data:** Traffic images and videos representing road scenes from North, South, East, and West lanes are provided to the system through the web interface.
- **Media Processing:** OpenCV (cv2) is used to read uploaded videos, determine frame information, sample candidate frames, and extract relevant traffic scenes for AI analysis.
- **Keyframe Extraction:** Instead of processing every frame of a video, the implementation samples candidate frames and analyzes them to identify frames containing significant vehicle, emergency, or accident information.
- **Image Processing:** Extracted frames are converted into the appropriate image representation before

being passed to the detection and classification models.

- **Lane Association:** Each uploaded media file is associated with a specific lane, allowing the system to maintain independent traffic information for N, S, E, and W.
- **Output Generation:** AI-processed frames can be annotated with detected objects and stored under the configured static/outputs/ directory for visualization through the web dashboard.

B. YOLOv8-Based Vehicle and Emergency Detection
ATS uses a custom Ultralytics YOLOv8 object detection model to identify vehicles and emergency vehicles from traffic images and video frames.

- **Model:** The trained YOLOv8 weights are loaded from models/emergency_model.pt.

- **Vehicle Detection:** The model identifies configured vehicle objects appearing in the traffic scene and provides their corresponding bounding boxes and confidence scores.
- **Vehicle Counting:** Detected vehicle instances are counted for each lane and stored as the `vehicle_count` value in the global traffic state.
- **Emergency Detection:** The model is configured to identify emergency categories including ambulance, fire truck, police, army, and related configured classes.
- **Lane Identification:** Once an emergency vehicle is detected, its associated upload lane is used to identify the lane requiring priority.
- **Detection Output:** The detection results are transferred to the traffic controller for adaptive signal management.

The resulting information can be represented as:

$$V_L = \sum_{i=1}^n V_{vehicle_i}$$

where V_L represents the detected vehicle count for a particular lane.

C. Accident Classification Using Vision Transformer
An additional AI component is implemented for accident detection using a Vision Transformer (ViT) model.

- **Model:** tiya1012/vit-accident-image
- **Classification:** The model performs binary scene classification between accident and non-accident conditions.
- **Output Classes:**
LABEL_0 → No Accident
LABEL_1 → Accident
- **Confidence Threshold:** An accident confidence threshold of 0.35 is applied in the proposed implementation.
- **Video Integration:** Candidate frames generated by OpenCV are passed to the ViT model for classification.
- **Accident State:** When the accident confidence reaches the configured threshold, the corresponding lane is updated with an accident event.

The accident information is subsequently transferred to the traffic controller, which activates the defined accident-response phase.

D. Adaptive Traffic Signal Controller

The traffic controller is the primary decision-making component of the proposed ATS. It maintains the current state of all four lanes and continuously determines the appropriate signal phase.

Each lane maintains:

- Vehicle Count
- Emergency Count
- Accident Count
- Vehicle Signal
- Pedestrian Signal
- The global controller maintains:
 - Active Lane
 - Current Phase
 - Phase Timer
 - Pedestrian Override
 - Manual Override

A background daemon thread executes the traffic-control loop, while threading. Lock () is used to synchronize access to the shared traffic state.

E. Dynamic Green-Time Calculation

During normal traffic operation, the green duration is dynamically calculated according to the detected vehicle count.

The proposed equation is:

$$G = \min(90, \max(30, 30 + 2V))$$

where:

- G = green signal duration in seconds
- V = detected vehicle count

For example:

Vehicle Count	Green Duration
0	30 seconds
5	40 seconds
10	50 seconds
20	70 seconds
30 or more	90 seconds

The minimum green duration is restricted to 30 seconds, while the maximum duration is limited to 90 seconds.

During normal operation, the controller follows:

$N \rightarrow S \rightarrow E \rightarrow W \rightarrow N$

A 2-second yellow transition is introduced between green phases.

F. Priority-Based Traffic Management

ATS implements a priority hierarchy to handle exceptional traffic situations.

Pedestrian Override



Emergency Vehicle



Accident Response



Normal Dynamic Traffic

1. Pedestrian Override

When a pedestrian crossing request is received, the controller temporarily interrupts normal traffic operation and activates the pedestrian phase for the configured 5-second period.

2. Emergency Green Corridor

When an emergency vehicle is detected, the controller:

1. Identifies the emergency lane.
2. Interrupts the normal traffic cycle.
3. Sets the emergency lane to Green.
4. Sets conflicting lanes to Red.
5. Maintains the priority phase for 30 seconds.
6. Returns to automatic traffic control.

3. Accident Response

When an accident is detected through the ViT model, the corresponding lane is marked with an accident condition and the controller activates the Accident Response phase for the configured response period.

G. Flask Backend and REST API Implementation

The backend is implemented using Python Flask and provides communication between the AI-processing layer, traffic controller, and web dashboard.

The major APIs are:

GET /api/state

POST /api/upload/<lane>

POST /api/override

POST /api/override_pedestrian

The /api/state endpoint returns the current traffic state as JSON. The upload endpoint accepts lane-specific image or video files and initiates AI processing. The override endpoints provide pedestrian and manual signal-control functionality.

The Flask application also serves the HTML dashboard and manages communication between the frontend and backend components.

H. Real-Time Dashboard Implementation

The frontend is developed using HTML5, CSS3, and JavaScript. It provides a visual representation of the four-way intersection and allows the operator to monitor the system in real time.

The dashboard displays:

- North, South, East, and West signal states
- Vehicle counts
- Emergency vehicle alerts
- Accident alerts
- Active lane
- Current phase
- Phase timer
- Pedestrian status
- Traffic statistics
- Manual override controls
- Uploaded/annotated traffic outputs

The JavaScript frontend periodically requests /api/state and updates the displayed traffic information without requiring a complete page reload.

The uploaded images and videos of the traffic are processed by the backend, and its annotated version is available in the dashboard. In this case, one can see what triggered the AI-powered detection modules – literally see the traffic as the system does. It provides a visual confirmation of the detected vehicles and other traffic occurrences, enabling one to assess and review the decisions being made by the system.

The manual control panel is the other feature of the dashboard. It allows one to take over a specific channel and change the state of its traffic signal manually. It can be used during testing or to rare edge cases, while the override can be released to resume automatic operations.

In general, the dashboard is a place to view information about the traffic, observe the decision-making process of the system, receive emergency alerts, and manually interact with the Automated Traffic System. It ties together the AI perception module with the signal control panel providing an overview of the prototype's operations.



Fig. 4: Real-Time Traffic Signal and Monitoring Dashboard

I. Traffic State and Control Logic

The global traffic state is maintained as a structured object containing lane-specific and intersection-level information.

The simplified state representation is:

Lane State

- Vehicle Count
- Emergency Count
- Accident Count
- Signal State
- Pedestrian State

Global State

- Active Lane
- Current Phase
- Phase Timer
- Pedestrian Override
- Manual Override

The controller continuously evaluates this state and determines whether to remain in normal operation or transition into a priority phase.

J. System Evaluation Metrics

The proposed implementation can be evaluated using AI, traffic-control, and system-level metrics.

AI Detection Metrics

- Precision
- Recall
- F1-Score
- Detection Confidence
- Vehicle Counting Accuracy
- Emergency Vehicle Detection Accuracy

Accident Classification Metrics

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- Classification Confidence

Traffic-Control Metrics

- Average green time
- Average waiting time
- Emergency response phase duration
- Number of correctly triggered emergency corridors
- Accident-response activation accuracy
- Pedestrian override response time

System Metrics

- API response time
- AI inference time
- Video processing time
- Dashboard update latency
- CPU/GPU utilization
- Memory consumption

These metrics provide a comprehensive evaluation of both the AI components and the overall traffic-management functionality.

V. CONCLUSION

The proposed Automated Traffic System (ATS) provides an intelligent and automated approach to traffic management by integrating computer vision, deep learning, adaptive signal control, and real-time web-based monitoring. The system is designed for a four-way intersection consisting of North, South, East, and West lanes, where traffic conditions can continuously change. Unlike conventional fixed-time traffic signals, the proposed system dynamically adjusts signal duration according to the detected vehicle density. The system uses YOLOv8 for vehicle detection, counting, and identification of emergency vehicles such as ambulances, fire trucks, and police vehicles. A Vision Transformer (ViT) model is incorporated to classify accident conditions from traffic images and video frames. OpenCV is used for video processing and keyframe extraction, enabling the system to identify relevant frames for AI-based analysis. Based on the detected traffic conditions, the

controller calculates green-signal duration using a dynamic timing mechanism with predefined minimum and maximum limits.

A priority-based traffic control mechanism further improves the system by providing pedestrian override, emergency Green Corridor, accident response, and manual signal control. Emergency vehicles receive a dedicated 30-second Green Corridor, while pedestrian requests and accident situations are handled according to the defined priority hierarchy. The Flask backend and REST APIs enable communication between the AI processing modules, traffic controller, and real-time dashboard. Overall, ATS demonstrates how artificial intelligence and computer vision can be combined to create a responsive and scalable traffic management solution. The proposed system provides a strong foundation for future integration with live CCTV cameras, IoT-based traffic signals, multi-intersection coordination, edge computing, and predictive traffic analysis, supporting the development of smarter and safer transportation infrastructure.

REFERENCES

- [1] M. Alzubaidi *et al.*, “Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications,” *Digital Signal Processing*, vol. 152, p. 104594, 2024, doi: 10.1016/j.dsp.2024.104594.
- [2] Y. Zhang *et al.*, “Enhancing vehicle detection in intelligent transportation systems via autonomous UAV platform and YOLOv8 integration,” *Applied Soft Computing*, vol. 164, p. 112015, 2024, doi: 10.1016/j.asoc.2024.112015.
- [3] W. Choi, B.-J. Jang, I. Jung, C. Ryu, and S. Miraliev, “A comparative study on dynamic traffic object detection performance using YOLOv8-based deep learning model,” *Journal of the Korea Institute of Intelligent Transport Systems*, vol. 23, no. 6, pp. 159–173, 2024, doi: 10.12815/kits.2024.23.6.159.
- [4] C. Cai and M. Wei, “Adaptive urban traffic signal control based on enhanced deep reinforcement learning,” *Scientific Reports*, vol. 14, p. 14116, 2024, doi: 10.1038/s41598-024-64885-w.
- [5] G. Zhang, F. Chang, J. Jin, F. Yang, and H. Huang, “Multi-objective deep reinforcement learning approach for adaptive traffic signal control system with concurrent optimization of safety, efficiency, and decarbonization at intersections,” *Accident Analysis & Prevention*, vol. 199, p. 107451, 2024, doi: 10.1016/j.aap.2023.107451.
- [6] D. Li, F. Zhu, J. Wu, Y. D. Wong, and T. Chen, “Managing mixed traffic at signalized intersections: An adaptive signal control and CAV coordination system based on deep reinforcement learning,” *Expert Systems with Applications*, vol. 238, p. 121959, 2024, doi: 10.1016/j.eswa.2023.121959.
- [7] D. Chen, T. Xu, S. Ma, X. Gao, and G. Zhao, “Research on intelligent signal timing optimization of signalized intersection based on deep reinforcement learning using floating car data,” *Transportation Research Record*, vol. 2678, no. 12, 2024, doi: 10.1177/03611981241249736.
- [8] H. Yu, X. Zhang, Y. Wang, Q. Huang, and B. Yin, “Fine-grained accident detection: Database and algorithm,” *IEEE Transactions on Image Processing*, vol. 33, pp. 1059–1069, 2024, doi: 10.1109/TIP.2024.3355812.
- [9] C. Cai and M. Wei, “Adaptive urban traffic signal control based on enhanced deep reinforcement learning,” *Scientific Reports*, vol. 14, p. 14116, 2024, doi: 10.1038/s41598-024-64885-w.
- [10] G. Zhang, F. Chang, J. Jin, F. Yang, and H. Huang, “Multi-objective deep reinforcement learning approach for adaptive traffic signal control system with concurrent optimization of safety, efficiency, and decarbonization at intersections,” *Accident Analysis & Prevention*, vol. 199, p. 107451, 2024, doi: 10.1016/j.aap.2023.107451.
- [11] “A survey on deep reinforcement learning approaches for traffic signal control,” *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108100, 2024, doi: 10.1016/j.engappai.2024.108100.
- [12] “Enhancing adaptive traffic control systems with deep reinforcement learning and graphical models,” in *Proceedings of the 2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS)*, 2024, doi: 10.1109/FMLDS63805.2024.00017.
- [13] “Optimizing traffic signal control in mixed traffic scenarios: A predictive traffic information-based deep reinforcement learning approach,” in *Proceedings of the 2024 Forum for Innovative Sustainable Transportation Systems (FISTS)*, 2024, doi: 10.1109/FISTS60717.2024.10485533.