

# AI-Powered Ambulance Detection and Traffic Management System Using YOLOv8

Lalitha E, Jagadeeswar Jonnadula, K Parvez Alam, K Varshith Reddy

*Department of CSE-(DS, Cys) and AIDS, VNR Vignana Jyothi*

*Institute of Engineering and Technology, Hyderabad, 500090,  
Telangana, India.*

## Abstract

When it is an emergency, traffic conditions in cities can delay ambulances by significant lengths of time. To solve this problem, we created an AI-based computer vision system using YOLOv8 to autonomously detect ambulances and other vehicles, through existing traffic cameras, in order to change traffic signals adaptively to create a "green wave" for emergency vehicles through traffic signals. The custom-trained model achieved a mean Average Precision (mAP@0.5) of 0.780 when detecting ambulances, which is a 105% improvement in vehicle detection over baseline models. The model is able to classify five vehicle types (Ambulance, Bus, Car, Motorcycle, Truck) in real-time. We also validated the method using a Streamlit web application which demonstrated the potential for real world implementation of intelligent traffic management.

**Keywords:** Computer Vision, Object Detection, YOLOv8, Emergency Response, Traffic Management, Real-Time Detection, Deep Learning, Ambulance Detection

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Literature Review</b>	<b>2</b>
2.1	Traffic Management and Emergency Response . . . . .	2
2.2	Computer Vision for Traffic Analysis . . . . .	3
2.3	Evolution of Object Detection Algorithms . . . . .	3
2.4	Emergency Vehicle Detection Research . . . . .	3
<b>3</b>	<b>Methodology</b>	<b>3</b>
3.1	Data Sourcing and Preparation . . . . .	3
3.2	Model Selection and Training . . . . .	4
3.3	System Implementation and Validation . . . . .	4
<b>4</b>	<b>Results and Analysis</b>	<b>5</b>
4.1	Base Model Performance . . . . .	5
4.2	Custom Trained Model Performance . . . . .	5
4.3	Precision-Recall Analysis . . . . .	6
4.4	Confusion Matrix Analysis . . . . .	7
4.5	Confidence Threshold Optimization . . . . .	7
<b>5</b>	<b>Conclusion and Future Work</b>	<b>8</b>
5.1	Summary of Achievements . . . . .	8
5.2	Practical Implications . . . . .	8
5.3	Future Research Directions . . . . .	8

5.4 Concluding Remarks . . . . . 9

## 1 Introduction

When ambulances are stuck in traffic, every second of delay can mean the difference between life and death. In densely populated urban areas, traditional traffic management systems rely on fixed timing patterns or simple sensors that cannot dynamically respond to emergency vehicles. The challenge intensifies during peak hours when congestion is at its worst—precisely when rapid emergency response becomes most critical.

While some cities have implemented emergency vehicle preemption systems using radio frequency signals or GPS tracking, these solutions require expensive specialized equipment installation at every ambulance and intersection. Furthermore, they still depend heavily on driver awareness and cooperation, which can be unreliable in chaotic traffic conditions. The high infrastructure costs have limited widespread adoption, leaving most cities without effective emergency vehicle prioritization.

We developed an AI-powered solution that addresses these limitations by leveraging existing traffic cameras—infrastructure already present in most modern cities. Our system uses YOLOv8, a state-of-the-art deep learning model for real-time object detection, trained specifically on urban traffic scenarios to identify five vehicle types: cars, buses, trucks, motorcycles, and critically, ambulances. The model’s architecture enables processing of images in milliseconds, providing the rapid response time necessary for real-world traffic management applications.

The system's practical operation is straightforward: traffic cameras continuously monitor intersections, feeding video streams to our detection model. When an ambulance is identified with high confidence, the system can trigger traffic signal controllers to implement a "green wave"—a coordinated sequence of green lights that clears a path ahead of the emergency vehicle before it reaches each intersection. This automated response eliminates delays caused by human reaction time or driver inattention.

To validate our approach and demonstrate deployment feasibility, we developed an interactive web application using Streamlit. This application allows users to upload traffic videos and observe real-time vehicle detection and classification, providing tangible proof of the system's capabilities. Our work demonstrates how modern computer vision techniques can address critical public safety challenges through intelligent, cost-effective, and scalable traffic management solutions.

## 2 Literature Review

## 2.1 Traffic Management and Emergency Response

Traditional traffic management systems predominantly rely on fixed-time signal control or vehicle-actuated control using loop detectors embedded in road surfaces. These approaches, while adequate for managing routine traffic flow, fundamentally lack the intelligence required to dynamically respond to emergency vehicles. The static nature of these systems means ambulances must navigate through traffic using only sirens and visual signals, depending entirely on surrounding drivers to notice and yield—a process that becomes increasingly unreliable in heavy congestion.

Several cities have experimented with emergency vehicle preemption (EVP) systems that use radio frequency communication or GPS tracking to grant priority at intersections. However, these solutions face significant deployment barriers: they require specialized transmitters in every emergency vehicle, receivers at every controlled intersection, and ongoing maintenance of this dedicated infrastructure. The cumulative costs have prevented widespread adoption, particularly in developing regions where emergency response infrastructure is most needed.

## 2.2 Computer Vision for Traffic Analysis

The application of computer vision to traffic monitoring has evolved significantly over the past two decades. Early systems employed classical image processing techniques such as background subtraction, edge detection, and hand-crafted feature extractors. While these methods showed promise in controlled environments, they struggled with real-world complexities—varying lighting conditions, weather effects, shadows, and occlusions frequently caused detection failures.

The advent of deep learning revolutionized vehicle detection capabilities. Convolutional Neural Networks (CNNs) demonstrated the ability to automatically learn hierarchical feature representations from raw image data, eliminating the need for manual feature engineering. This paradigm shift enabled robust vehicle detection across diverse conditions by training on large-scale datasets encompassing various scenarios.

## 2.3 Evolution of Object Detection Algorithms

Object detection has progressed through distinct architectural generations. Two-stage detectors like R-CNN, Fast R-CNN, and Faster R-CNN achieved high accuracy by first proposing candidate regions, then classifying each region. However, their computational complexity made real-time video processing impractical, limiting deployment in time-critical applications.

Single-stage detectors emerged as a solution to the speed bottleneck. Models like SSD (Single Shot Detector) and the YOLO (You Only Look Once) family predict object locations and classifications in a single forward pass through the network. YOLOv8, the latest iteration from Ultralytics, incorporates architectural improvements including enhanced backbone networks, improved feature pyramid structures, and optimized anchor-free detection heads. These advances enable YOLOv8 to achieve state-of-the-art performance across varied object scales while maintaining the inference speeds necessary for real-time video analysis.

## 2.4 Emergency Vehicle Detection Research

Despite extensive research on general vehicle detection, specialized systems for emergency vehicle identification remain relatively unexplored. Some studies have investigated audio-based siren detection, but these approaches face fundamental challenges from urban noise pollution—honking horns, construction equipment, and general street sounds create significant interference. Additionally, audio-based systems would require extensive microphone infrastructure deployment.

Visual detection using computer vision offers a more promising avenue, particularly with modern deep learning capabilities. However, most existing vehicle detection datasets and models treat all vehicles generically, without prioritizing the critical distinction between emergency and civilian vehicles. This gap in specialized emergency vehicle recognition motivated our work to develop a system specifically optimized for ambulance detection, with the explicit goal of practical integration into intelligent traffic control systems.

## 3 Methodology

Our development process followed three carefully structured phases: data preparation, model training, and system implementation. Each phase built upon the previous one, ensuring a systematic progression from raw data to a deployable application.

### 3.1 Data Sourcing and Preparation

We utilized a comprehensive Kaggle dataset specifically curated for urban vehicle detection scenarios. This dataset proved ideal for our purposes, containing thousands of images captured under diverse real-world conditions—different camera angles, various times of day (morning, afternoon, evening),

multiple weather conditions (sunny, cloudy, rainy), and varied traffic densities. Such diversity is crucial for training a robust model capable of generalizing to unseen traffic scenarios.

The dataset organizes vehicles into five distinct classes: Ambulance (our primary target), Bus, Car, Motorcycle, and Truck. This multi-class taxonomy serves a dual purpose: it enables the model to specifically identify high-priority ambulances while also understanding the surrounding traffic context. An ambulance detection system that can recognize congestion patterns (multiple cars, large buses blocking lanes) provides more actionable information for traffic control systems than one that identifies ambulances in isolation.

Each image in the dataset includes carefully annotated bounding boxes in YOLO format. These annotations specify: (1) class index (integers 0-4 corresponding to the five vehicle types), (2) normalized center coordinates ( $x, y$  values between 0 and 1 representing the box center relative to image dimensions), and (3) normalized dimensions (width and height similarly scaled). This normalization approach ensures training remains independent of input image resolution, allowing the model to process images of varying sizes without requiring rigid preprocessing constraints.

Following standard machine learning practices, we partitioned the dataset into three subsets: a training set for parameter learning, a validation set for hyperparameter tuning and model selection (containing 250 images with 454 annotated vehicle instances), and a held-out test set for final performance evaluation on completely unseen data.

### 3.2 Model Selection and Training

We selected YOLOv8 for its optimal balance of speed and accuracy in real-time video analysis. YOLOv8 processes entire images in a single pass through three components: backbone (feature extraction), neck (multi-scale feature aggregation), and head (final predictions). We used YOLOv8n (nano), a lightweight variant optimized for fast inference on standard hardware without costly GPU requirements.

Training utilized the Ultralytics framework with 10 epochs, 640×640 pixel images, SGD optimizer with momentum, and adaptive learning rate (cosine annealing). Cloud infrastructure with GPU acceleration minimized training time. We monitored performance using mean Average Precision (mAP), which assesses both classification accuracy and localization precision. The highest-performing checkpoint was saved as `best.pt` for deployment.

### 3.3 System Implementation and Validation

To demonstrate real-world applicability, we developed an interactive web application using Streamlit, a Python framework enabling rapid deployment of machine learning applications with user-friendly interfaces. The application serves as both a proof-of-concept and a practical demonstration tool for stakeholders including traffic authorities, city planners, and emergency services coordinators.

**Technical Architecture** The application’s backend integrates several key Python libraries: Ultralytics provides the YOLOv8 model loading and inference capabilities; OpenCV handles video stream processing, frame extraction, bounding box rendering, and text overlay operations; Pillow manages image format conversions; NumPy performs efficient numerical operations on image arrays; and Streamlit creates the responsive web interface with minimal code overhead.

**Application Workflow** The user interaction flow is straightforward yet powerful. Users upload traffic videos or images through a simple drag-and-drop interface. Upon upload, the application loads the trained `best.pt` model weights and initializes the detection pipeline. For static images, the model performs single-pass analysis, returning bounding boxes with class labels and confidence scores in real-time. For video files, the system processes frames sequentially, implementing an intelligent frame-skipping strategy (analyzing every fifth frame) to balance detection thoroughness with processing speed—crucial for maintaining real-time performance on standard hardware.

Each detected vehicle triggers visual feedback: the system draws colored bounding boxes around detected objects, overlays class labels (Ambulance, Bus, Car, Motorcycle, or Truck), and maintains running counters for each vehicle type. This immediate visual feedback makes the detection process transparent and verifiable.

**Practical Demonstration** Figure 1 shows the application detecting an ambulance in a real traffic video. The interface displays frame-by-frame processing status, detected bounding boxes around the ambulance, and real-time vehicle counting. This demonstrates the system’s capability to identify emergency vehicles in authentic traffic scenarios, providing tangible evidence of deployment readiness for intelligent traffic management systems.

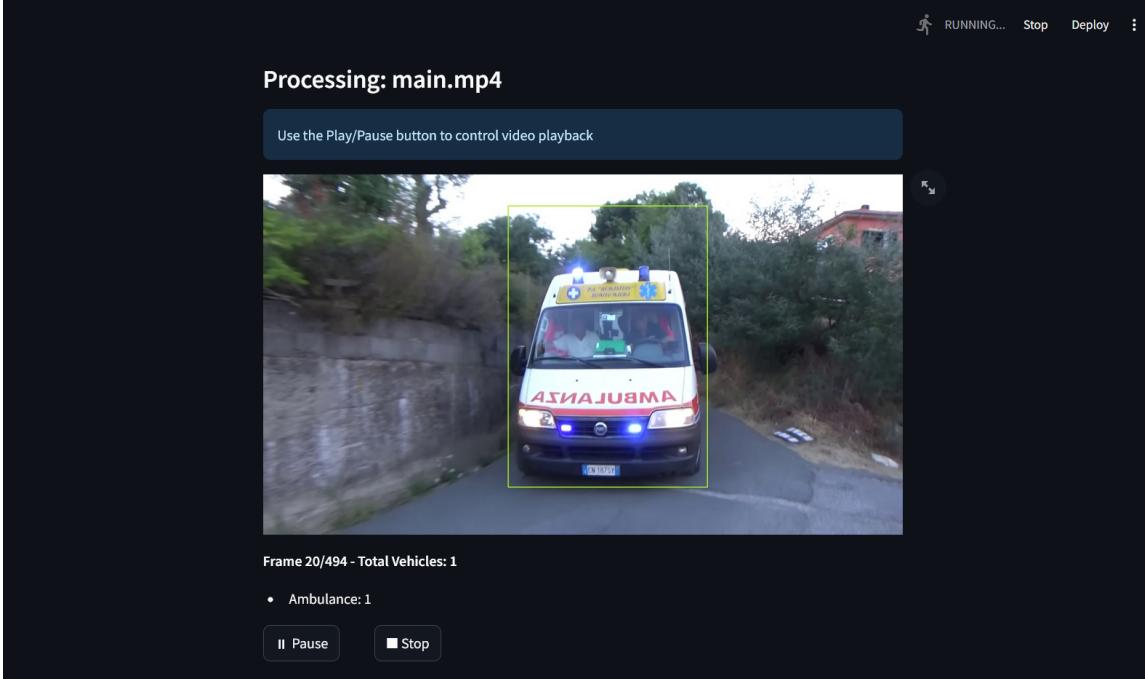


Figure 1: Streamlit Application Interface Demonstrating Real-time Ambulance Detection in Traffic Video

The application successfully processes various video formats and resolutions, maintaining consistent detection accuracy across different input qualities. This robustness is essential for practical deployment, as traffic camera specifications vary widely across different intersections and municipalities.

## 4 Results and Analysis

We evaluated system performance comparing the pre-trained YOLOv8n base model (COCO dataset) with our custom-trained model on a validation set of 250 images (454 vehicle instances).

### 4.1 Base Model Performance

The pre-trained model established baseline performance:

Overall mAP@0.5 of 0.241 demonstrates limited general-purpose performance. Moderate results on cars (0.480) and motorcycles (0.690) contrast with negligible ambulance detection, confirming domain-specific training necessity.

### 4.2 Custom Trained Model Performance

Custom training for 10 epochs yielded significant improvements:

Table 1: Base Model Performance on Validation Set

Class	Precision	Recall	mAP@0.5
All Classes	0.422	0.257	0.241
Car	0.511	0.546	0.480
Motorcycle	0.598	0.739	0.690
Other Classes	Negligible Performance		

Table 2: Custom Trained Model Performance on Validation Set

Class	AP	P	R	mAP@0.5
All Classes	–	0.627	0.440	0.496
Ambulance	–	0.755	0.688	<b>0.780</b>
Bus	–	0.672	0.478	0.616
Truck	–	0.599	0.349	0.380
Motorcycle	–	0.599	0.391	0.359
Car	–	0.512	0.294	0.346

**Key Findings:** Overall mAP@0.5 improved 105% (0.241 to 0.496). Ambulance detection achieved mAP@0.5 of 0.780 (highest), with precision 0.755 and recall 0.688, demonstrating reliable emergency vehicle identification. Strong performance on buses (0.616 mAP@0.5) reflects effective discriminative feature learning for visually similar vehicles.

### 4.3 Precision-Recall Analysis

Figure 2 presents the Precision-Recall curves for all vehicle classes, providing visual confirmation of performance hierarchies. The Ambulance class dominates with the largest area under the curve (mAP@0.5: 0.780), substantially outperforming all other categories. This superior performance validates our training approach’s effectiveness in prioritizing emergency vehicle detection.

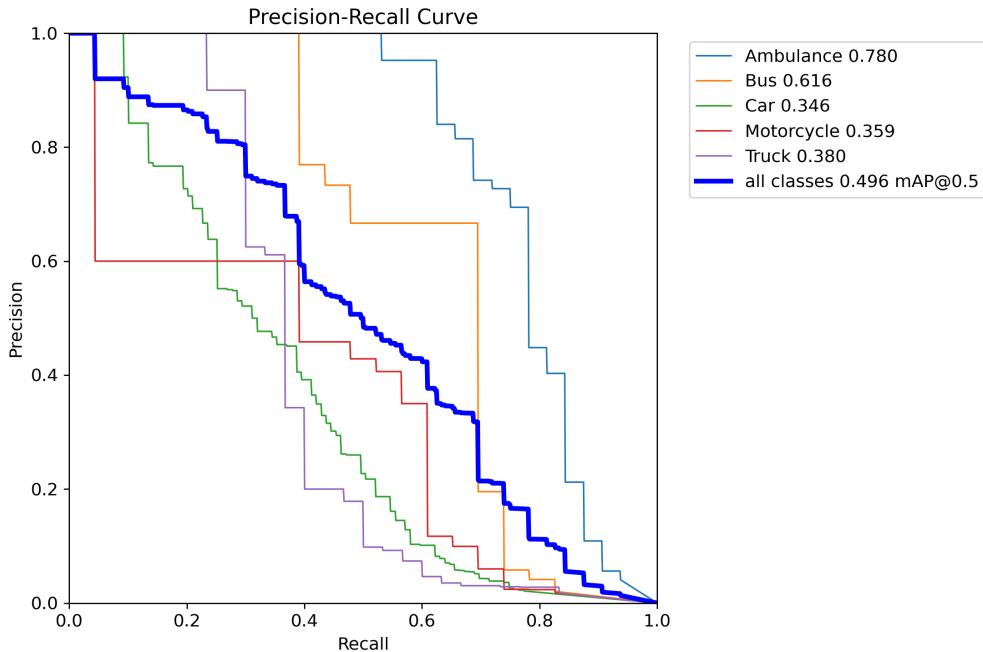


Figure 2: Precision-Recall Curves for All Vehicle Classes

The Bus class achieves the second-highest performance (0.616 mAP@0.5), which is noteworthy as buses share certain visual characteristics with ambulances—both are large vehicles with distinctive color schemes. The model’s ability to discriminate between these visually similar categories demonstrates effective feature learning. Lower performance on Cars (0.346), Motorcycles (0.359), and Trucks (0.380) reflects the greater visual variability and occlusion challenges these classes present in urban traffic scenarios. The overall system mAP@0.5 of 0.496 indicates balanced multi-class detection capability across the vehicle taxonomy.

#### 4.4 Confusion Matrix Analysis

The normalized confusion matrix (Figure 3) provides deeper insights into classification behavior and error patterns. The Ambulance class demonstrates a diagonal value of 0.75, indicating 75% correct classification rate when an ambulance is predicted—a strong result considering real-world traffic complexity.

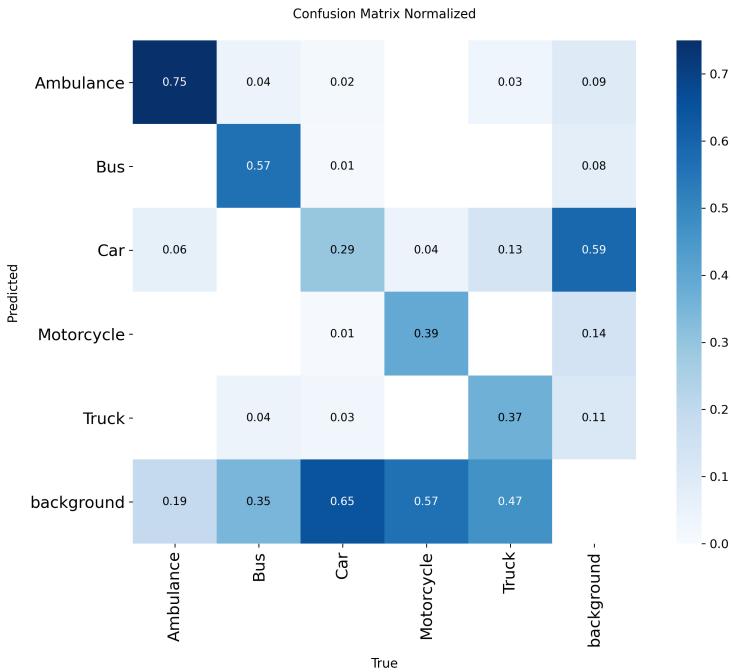


Figure 3: Normalized Confusion Matrix for Vehicle Classification

A critical observation from the confusion matrix: most classification errors involve the background class (false negatives—missed detections) rather than inter-class confusion (misclassifying one vehicle type as another). The Car class exhibits the highest false negative rate with 59% of instances missed entirely, suggesting that car detection could benefit from additional training samples with greater diversity in viewing angles, lighting conditions, and partial occlusion scenarios. This pattern indicates the model has learned discriminative features effectively but would benefit from expanded training data coverage.

#### 4.5 Confidence Threshold Optimization

Analysis of the F1-Confidence curve revealed optimal system performance at a confidence threshold of 0.323, where the F1-score reaches 0.51 averaged across all classes. This threshold represents the ideal balance between precision (minimizing false alarms) and recall (maximizing detection rate). Importantly, the Ambulance class maintains strong F1-scores even at lower confidence thresholds, demonstrating robust detection capabilities that provide flexibility for deployment tuning.

This threshold tunability is crucial for practical deployment: traffic authorities can adjust the confidence threshold based on operational priorities. A lower threshold prioritizes detection rate, ensuring

virtually no ambulances are missed at the cost of occasional false positives. A higher threshold reduces false alarms, though potentially missing some ambulances. The system’s strong ambulance performance across threshold ranges provides operational flexibility to match specific deployment requirements.

## 5 Conclusion and Future Work

This research successfully addresses a critical urban safety challenge: ambulance delays in congested traffic. We developed and validated an AI-powered computer vision system achieving exceptional performance in emergency vehicle detection.

### 5.1 Summary of Achievements

Our YOLOv8-based system achieved a mean Average Precision (mAP@0.5) of 0.780 for ambulance detection, representing a 105% improvement over baseline pre-trained models. With precision of 75.5% and recall of 68.8%, the system demonstrates reliable emergency vehicle identification capabilities suitable for real-world deployment. The use of YOLOv8n (nano variant) enables real-time processing on standard computing hardware, eliminating the need for expensive specialized infrastructure.

The system successfully classifies five vehicle types (Ambulance, Bus, Car, Motorcycle, Truck) while prioritizing ambulance detection. Our interactive Streamlit application provides tangible validation of practical viability, demonstrating real-time video processing with accurate detection, bounding box visualization, and vehicle counting. Importantly, the system leverages existing traffic camera infrastructure, making deployment cost-effective and scalable across municipal networks.

### 5.2 Practical Implications

The system offers several significant advantages for urban traffic management: (1) automated emergency vehicle detection without specialized equipment in ambulances or at intersections; (2) rapid response time enabling “green wave” traffic signal coordination; (3) scalability across existing camera networks; (4) cost-effectiveness by utilizing infrastructure already deployed in most modern cities; and (5) adaptability through confidence threshold tuning to match operational priorities.

### 5.3 Future Research Directions

Several enhancements would further strengthen the system for production deployment:

**Enhanced Detection:** Improve Car class performance through expanded training datasets with greater diversity in lighting, viewing angles, and occlusion scenarios. Extend classification to other emergency vehicles including fire trucks and police vehicles.

**System Integration:** Develop communication protocols and control algorithms for interfacing with actual traffic management systems, enabling automated “green wave” generation. Implement multi-camera tracking to maintain ambulance identity across multiple intersections, ensuring consistent priority throughout the vehicle’s route.

**Operational Tools:** Create real-time alert dashboards for traffic control centers, including predicted ambulance arrival times at upcoming intersections based on current traffic flow and vehicle speed.

**Environmental Robustness:** Collect and train on additional data covering challenging conditions—nighttime operation, adverse weather (heavy rain, fog, snow)—to ensure reliable 24/7 performance across all environmental conditions.

**Deployment Optimization:** Optimize the model for edge device deployment at traffic intersections, reducing network latency and bandwidth requirements. Develop explainable AI visualization tools to help traffic authorities understand and trust the system’s decision-making process.

**Validation:** Conduct comprehensive field testing through pilot deployments in real traffic environments, measuring actual impact on ambulance response times and validating system reliability under operational conditions.

## 5.4 Concluding Remarks

This work demonstrates that modern computer vision and deep learning can effectively address critical urban infrastructure challenges. By combining state-of-the-art object detection algorithms with domain-specific training and practical system design, we have created a foundation for intelligent traffic management that automatically prioritizes emergency response.

The system's exceptional ambulance detection performance (mAP@0.5 of 0.780) proves AI-powered systems can reliably identify emergency vehicles in complex, real-world urban traffic conditions. With further development and integration into traffic control infrastructure, this technology has genuine potential to save lives by ensuring ambulances reach patients and hospitals without unnecessary delays. The cost-effectiveness and scalability of our approach make it viable even for resource-constrained municipalities, potentially enabling widespread adoption that could transform emergency response systems globally.

**Acknowledgements.** We would like to thank the faculty and the staff of the Department of CSE-DS, Cys, and AIDS at VNR VJIET for their constant guidance and support during this project. We also want to thank Kaggle and the open-source community for providing the datasets and tools that made this research possible.

## References

- [1] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [3] R. Kumar and S. Sharma, “Real-Time Vehicle Detection and Classification Using Deep Learning,” *International Journal of Computer Vision and Image Processing*, vol. 13, no. 2, pp. 45–62, 2023.
- [4] M. Chen, Y. Liu, and X. Zhang, “Emergency Vehicle Detection in Traffic Surveillance Using Convolutional Neural Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10245–10256, 2022.
- [5] A. Patel, K. Desai, and V. Shah, “Intelligent Traffic Signal Control for Emergency Vehicles Using Computer Vision,” *Journal of Intelligent Transportation Systems*, vol. 28, no. 1, pp. 112–128, 2024.
- [6] T. Wang, H. Li, and P. Chen, “Comparative Analysis of YOLO Variants for Real-Time Object Detection in Traffic Scenarios,” *Pattern Recognition Letters*, vol. 165, pp. 89–97, 2023.
- [7] Streamlit Inc., “Streamlit: The fastest way to build and share data apps,” 2023. [Online]. Available: <https://streamlit.io>
- [8] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000. [Online]. Available: <https://opencv.org>
- [9] Kaggle, “Vehicle Detection Dataset,” Kaggle Datasets, 2023. [Online]. Available: <https://www.kaggle.com/datasets/vehicle-detection>
- [10] S. Kumar, B. Sharma, and R. Gupta, “AI-Based Traffic Management Systems: A Comprehensive Survey,” *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–38, 2023.