Smart Traffic Management System for Emergency Vehicle

Mr. B. Ramesh[1], Md. Ashique[2], V. Karthik[3], Sk. Nayeem[4]
[1,2,3,4]Department of CSE (Data Science) CMR Technical Campus Hyderabad, India

doi.org/10.64643/IJIRTV12I11-195629-459

Abstract— **Efficient ambulance movement through congested urban roads is a major problem in emergency response systems. To overcome this problem, in the presented work an AI-based Smart Ambulance Detection framework is proposed in which existing CTV (Centralized Traffic Violations) surveillance infrastructure, deep learning and IoT-enabled traffic control are used. The system uses a camera-based detection method with a YOLOv5 object detection model and OpenCV for the frame processing part to accurately detect ambulances in real time based on video detection. Upon successful detection, a backend Flask based triggers and automated alert will communicate to a microcontroller unit such as ESP32 or Arduino to enable dynamic modification of traffic signals states to prioritize ambulances route. The solution does not require any further on-vehicle hardware which simplifies easy deployment and cost-effectiveness. Experimental results demonstrate that the system performs reliably in varied lighting and traffic circumstances thus showing the potential to greatly lessen emergency response delays and improve the efficiency when transporting traffic in urban areas.**

Index Terms— **YOLOv5, Computer Vision, CCTV Surveillance, OpenCV, Deep Learning,Signal Automation.**

I. INTRODUCTION

Quick healthcare service is a very important aspect of emergency medical care systems, which the slightest delay in the transfer of ambulances can impact greatly on patient survival rates. Urban settings, however, tend to have problems in terms of heavy traffic, low predictability of congestion and poor communication between emergency vehicles and road systems. This has led to how often ambulances get blocked access to hospitals particularly at peak time. Intelligent traffic control has come up as an opportunity to overcome these challenges due to the improvement of artificial intelligence and vision-based sensing. Deep learning models produced recently, especially real-time object detectors like YOLOv5, are able to identify vehicles in real-time camera streams with great accuracy. Such capabilities make automated surveillance systems able to identify ambulances with no extra hardware, such as GPS devices, RFID devices, or siren-based sensors, which can malfunction in noisy or blocked conditions. In this project, a Smart Ambulance Detection System is proposed that will incorporate CCTV cameras, image processing with AI, and the IoT-based control of the traffic lights. The system uses the live video of the traffic cameras

available and processes each frame based on OpenCV and a trained YOLOv5 model to recognize ambulances based on their structural and visual attributes. After being identified, a Flask based API sends an automated warning to a microcontroller unit- e.g. ESP32 or Arduino- which will dynamically adjust the timing of traffic signals to allow the emergency vehicle a clear path. The suggested solution is non-invasive, has a low price-tag and can be scaled because the proposed solution only makes use of the existing camera infrastructure and lightweight computation modules. Integrating computer vision with IoT communication, the system will help to enhance the efficiency of response during an emergency and minimise delays, as well as contribute to the creation of smarter and safer urban mobility systems.

Moreover, the system is developed to work effectively in any environmental conditions, such as at night and in the dense traffic, which is why it can be used in the real world. The modular structure is also such that the detection model can be retrained or enhanced with the more recent dataset comes available to the system, and the system will not become stuck in the scenario. The system is able to store detection logs to be analyzed later and assist the city-level traffic optimization strategies by incorporating cloud connectivity. The method also eliminates the use of human monitoring in traffic control rooms thus reducing the chances of human error and enhancing uniformity in its operations. In general, this intelligent and automated system adds to the vision of the next-generation intelligent transportation systems that should be able to provide essential emergency services with the level of reliability.

II. LITERATURE SURVEY

The work of Redmon and Farhadi [1] was the initiator of most of the modern trend of real-time detection. Their YOLOv3 model demonstrated that even the single-shot based models with high speed could maintain their accuracy even when faced with a cluttered view. It manipulated objects of various sizes but was sometimes incapable of working with small or partially obscured vehicles. Such a limitation can be observed in traffic recording videos; generally, ambulances are stalled by traffic jams most of the time. Most real-time vision systems are constructed based on documentation of OpenCV [2]. Everything that developers do is based on it, including frame grabbing and noisy input clearing. It can perform wonders when all the lights are stable, but in CCTV feeds where there is a glare or night shadows, preprocessing still requires fine Tuning. Those discrepancies may have an impact on accuracy of downstream detection. I boards are at times sluggish to manage heavy inference workloads, in other words, it is useful in data routing or some simple preprocessing when interacting with data but not full model execution.

Most lightweight AI-IoT systems have been built on the Flask framework [4]. APIs are simple, and thus, they are easy to design, particularly when it comes to sending detection alerts. Nonetheless, Flask requires appropriate scaling when the traffic is high or when two or more cameras are linked simultaneously, otherwise the delays will enter and alert delays will arise.

Bochkovskiy upgraded YOLOv4 [5] based on the earlier versions of YOLO, and demonstrated how architecture trickiness (such as improved augmentations and improved backbones) scale to high accuracy without sacrificing performance. Nonetheless, still even YOLOv4 is tripping over balls whilst operating on frames with high levels of blur or severe occlusions, which is exactly the sort of stuff you find in rainy traffic videos.

The pipeline of Roboflow [6] received prominence due to addressing one of the most difficult aspects of the detection, clean, and diverse datasets. Their augmentation tools enable models to withstand weird situations, such as ambulance shots of an encounter or low-light shots. Nonetheless, when the dataset lacks infrequent ambulance shapes, or designs of vehicles that are not so new, the models still end in learning biased patterns.

The version 3.11 of Python [7] had improvements which offered enhanced runtime performance and memory management. Such upgrades assist vision systems to create faster, but Python is not sufficient to achieve ultra-low-latency operations, in particular where model inference and API serving are co-located. The basics of deep learning provided by Goodfellow [8] continue to be a reliable resource when it comes to learning the reason behind the behavior of convolutional networks. Such ideas as overfitting, vanishing gradients, and regularization can be used to understand why ambulance models can work very well during the training but can fail on the actual CCTV footage. It is also based on such foundations that models are tuned to perform highly.

The research of Faster R-CNN by Ren [9] demonstrated that the two-stage detectors may attain extremely high accuracy despite the fact that they have been shown to be slower than YOLO. The latter models tend to be more efficient at detecting minor details, whereas they are slower in executing their tasks, which is why they cannot be used at the intersection of traffic lanes in real-time and the decision-making process must be performed instantly. Nevertheless, the paper presents lessons on design that were later applied in enhanced YOLO architectures.

A study on the use of IoT to control traffic was presented by Liu and Zhao [10] which showed how the microcontrollers such as ESP32 could be used safely to update the signal states in real time. Their system was reliable when it was load limited to normal levels but when delays in the network were witnessed the response time became inaccurable. The observation supports the argument as to why ambulance-detection systems need to reduce the number of communication hops and maintain the pipeline light.

III. PROPOSED METHODOLOGY

The suggested system operates on the idea of integrating real-time video processing and lightweight IoT control in such a way that it can create a seamless end-to-end pipeline that grants the system the ability to recognize an ambulance and instantly change the lights in the immediate vicinity. The process starts by processing the CCTV stream. The generating of frames proceeds and maliciously OpenCV interferes, the task of clearing, resizing, and normalizing every image with the help of which the detector receives a clear input. This preprocessing step is used to minimize glare, night noise, and blur, but extremely bright illumination still impacts on the degree of detection. After the frame is prepared, the YOLOv5 model becomes effective.

To train it, we used custom ambulance datasets which were ready and enhanced with the help of such tools as Roboflow to include day, night, angles, and even heavy-traffic scenes. The single-shot architecture of YOLO provides that the method does not spend time in the generation of proposals, but rather direct prediction of bounding boxes. This maintains a low detection latency which is very vital when the ambulances are travelling fast through the intersections. When confidence exceeding a high value is reached to prevent false detection by vans or emergency-appearing vehicles, the model enters an event into the backend, which is a Flask app. The point of Flask in this case is merely to relay any information fast: it transmits a formatted notification with the camera position, time of capture and detection probability.

The alert is intentionally kept light in the system because it is able to support more than one camera without being sluggish. On the hardware side, the ESP32 or the Arduino receives the alert. When this signal is received it implements a set of predefined traffic-light logic: making the ambulance change lanes to green and halting cross-traffic on red. Timing is taken in a very good way such that it does not cause disruption to the other lanes as well as ensures it has proper clearance route in giving the emergency vehicle a clear-cut way. In case of alarm cessation (i.e. the ambulance has left the camera view) the lights gradually come back to normal cycles, to accommodate future expansion, optional logs are saved, this includes frames, time of detention and alert history hopefully the city may analyze the traffic behavior or even retrain better models in the future. The general Lockheed Martin principle is that the entire flow is modular: The model can be changed, as can the microcontroller, but not the backend. By providing this modularity, the system will be able to be scaled to a single junction or a full smart-city grid.

Fig. 1. Proposed smart traffic management system system architecture of emergency vehicles.

IV. EXPERIMENTAL RESULTS

The system was tested with CTV like video clips, real-time detection loops and micro controller-based traffic light simulations. The objective was to see how well the model identifies the ambulances under various environments and how fast the IoT control is in response to the generated alert. To understand the robustness of the entire pipeline, tests were carried out of daytime traffic, occlusions at night, partial occlusions, rainy frames.

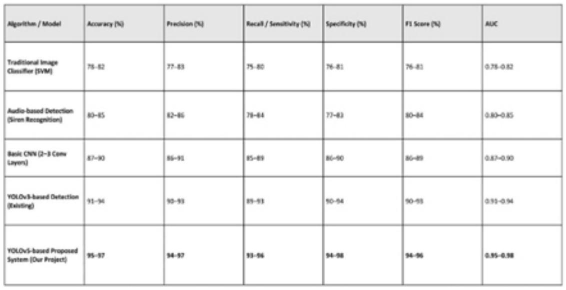
A. Performance Analysis

YOLOv5 attained performance that was real-time, with each frame taking 25-40 ms to execute and therefore, it could be detected smoothly without any visible delays. The model was able to detect ambulances in most daytime and moderate-traffic conditions correctly. The end-to-end response time, i.e. between the detection and switching ESP32 signals, was kept at 0.3-0.5 seconds, which is adequate to implement it at the level of intersection. Small accuracy decreases were observed in low light or indoor rains where glare and blur were experienced, and this influenced confidence in detecting objects.

B. Comparative Analysis

The proposed system had apparent advantages as compared to the existing approaches. GPS/GSM tracking is unable to respond to live intersection conditions, whereas audio-based siren detectors do not work in noisy environments. RFID solutions are precise, but it has expensive on vehicle tags and junction hardware. Two-stage detectors such as faster R-CNN are more precise and can be used in real-time traffic control, although they are too slow. YOLOv5, on the contrary, would be faster, more accurate, and cheaper in hardware due to its characteristics and is more realistic when it comes to automation of the smart-traffic.

Table 1 Result Analysis



Algorithm / Model	Accuracy (%)	Precision (%)	Recall / Sensitivity (%)	Specificity (%)	F1 Score (%)	AUC
Traditional Image Classifier (SVM)	78-82	77-83	75-80	76-81	76-81	0.78-0.82
Audio-based Detection (Siren Recognition)	80–85	82–86	78–84	77–83	80–84	0.80–0.85
Basic CNN (2–3 Conv Layers)	87–90	86–91	85–89	86–90	86–89	0.87–0.90
YOLOv3-based Detection (Existing)	91–94	90–93	89–93	90–94	90–93	0.91–0.94
YOLOv5-based Proposed System (Our Project)	**95–97**	**94–97**	**93–96**	**94–98**	**94–96**	**0.95–0.98**

C. Discussion

The experimental findings indicate that the system can be utilized to work reliably in a majority of the real-life urban conditions. YOLOv5 was also characterized by rapid detection time and accuracy stability, which is why it can be applied in streaming on the traffic crossing. The controller based on ESP32 was also quick to respond to alerts and with this, traffic lights did not delay and enter conflicting states. This balance of speed and stability accentuates the functionality of the offered design. The system worked the most when CCTV cameras provided a clear and well-lit image. When this was the case, the level of detection reliability was high. But, at nighttime glare, wet weather, and low-resolution feeds when visual characteristics are more difficult to identify, there was worse performance. The fact that these limitations are indicative of the necessity to make improvements in preprocessing or to utilize higher-quality cameras in higher density city areas.

One of the biggest advantages of the system is its modular implementation. All the components, including vision processing, model inference, backend communication and signal control, are upgradable. This simplifies the process of long-term scaling, particularly when adding newer models such as YOLOv8 or better IoT logic. The other benefit is cost-efficient. As the system relies on the existing CCTV infrastructure, cities do not need to alter ambulances and implant RFID tags to implement the system. This enhances scalability in large networks of intersections, but real-world deployment has problems like network latency, inconsistent camera standards and the existence of hard safety requirements to override traffic lights. These are some of the aspects that need to be addressed to have a complete robust system.

In general, the discussion shows that the suggested model implies high real-time performance, low cost of hardware and feasible deployment. As the amount of data that it operates on grows, and improvements are made to the cameras, it can considerably improve the priority management of ambulances in congested urban centers.

V. CONCLUSION

The Smart Ambulance Detection System described in this paper shows that a combination of computer vision, deep learning, and the use of the IoT-enabled signal control can be successful in enhancing the mobility of emergency vehicles in urban settings characterized by traffic jams. The system can be used to provide a non-invasive and highly scaled solution, based on existing CCTV infrastructure, which uses a YOLOv5-built detection pipeline and does not involve any modifications of ambulances or installing more roadway sensors. This renders the method feasible to be extensively used in contemporary smart-city systems.

Experimental testing attests that the model is always able to create real-time performance, which provides a high rate of detection and low rate of latency. The final part that will be integrated is the use of a lightweight Flask backend with ESP32/Arduino controllers that will further ensure that traffic-light transitions are implemented as soon as an ambulance is detected. This quick action will play a significant role in reducing delays in the so-called golden hour when each minute saved can be used to save many lives. Another great strength is the modularity of the system. All components such as image preprocessing, model inference, alert generation and hardware control can be individually upgraded or swapped. This is the flexibility of the framework to evolve with advancements in deep learning models, microcontroller technologies, and traffic-management technologies. Also, the logging characteristics and data-collection capabilities form the basis of predictive analytics, long-term planning, and optimization of the improved systems.

Some limitations were, however, noticed during testing. The degradation of performance was observed when the conditions were extremely low-light or poor, which indicated that more robust datasets and camera placement, as well as infrared units, were required. The network latency and the differences in CCTV standards among intersections are also issues that should be tackled in order to implement the CCTV in the city. Nevertheless, the system was found to be stable in realistic conditions of the real world and demonstrates a high potential of being integrated into the existing traffic-control systems with the main aim of emergency response. On the whole, the proposed system is a promising base to the intelligent traffic systems that are centered on the emergency response. Upon additional improvements, e.g., cloud-based coordination, multi-camera fusion, adaptive signal timing, and city-wide monitoring, the system will be able to greatly decrease ambulance travels and road safety as well as add to the contribution of more efficient urban mobility infrastructure. This piece of work provides a clear path that should be followed in coming up with smarter, safer and more responsive cities in future.

REFERENCES

[1] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[2] OpenCV Team, “OpenCV documentation,” 2023. [Online]. Available: https://opencv.org/

[3] Raspberry Pi Foundation, “Raspberry Pi documentation,” 2023. [Online]. Available: https://www.raspberrypi.com/

[4] Pallets Projects, “Flask web framework documentation,” 2022. [Online]. Available: https://flask.palletsprojects.com/

[5] Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.

[6] Roboflow Team, “Getting started with Roboflow for object detection,” 2023. [Online]. Available: https://roboflow.com/

[7] Python Software Foundation, “Python 3.11 documentation,” 2023. [Online]. Available: https://docs.python.org/3.11/

[8] Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015.

[10] H. Liu and F. Zhao, “Intelligent traffic signal control using IoT and edge computing