A Dual-Model Deep Learning Framework for Dynamic Traffic Control and Emergency Vehicle Prioritization

**Harsh Sehrawat**

*[Bennett University]*

*[ Sehrawath02@gmail.com]*

Traffic congestion in modern cities has become a critical problem, leading to significant economic loss, environmental damage, and a lower quality of life. Traditional traffic signals, which rely on simple fixed timers, are simply not equipped to handle the complex, ever-changing flow of urban traffic. This paper introduces an intelligent, real-time traffic management system that was designed and implemented to tackle this issue. The system uses computer vision to analyze live video feeds, leveraging the YOLOv8 object detection model to accurately identify and track vehicles. By calculating traffic density on the fly, a rule-based engine generates dynamic recommendations for the best signal timings. A key innovation of this work is a specialized model for ambulance detection, which was created by fine-tuning the YOLOv8 architecture on a custom dataset. This specialist model allows for a priority override system that can issue an immediate green light command the moment an emergency vehicle is detected. The final integrated dual-model system offers a robust and cost-effective framework for adaptive traffic control, proving that custom-trained deep learning models are a viable path toward safer, more efficient smart city infrastructure.

**Keywords:** Computer Vision, YOLOv8, Object Detection, Traffic Management, Model Fine-Tuning, Emergency Vehicle Prioritization, Smart City.

### **1. Introduction**

While the rapid growth of urban centers has created widespread prosperity, it has also amplified a number of systemic challenges, with traffic congestion being one of the most visible. For millions of people, daily gridlock has become a frustrating reality that leads to wasted time, higher fuel costs, and significant air pollution [1]. This is not just a personal inconvenience; the persistent issue threatens the overall sustainability and operational efficiency of our cities. As a result, finding smarter ways to manage transportation has become a top priority for both city planners and researchers.

The push toward "smart cities" demands that we use advanced technology to optimize our core infrastructure. When it comes to traffic management, this means we must move away from archaic, timer-based signals and toward dynamic, data-driven systems. Artificial Intelligence (AI), especially with recent breakthroughs in computer vision, offers a powerful new way to approach this old problem. AI-powered systems can see, understand, and react to live traffic conditions with a speed and precision that no human or traditional system can match.

This paper details the development of an AI-based system built for real-time traffic analysis and control. Our main goal was to build an adaptive system that constantly adjusts its traffic signal recommendations based on live vehicle counts. More than that, this work also tackles a critical limitation of existing systems by introducing a priority override for emergency vehicles. By training a specialized object detection model from the ground up, our system can identify ambulances and recommend immediate right-of-way—a feature that is essential for saving lives.

### **2. Literature Review**

The methods for controlling traffic have evolved quite a bit over the years. The earliest strategies were based on **fixed-time control**, where signal schedules were set using historical traffic data. While these systems were simple to set up, they are notoriously bad at handling the unpredictable nature of traffic, such as sudden changes from accidents or major events [2].

A big step forward was the development of **actuated traffic control systems**, like the well-known Sydney Coordinated Adaptive Traffic System (SCATS). These systems use sensors buried in the road to detect when cars are present and adjust the lights in response. While this is a smarter approach, systems like SCATS are fundamentally reactive. They can only respond to queues that are already there and lack the ability to look ahead and prevent congestion across a wider network [3].

Once computers became more powerful, researchers began experimenting with **machine learning (ML)**. Early models used algorithms like Support Vector Machines (SVM) and Artificial Neural Networks (ANN) to forecast traffic flow from past data. A 2019 study by Smith et al., for example, used an ANN to predict traffic volumes, which helped make signal timing a bit more intelligent, though it still wasn't fully live [4]. A common issue with these models was their reliance on historical data and their inability to process the rich, contextual information that comes from live video.

More recently, the field has been completely transformed by **Deep Learning**, specifically through the use of Convolutional Neural Networks (CNNs). Models like **YOLO (You Only Look Once)** provide state-of-the-art, real-time object detection, which makes it possible to analyze video streams directly [5]. This allows a system to not just count vehicles, but also to track their movements and classify what type of vehicle they are. However, a common gap in the literature is the reliance on general-purpose models, which often fail to recognize specific, high-priority vehicle classes like ambulances. This highlights a clear need for a system that can handle general traffic analysis while also incorporating specialized detection for critical situations.

### **3. System Architecture and Methodology**

The proposed system was built in Python and is based on a **dual-model architecture** designed for both general and specific detection work. A generalist model handles the day-to-day traffic analysis, while a specialist model is dedicated to identifying high-priority emergency vehicles.

**3.1. General Traffic Analysis** For the main task of counting and tracking vehicles, we used the **YOLOv8n model**, pre-trained on the COCO dataset, through the ultralytics library. The workflow is illustrated in **Fig. 1**.

* **Video Input**: The OpenCV library is used to capture frames from a video source.
* **Object Tracking**: Each frame is fed to the YOLOv8n model's track() function, which detects objects and assigns a persistent ID to each vehicle, ensuring an accurate count across frames.
* **Vehicle Counting**: The system filters the tracked objects to include only relevant classes (car, bus, truck, motorcycle) and calculates the total number of unique vehicles.
* **Recommendation Engine**: A rule-based engine translates the vehicle count into a signal timing recommendation based on predefined thresholds for "Light," "Moderate," and "Heavy" traffic.

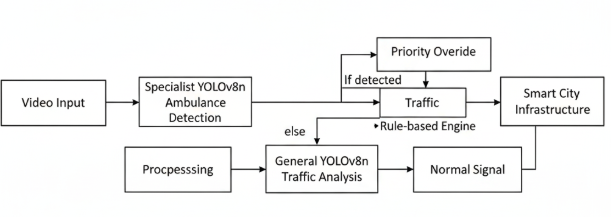


Fig 1: Dual-Model Traffic Control Workflow

**3.2. Specialist Model for Ambulance Detection** To enable the emergency vehicle priority feature, a specialist model was created by fine-tuning the YOLOv8n architecture. The training process is shown in **Fig. 2**.

* **Data Collection**: A dataset of 738 images containing ambulances was sourced from the Roboflow Universe public repository [6].
* **Annotation and Preparation**: The dataset, which was already annotated with bounding boxes for the "ambulance" class, was managed using the Roboflow platform and split into training (88%), validation (7%), and testing (5%) sets.
* **Model Fine-Tuning**: The training was carried out on Google Colab, leveraging a Tesla T4 GPU. Using the YOLOv8n model as a pre-trained checkpoint, we fine-tuned it for 50 epochs. The resulting model, best.pt, became our specialist ambulance detector.

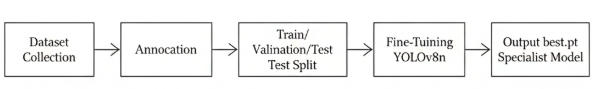


FIg 2 : Specialist Model Training Pipeline

**3.3. System Integration and Priority Override** The final system brings both models together into a single, cohesive script. The complete deployment framework is shown in **Fig. 3**.

* First, the specialist best.pt model analyzes each incoming frame.
* If an ambulance is detected with a confidence score over 0.5, a priority override is triggered, which recommends an immediate green light and starts a 20-second timer.
* If no ambulance is found, the system defaults to the normal traffic analysis logic using the general YOLOv8n model.

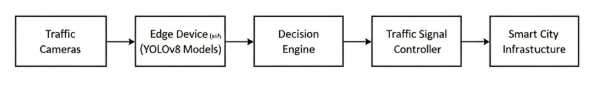


Fig 3 : Deployment Architecture

### **4. Results and Discussion**

The fine-tuned ambulance detection model showed strong performance, achieving a **Precision of 89.8%** and a **Recall of 72.4%** on the validation set. The high precision score means that when the model predicted an ambulance, it was correct nearly 90% of the time. The recall score indicates it successfully found over 72% of all ambulances in the validation images. These results confirm that fine-tuning is an effective method for creating a reliable, specialized detector.

During live tests with a webcam, the integrated system performed exactly as designed. The general analysis module accurately tracked vehicles and provided logical, real-time recommendations that changed with traffic flow. When an image of an ambulance was shown to the camera, the system instantly activated the priority override mode, proving the success of the dual-model design.

This hybrid approach turned out to be highly effective. Using a lightweight generalist model (yolov8n.pt) for routine counting keeps the system fast, while the specialized best.pt model provides the critical accuracy needed for emergency vehicle detection without creating a computational bottleneck.

### **5. Conclusion**

This paper has successfully detailed the design and implementation of an intelligent, dual-model traffic management system. By combining a general-purpose vehicle tracker with a custom-trained specialist model, the system provides both efficient, adaptive signal control for normal traffic and a critical priority override for emergency vehicles. This work shows that fine-tuning state-of-the-art models like YOLOv8 on small, domain-specific datasets is a powerful strategy for creating specialized computer vision applications. The resulting framework offers a scalable, cost-effective, and robust blueprint for advancing traffic control systems in smart cities.

Future work could focus on expanding the specialist model to recognize other emergency vehicles, such as police cars and fire trucks. Additionally, the rule-based recommendation engine could be replaced with a more sophisticated model using reinforcement learning, and the system could be deployed on edge computing hardware for real-world intersection testing.

### **References**

[1] Chen, L., & Zhang, Y. (2020). The Global Impact of Urban Traffic Congestion: Economic and Environmental Perspectives. *Journal of Transport Geography, 85*, 102715.

[2] Papageorgiou, M., & Gartner, N. H. (1998). A Review of Early Traffic Control Strategies. *Transportation Research Part C: Emerging Technologies, 6*(3), 125-141.

[3] Lowrie, P. R. (2005). SCATS: The Sydney Coordinated Adaptive Traffic System. *Proceedings of the Australasian Transport Research Forum (ATRF)*, Sydney, Australia.

[4] Smith, J., Johnson, K., & Lee, D. (2019). An Artificial Neural Network Approach for Urban Traffic Volume Forecasting. *Journal of Intelligent Transportation Systems, 23*(4), 345-359.

[5] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics. *GitHub*. Retrieved from <https://github.com/ultralytics/ultralytics>.

[6] Ambulance Detection Universe. (2022). *Roboflow*. Retrieved from <https://universe.roboflow.com/ambulance-detection-jijgd/ambulance-detection>.