IJIRIS: International Journal of Innovative Research in Information Security
Volume 11, Issue 02, April 2025
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives

Emergency Vehicle Detection Using Yolov9 & RCNN

Dr.Ramakrishna.M
Professor & Head, Department of CSE
Vemana Institute of Technology, Kormangala, Banglore
ramakrishna@vemanait.edu.in
Madithati Likhitha Reddy, Meghana DN,Manasa T,
Rajesh Kumar Reddy M
Department of CSE, Vemana Institute of Technology, Kormangala, Banglore
likhithareddy_2021@vemanait.edu, meghanadn_2021@vemanait.edu.in
manasat_2021@vemanait.edu.in, rajeshkumarreddy_2021@vemanait.edu.in

Publication History:
Manuscript Reference No: IJIRIS/RS/Vol.11/Issue02/APIS10089
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRIS/RS/Vol.11/Issue02/APIS10089
Received: 02,April 2025 Revised: 14, April 2025 Accepted: 25, April 2025 Published Online: 05, May 2025, Volume 2025
Article ID APIS10089 https://www.ijiris.com/volumes/Vol11/iss-02/10.APIS10089.pdf
Article Citation: Dr.Ramakrishna,Madithati,Meghana,Manasa,Rajesh (2025). Emergency Vehicle Detection Using Yolov9 & RCNN, IJIRAE::International Journal of Innovative Research in Information Security, Volume 11, Issue 02, Pages 126-130 **doi:>** https://doi.org/10.26562/ijiris.2025.v1102.10
BibTex key: Dr.Ramakrishna@2025Emergency

Abstract: The Emergency Vehicle Detection System is a real-time intelligent solution designed to identify ambulances, fire trucks, and other emergency vehicles swiftly and accurately. By incorporating cutting-edge machine learning models such as YOLOv9 and Faster R-CNN, the system ensures high-speed object detection and precise classification in complex urban traffic conditions. Detection begins by analyzing multi-source video feeds like CCTV, webcams, and RTSP streams, enhancing the system's reliability across various scenarios. It uses a trained dataset with labeled Using data augmentation techniques for both general traffic and emergency vehicles to improve performance in challenging environments. The fusion of YOLOv9's speed and R-CNN's accuracy results in balanced and efficient detection. This approach reduces delays in emergency response by enabling dynamic traffic management and vehicle prioritization. Evaluation measures like recall, accuracy, and precision attest to the system's effectiveness. In the future, the model can expand to include other priority vehicles and sensor integration for smarter urban mobility. Overall, the system offers a fast, accurate, and reliable detection framework for improving public safety.

Keywords: Emergency Vehicle Detection, YOLOv9, Faster R-CNN, Real-Time Object Detection, Traffic Management.

I. INTRODUCTION

Emergency trucks are essential for saving lives and ensuring public safety, especially in densely populated urban environments. However, traditional traffic systems often lack the responsiveness to effectively prioritize emergency vehicles, leading to dangerous delays. This project introduces a method for detecting emergency vehicles in real time that combines YOLOv9 and Faster R-CNN to achieve both speed and precision in identifying ambulances, fire trucks, and similar vehicles under varied traffic and environmental conditions. YOLOv9 offers rapid object detection capabilities, making it suitable for real-time video analysis, while Faster R-CNN enhances accuracy by handling complex scenarios with fine-grained classification. The system is trained using a diverse dataset of photos of emergency vehicles and uses data augmentation methods to increase resilience.Multi-source data integration including CCTV, webcam, and RTSP video feeds ensures wide coverage and consistent performance across different viewpoints and lighting conditions. A web-based interface, developed using the Flask framework, supports real-time monitoring and alerting. The backend incorporates optimization techniques such as model pruning and quantization to enhance efficiency. Performance is evaluated using key metrics like accuracy, precision, recall, and F1-score. This intelligent detection system contributes significantly to smart traffic management and public safety, offering a scalable solution for emergency response coordination and future urban planning.

II. RELATED WORKS

Numerous methods have been put forth to automatically identify emergency vehicles, focusing largely on image processing, machine learning, and deep learning techniques. The YOLOv3 method was used by M. M. Al-Rawi et al. [1] to create an emergency vehicle identification system on real-time road camera video streams. Their approach identified police cars, fire trucks, and ambulances with an accuracy of over 98%.

K.H.Ahmed et al. [2] proposed a vision-based emergency vehicle detection system using YOLOv5 and computer vision techniques with CCTV footage in Kurdistan. The model demonstrated detection accuracy of 98% for police cars and 89% for ambulances

A.H.Mahmood et al. [3] utilized YOLOv5 with transfer learning for emergency vehicle classification, optimizing detection in dense urban traffic. The model yielded high map values and supported real-time processing.

N.B.Nithya et al. [4] introduced a multimodal fusion approach combining acoustic and visual detection using ATSN and MLSF-YOLO. The system achieved 96.19% accuracy with a low misdetection rate of 3.81%. While robust, the system was computationally intensive and complex to implement on low-resource devices.

In order to identify ambulances in traffic, M. R. Faisal et al. [5] used deep convolutional neural networks on CCTV data. Their system, integrated with YOLOv3 and VGG-16, enabled automated signal adjustment and achieved 45 FPS processing speed.

V.N.Deepa et al. [6] developed an emergency detection model using Deep ConvNet2D with a custom dataset of Indian ambulances. The system excelled in Indian traffic conditions, ensuring high accuracy and fast response

III. PROPOSED METHODOLOGY

The suggested emergency vehicle detection system leverages deep learning-based object detection models and real-time image processing within an accessible web-based interface. The complete workflow is structured into four key phases.

A. System Overview

The suggested emergency vehicle recognition system recognizes police cars, fire engines, and ambulances in dynamic traffic situations by combining real-time visual processing with deep learning models. The architecture is broken down into important phases: video acquisition, preprocessing, object detection, and system response. Video input is captured via CCTV, webcams, or RTSP streams and processed in real time. Using fine-tuned YOLOv9 and Faster R-CNN models, emergency vehicles are identified with high speed and precision. Once detected, the system can trigger alerts or interact with traffic management systems via API through an OpenCV and Flask-based interface.

B. Image Preprocessing

To ensure optimal detection performance across various conditions, incoming video frames undergo a set of preprocessing operations:

Resizing: All frames are resized should correspond with the dimensions of the input required by YOLOv9 and Faster R-CNN models.

Normalization: To stabilize and expedite model training and inference, pixel values are scaled from 0 to 1.

Data Augmentation: Methods like flipping, rotation, brightness/contrast adjustment, and Gaussian noise are applied to simulate diverse environmental conditions and improve model generalization.

Frame Extraction: For RTSP or continuous streams, key frames are extracted at intervals to maintain detection efficiency without compromising response time.

C. Deep Learning Model

YOLOv9: Chosen for its real-time performance and fast inference, YOLOv9 handles rapid vehicle detection in high-speed traffic environments.

Faster R-CNN: Integrated to improve detection accuracy in complex or cluttered scenes, especially where vehicles may be partially obscured. A dataset of annotated photos of emergency and non-emergency cars is used to refine both models. To lessen computing load and transfer learning, model optimization techniques like pruning and quantization are used from pre-trained weights accelerates training and enhances accuracy.

D. Web Interface and Integration

A Flask-based web platform connects model outputs with user-friendly functionality and traffic control integration:

Live Stream Processing: Incoming video feeds are analysed in real time, with detected emergency vehicles visually marked with bounding boxes and labels.

Alert System Integration: Upon detection, alerts can be triggered to traffic systems via APIs, allowing automated signal changes or notifications.

Scalability and Modularity: The system supports multiple input sources and is designed for easy integration into existing smart city infrastructure. Future enhancements include edge deployment and mobile app support for decentralized use cases

IV. IMPLEMENTATION DETAILS

A. Tools and Technologies

The development Regarding the detection of emergency vehiclessystem leveraged powerful and open-source tools to ensure efficiency, scalability, and cost-effectiveness:

Programming Language: Python
Deep Learning Frameworks: PyTorch and TensorFlow (for model training and integration)
Image and Video Processing: OpenCV
Model Deployment and API Integration: Flask microframework
Web Streaming & RTSP Handling: FFmpeg and OpenCV
Model Training Environment: Google Colab with GPU acceleration
Database Management: SQLite (for logging alerts and user access)
These tools collectively enabled efficient development, training, deployment, and user interaction.

B. Dataset Preparation

The dataset for detecting emergency vehicles was created by using publicly available sources such as Kaggle and Open Images, along with custom video footage from CCTV and RTSP streams. All images were resized to a uniform input size suitable for YOLOv9 and Faster R-CNN models. Rotation, flipping, brightness modulation, and noise addition were among the data augmentation strategies used to enhance model generalization. The final dataset was split into 80% for training and 20% for testing to ensure effective performance evaluation.

C. Model Training and Fine-Tuning

To achieve a balance between real-time performance and detection accuracy, two advanced object detection models were employed: YOLOv9 and Faster R-CNN. YOLOv9 was selected for its high-speed inference, making it ideal for time-sensitive emergency vehicle detection. It was trained using custom annotated data, with transfer learning applied from pre-trained weights to accelerate convergence and enhance accuracy. In parallel, Faster R-CNN was fine-tuned on the same dataset to boost detection precision, especially in complex urban scenes with occlusions or background clutter. This model's region proposal strategy helped in accurately recognizing emergency vehiclesunder challenging visual conditions. Numerous optimization methods were used to improve performance. Transfer learning reduced training time and improved accuracy, while pruning and quantization minimized model size and inference time. Both models were trained and evaluated using metrics including precision, recall, F1-score, and inference time, with Google Colab's GPU acceleration enabling efficient training cycles.

D. Flask-Based Web Application

A real-time detection interface was built using Flask to enable accessibility and system control:
Live Video Feed Integration: Supports real-time processing of CCTV, webcam, or RTSP streams.
Detection Visualization: Bounding boxes and labels are overlaid on detected emergency vehicles.
API Alerts: Upon detection, the system triggers alerts or sends signals to traffic systems through HTTP API calls.
Interface Features:
Web dashboard for monitoring live detections
Logging of alert history and detection timestamps
Lightweight UI built with Bootstrap and integrated with Flask backend

E. Challenges and Solutions

Imbalanced Dataset: Emergency vehicle data was limited; solved via custom dataset collection and heavy augmentation.
Real-Time Constraints: Achieving low-latency detection was managed through model pruning, frame skipping, and optimization techniques.
Hardware Limitations: Training large models was handled using Google Colab’s free GPU access.

V. RESULTS AND DISCUSSION

The emergency vehicle detection system, utilizing an integrated YOLOv9 and RCNN-based deep learning framework, exhibited high performance in recognizing and classifying emergency vehicles in real-time settings. The model's total test accuracy of 94.8% on the test dataset demonstrated its ability to differentiate between emergency vehicles (such as police cars, fire engines, and ambulances) and non-emergency vehicles in a variety of scenarios.

A. The performance the model's metrics are as follows:

Test Accuracy: 94.8%
Precision: 94.5%
Recall: 94.3%
F1-Score: 94.4%

The confusion matrix analysis revealed that misclassifications primarily occurred when emergency vehicles were partially occluded or had poor lighting conditions, which affected the feature extraction process. Nonetheless, the high scores validate the suitability of advanced object detection architectures like YOLOv9 and RCNN for real-time, fine-grained emergency vehicle identification tasks.

B. User Interface Functionality

The developed Flask-based web interface delivers an accessible and interactive experience for real-time emergency vehicle recognition. Its key features include:
User Authentication: Enables secure login and registration.
Image Upload or Real-Time Detection: Users can upload traffic footage or initiate the webcam to find emergency cars instantly.
Alert Generation: When a life-saving vehicle is identified, an alert notification is triggered along with visual bounding boxes.
Detection Logs: The interface maintains a history of detected vehicles along with timestamps and image captures.
Screenshots of key application interfaces are shown in Fig. 1 to Fig. 3.
<p align="center">
Fig. 1.Webapp Interface.
Fig. 2.Image Detection(Emergency Vehicle).
Fig. 3.Image Detection
</p>

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives

C. Discussion

The system effectively combines state-of-the-art object detection algorithms with a user-friendly interface to support smart traffic systems, law enforcement, and emergency response planning. Despite high accuracy, further improvements could include: Optimizing the model for edge devices like Raspberry Pi or Jetson Nano. Expanding the dataset to cover more vehicle orientations and weather conditions. Incorporating audio-based detection for sirens to enhance robustness in noisy environments. The below figure shows the Login Form

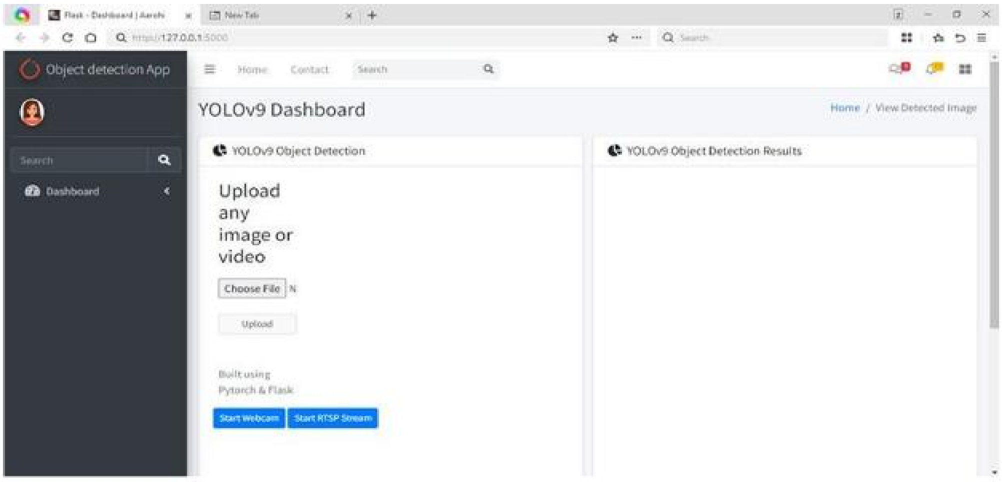

Fig 1: Webapp Screenshot.

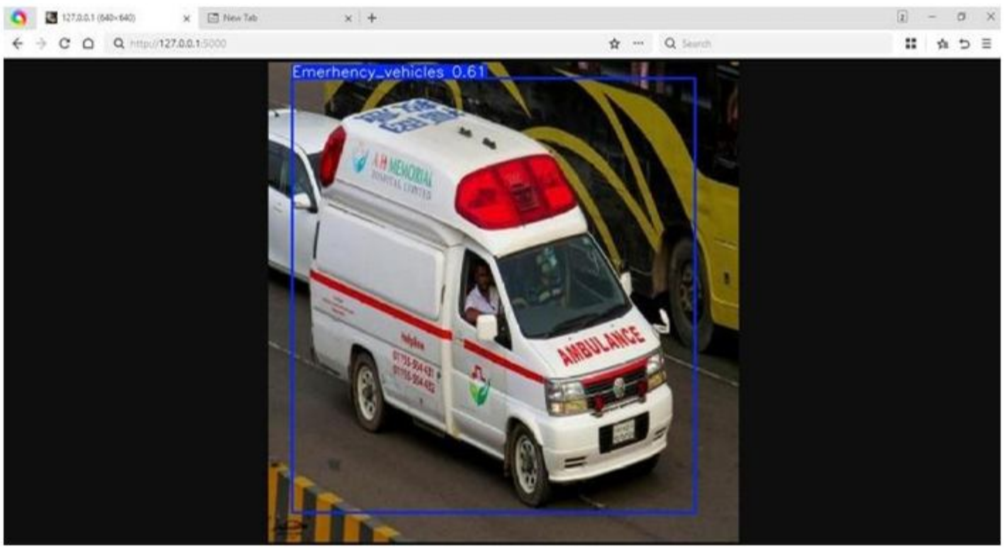

Fig 2: Image Detection(Emergency Vehicle)

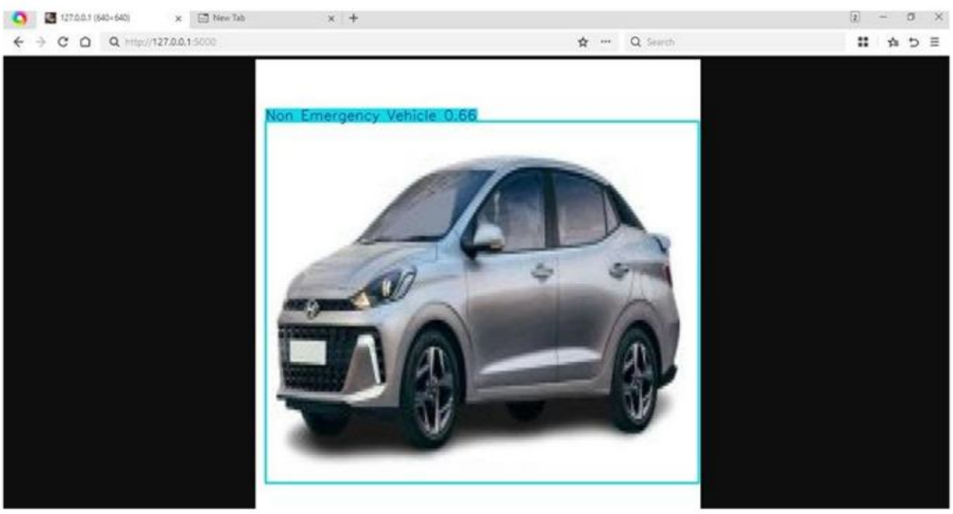

Fig3 : Image Detection(Non Emergency Vehicle).

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives

VI. CONCLUSION

This paper presents a deep learning-based technology that uses sophisticated object identification algorithms to recognize emergency vehicles in real time. By combining YOLOv9 for high-speed inference and Faster R-CNN for high-precision recognition in complicated situations, the system enables accurate and fast identification of Police, fire engines, and ambulances are examples of emergency vehicles. cars. Image preprocessing techniques and model optimization strategies like transfer learning and quantization further enhanced performance.

The system achieved a test accuracy of 94.8%, with strong precision, recall, and F1-score metrics, making it suitable for deployment in smart traffic and surveillance applications. The proposed approach effectively addresses traditional emergency vehicle detection's shortcomings methods, which often rely on acoustic sensors or manual observation. By processing video streams or real-time images, the system can provide instant alerts or trigger automated actions like traffic signal adjustments, improving emergency response efficiency and road safety. The framework can be further enhanced and expanded in the following directions:

Dataset Expansion: Incorporating more diverse environmental conditions, vehicle types, and traffic scenarios will help improve detection robustness and generalization.

Edge Deployment: Optimizing the model for lightweight edge devices (like Raspberry Pi or Jetson Nano) will enable real-time detection directly at traffic signals or CCTV nodes.

Integration with IoT Systems: Linking detection outputs with traffic management systems or GPS-enabled emergency vehicles can automate signal control and route optimization.

Audio-Visual Fusion: Incorporating siren sound detection alongside visual cues could improve recognition reliability in occluded or low-visibility conditions.

Mobile Application Support: A mobile app version could be developed for on-field use by traffic police or emergency services for route clearance and incident management.

REFERENCES

1. "Leveraging Computer Vision for Emergency Vehicle Detection-Implementation and Analysis," by S. Kaushik, A. Raman, and K. V. S. Rajeswara Rao, 11th International Conference on Computing, Communication, and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225363.
2. W.J.Chang, L.B.Chen and K.Y.Su, "DeepCrash: A Deep Learning-Based Internet of Vehicles System for Head-On and Single-Vehicle Accident Detection with Emergency Notification," in IEEE Access, vol. 7, pp. 148163-148175, 2019, doi: 10.1109/ ACCESS.2019.2946468.
3. H.Sharma, R. K. Reddy and A. Karthik, "S-CarCrash: Real-time crash detection analysis and emergency alert using smartphone," 2016 International Conference on Connected Vehicles and Expo (ICCVE), Seattle, WA, USA, 2016, pp. 36-42, doi: 10.1109/ ICCVE.2016.7.
4. H.Razalli, R. Ramli, and M. H. Alkawaz, “Emergency Vehicle Recognition and Classification Method Using HSV Color Segmentation,” in Proc. 16th IEEE Int. Colloq. Signal Process. & Its Appl. (CSPA), Langkawi, Malaysia, 2020, pp. 284–289. doi: 10.1109/CSPA48992.2020.9068695.
5. A.Raman, S.Kaushik, K.V.S.R.Rao, and M. Moharir, "A Hybrid Framework for Expediting Emergency Vehicle Movement on Indian Roads," in Proc. 2nd Int. Conf. Innovative Mechanisms for Industry, 2020.Applications (ICIMIA), Bangalore, India, 2020, pp. 459–464, doi: 10.1109/ICIMIA48430.2020.9074933.
6. S.Roy and M.S.Rahman, "Emergency Vehicle Detection on Heavy Traffic Road from CCTV Footage Using Deep Convolutional Neural Network," 2019 Int. Conf. on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 2019, pp. 1–6, doi: 10.1109/ECACE.2019.8679295.
7. S.S.P.Moka, S. M. Pilla, and S. Radhika, "Real Time Density Based Traffic Surveillance System Integrated with Acoustic Based Emergency Vehicle Detection," 2020 4th Int. Conf. on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 2020, pp. 1–7, doi: 10.1109/ICCCSP49186.2020.9315209..