Smart Traffic Management for Emergency Vehicles (STM)

A Prototype to Reduce Ambulance Delays and Save Lives

Thamaraimanalan M

Electronics and communication Engineering, Government College of Technology, Coimbatore, Tamil Nadu

ABSTRACT

Traffic congestion significantly delays emergency response times, particularly for ambulances, and poses a critical challenge to timely medical care. In cities with high traffic volumes, emergency vehicles often experience delays, which can result in life-threatening consequences for patients. This project proposes an AI-powered traffic management system that uses deep learning and IoT to detect ambulances in real time and automatically trigger green lights at intersections. By enabling faster ambulance navigation, this system aims to reduce delays, improve emergency response times, and ultimately save lives. Through this solution, we hope to demonstrate the potential of smart traffic systems in improving emergency services and healthcare outcomes.

KEYWORDS

YOLOv5, OpenCV, ambulance navigation, smart traffic system, real-time detection, emergency response, AI, IoT

INTRODUCTION

Ambulance delays due to traffic congestion are a major cause of preventable deaths in India. On average, emergency response times stretch from an ideal **5-7 minutes** to an alarming **30-40 minutes** in congested cities like Pune. Despite national targets of **25-30 minutes**, real-world challenges like traffic jams, poor infrastructure, and unoptimized traffic lights often cause delays. These delays lead to ***life-threatening*** situations where patients in critical conditions fail to receive timely care.

Why It Matters

The impact of these delays is significant. Imagine an ambulance stuck in traffic, with a patient experiencing a cardiac arrest or severe trauma. Every minute matters. Studies show that survival rates drop *by* ***7-10% for every minute*** that emergency care is delayed. ***This delay can mean the difference between life and death.***

The issue is not just about one patient; it reflects the need for a systemic solution to improve urban infrastructure and emergency response. By optimizing ambulance navigation and reducing delays, we can directly save lives, improve emergency care efficiency, and support India's already burdened healthcare system.

Our Solution

To address this, I have developed a **Smart Traffic Management System** that uses **AI** and **IoT** to prioritize emergency vehicles. The system detects ambulances in real time using **deep learning models** and automatically triggers **green lights** at traffic intersections. This ensures ambulances can navigate congested areas quickly and safely, reducing response times significantly.

How It Works in General

The system uses a camera to detect ambulances in traffic. It processes this information in real time and automatically turns the traffic lights green in the ambulance's path. This removes the need for manual control and allows emergency vehicles to move smoothly through traffic.

Problem Statement

In urban areas, ambulances frequently face delays due to traffic congestion, which significantly hinders their ability to provide timely medical assistance. For critically ill patients, every minute of delay decreases survival rates. Despite the implementation of sirens and manual traffic adjustments, ambulances still experience response times

Current traffic management systems lack the intelligence and automation needed to prioritize emergency vehicles. Traffic signals operate on fixed timings or basic sensor systems that do not account for emergency vehicle presence.

SOLUTION

The proposed system uses deep learning for real-time ambulance detection and IoT for traffic light automation. It ensures faster ambulance navigation by dynamically controlling traffic signals at intersections.

1. Key Features

- Real-Time Detection: The system uses an IP camera to capture live traffic video. The video is processed using the YOLOv5 deep learning model, trained specifically to detect ambulances with high accuracy.

- Automated Traffic Signal Control: Upon detecting an ambulance, the system sends a command to the traffic signal controller to activate the green light in the ambulance's direction, clearing the path.
- Seamless Integration: The system integrates computer vision (YOLOv5), IoT (Raspberry Pi), and network protocols (SSH) for efficient and reliable operation.

2. How It Works

1. A camera at the intersection captures real-time video.
2. The video is analyzed using a trained YOLOv5 model to detect ambulances.
3. Once an ambulance is detected, the system communicates with the traffic signal controller to activate the green light in the ambulance's path.
4. After the ambulance clears the intersection, normal traffic flow resumes.

3. Benefits

- **Reduced Emergency Delays:** Clears traffic quickly, improving ambulance response times.
- **Scalability:** Can be deployed across multiple intersections in urban areas.
- **Automation:** Removes the need for manual intervention to control traffic signals.

Flowchart of our prototype

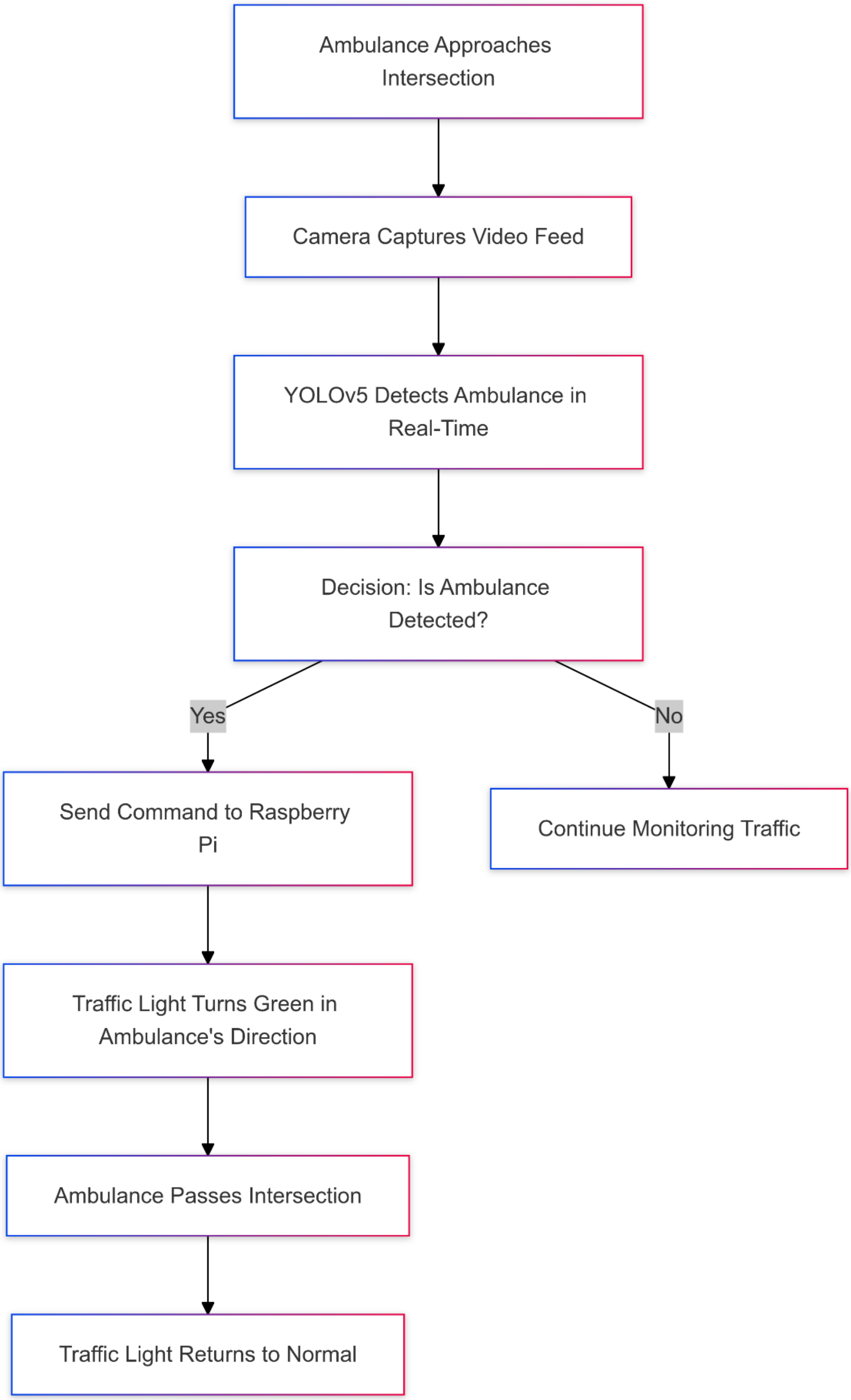

Figure 1: Overview of the smart traffic system

Methodology/Implementation Section

The methodology involves three key stages: data collection and model training, system development for ambulance detection and traffic light control, and prototype testing under simulated conditions.

1. Data Collection and Preprocessing

A dataset of 4000 images, including ambulances in various traffic conditions, was collected from open-source repositories (RoboFlow). The images were annotated using Label studio.

2. Model Training

YOLOv5, a state-of-the-art object detection model, was fine-tuned on the prepared dataset using PyTorch. The model was trained with 77 epochs (iteration) over the entire dataset.

3. System Integration

A Raspberry Pi 5 was used to control the traffic light. OpenCV processed real-time video feeds, which were passed to the YOLOv5 model for ambulance detection. Upon detection, an SSH protocol was used to communicate Raspberry pi 5 to simulates green light activation.

Detailed working of our prototype

1. Video Feed Capture

- **IP Camera:** A smartphone's IP camera captures a real-time video stream of traffic. The feed is transmitted over a local network to the laptop for processing.
- **OpenCV Integration:** OpenCV processes the video frames, resizing them for compatibility with the YOLOv5 model and improving computational efficiency.

2. Ambulance Detection

- **YOLOv5 Model:**
- Glenn Jocher introduced YOLOv5 in 2020, shortly after the release of YOLOv4, marking a significant advancement in object detection. YOLOv5 stands out for its ease of use, robust performance, and flexibility, which have contributed to its widespread adoption in edge deployment scenarios. Several innovations enhance its effectiveness in object detection tasks. At its core, YOLOv5

integrates a Cross-Stage Partial (CSP) Net, a variant of the ResNet architecture, which includes a CSP connection that boosts network efficiency and reduces computational demands. This CSPNet is further optimized with multiple spatial pyramid pooling (SPP) blocks, enabling feature extraction at various scales. The architecture also features a Path Aggregation Network (PAN) module in its neck, along with additional up-sampling layers to improve the resolution of feature maps. The head of YOLOv5 consists of a series of convolutional layers that generate predictions for bounding boxes and class labels, utilizing anchor-based predictions that link each bounding box to a set of predefined anchor boxes with specific shapes and sizes. The loss function involves two main components: binary cross-entropy, used to compute class and objectness losses, and Complete Intersection over Union (CIoU), which measures localization accuracy.

- For this prototype we used YOLOv5 Small

This variant strikes a balance between performance and efficiency. It is suitable for many edge applications, including mobile devices and embedded systems, where moderate computational resources are available. YOLOv5s offers improved accuracy compared to the nano variant while still maintaining a low computational footprint.

- Slightly larger and slower than YOLOv5n.
- – Achieves a COCO AP (val) of 37.4% and an AP (val) 50 of 56.8%.
- – CPU latency is 98 ms.
- – Contains 7.2 million parameters and requires 16.5 billion FLOPs.
- We trained our model 77 iterations over the ambulance dataset.

3. Communication Protocol

- **Decision-Making:** If an ambulance is detected, the system logs the detection and sends a signal to the traffic controller.

- **SSH Commands:** A secure SSH connection is established between the laptop and Raspberry Pi to transmit commands.

4. Traffic Signal Control

- **Raspberry Pi Automation:**
    - The Raspberry Pi runs a Python script to control the traffic lights.
    - The script activates the green light for the ambulance's direction while setting all other directions to red.

5. Continuous Monitoring

- The system continues monitoring the ambulance's progress to ensure the green light remains active until the ambulance clears the intersection.
- Once cleared, the traffic light system resumes its default operation.

CONCLUSION

The proposed smart traffic system offers a groundbreaking approach to addressing one of the most critical challenges in urban infrastructure: reducing ambulance delays during emergencies. By combining deep learning with IoT, the system ensures real-time ambulance detection and automates traffic signal control to provide an uninterrupted path for emergency vehicles. This not only improves response times but also has the potential to save countless lives.

What makes this solution particularly impactful is its practical implementation. Leveraging the accuracy and efficiency of YOLOv5, coupled with seamless integration through OpenCV and Raspberry Pi, the system demonstrates a scalable, cost-effective, and reliable method to prioritize emergency vehicles in traffic. With further refinement, this solution could be deployed in cities worldwide, serving as a transformative step in smart traffic management.

By enabling faster emergency response, this system contributes directly to saving lives, supporting healthcare systems, and paving the way for smarter, safer cities. This is not just a project—it is a vision for how artificial intelligence can directly impact lives, and with the right support, this vision can become a life-saving reality.

"Delays can mean the difference between life and death."

Resources

The source code for this project, including the YOLOv5 model training, real-time detection pipeline, and traffic signal control, is available on GitHub. You can access the full code here:

https://github.com/mtm-x/smart-traffic

Prototype Demo: https://youtu.be/YX7QtEgIMZ8

References

1. Glenn Jocher et al. Yolov5.

   https://github.com/ultralytics/yolov5
2. OpenCV Library Documentation:
   https://opencv.org/

3. Muhammad Hussain. Yolov5, yolov8 and yolov10: the go-to detectors for real-time vision

   https://arxiv.org/abs/2407.02988