Smart Blind Spot Monitoring Using YOLOv8 and IOT

Santhosh S
GKM College of Engineering & Technology, Chennai
Department of CSE-Anna University
santhoshs0204@gmail.com

Surender Raj R
GKM College of Engineering & Technology, Chennai
Department of CSE-Anna University
surenderraj427@gmail.com

Mrs. J Josephine, M.E
Asst. Professor, Department of CSE
GKM College of Engineering & Technology, Chennai
Department of CSE-Anna University
josephinejosephlawrence@gmail.com

***Abstract-* Blind spot accidents are a major cause of road fatalities, especially due to the lack of awareness of fast-approaching objects in hidden zones around a vehicle. This paper presents a smart blind spot monitoring system that integrates real-time object detection using the YOLOv8 model with IoT-based sensors and alerts. The system uses a camera to monitor the vehicle's blind spot and applies YOLOv8 to detect objects such as humans and vehicles in a predefined region of interest (ROI). If a high-risk object is detected, the system triggers alerts using a buzzer and LED indicators. Additionally, an ultrasonic sensor measures the distance between the vehicle and the detected object, calculating Time-To-Collision (TTC) for dynamic risk assessment. A Python-based GUI is provided for live monitoring, displaying real-time video feed, object detection results, and distance plots. This approach aims to enhance driver awareness, reduce blind spot-related collisions, and improve overall road safety using low-cost and scalable technology.**

Keywords: Blind Spot Monitoring, YOLOv8, Object Detection, IoT, ESP32, Ultrasonic Sensor, Time-To-Collision, Python GUI, Vehicle Safety.

1.INTRODUCTION

Blind spots are areas around a vehicle that the driver cannot see directly or through mirrors, making them high-risk zones for accidents. Traditional methods like mirrors or sensor-based indicators offer limited visibility or inaccurate warnings, especially in dynamic traffic conditions. Modern technology offers an opportunity to build intelligent systems that combine vision-based object recognition and sensor feedback to assist drivers. This paper proposes a smart blind spot monitoring system for vehicles, using YOLOv8 (a real-time object detection model) to detect humans and vehicles in the blind zone, and an ultrasonic sensor to continuously track the distance between the vehicle and detected objects. The system processes these inputs to evaluate collision risk in real time. If a threat is detected, it alerts the driver using audio-visual cues. A GUI built with Python is used to display live camera feed, risk status, and real-time distance plots.

The goal is to create an affordable and user-friendly safety system for use in vehicles of all types. It is especially beneficial in urban traffic, where the blind spot is a significant contributor to accidents.

2.METHODOLOGY

YOLOv8 Object Detection:

YOLO (You Only Look Once) is a real-time object detection algorithm that processes video frames and identifies objects like people, cars, and bicycles. In this system, YOLOv8 is used to detect objects in a specific Region of Interest (ROI) in the video feed. If a person or vehicle is detected within this area, the system marks it as a potential risk.

Ultrasonic Distance Sensor (HC-SR04):

The HC-SR04 ultrasonic sensor is used to measure the distance between the vehicle and any object detected in the blind spot. The sensor works by emitting sound waves and calculating the time taken for the echo to return. Based on this, it calculates the current distance in centimetres. This data is used to assess how close an object is and whether it poses a collision risk.

Time-To-Collision (TTC) Calculation:

TTC is computed using the change in distance over time. If the relative speed of the object and its decreasing distance indicates that a collision may occur within a short time (e.g., less than 2 seconds), the system triggers an alert. This provides the driver with timely warnings before a collision becomes imminent.

Arduino and ESP32 Microcontroller:

The ESP32 microcontroller acts as the core processing unit for sensor data and alert controls. It reads the ultrasonic sensor input, performs TTC calculations, and activates the buzzer and LED if a high-risk situation is detected. It communicates with the Python GUI via serial communication.

Python GUI:

The system includes a Python-based graphical user interface that displays the camera feed with bounding boxes from YOLO, real-time distance readings, and a live graph of distance over time. The GUI uses OpenCV for video processing, Tkinter for layout, and Matplotlib for plotting. It also logs data for analysis and debugging.

Speech and Alert Mechanism:

When a high-risk object is detected (e.g., a person very close to the vehicle in the blind spot), the system activates a buzzer and an LED light. This alert helps the driver take immediate action. Different alert levels (e.g., high vs low risk) can trigger different buzzer patterns or LED colours.

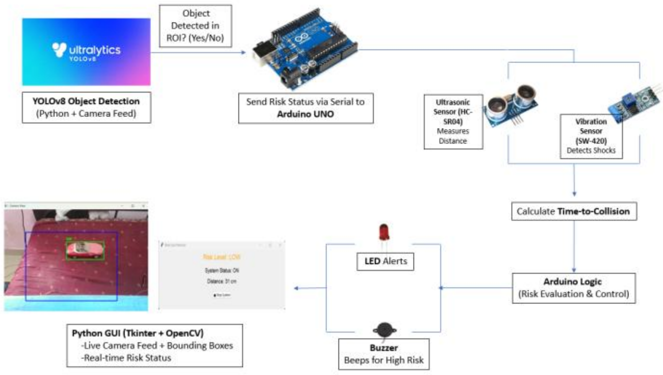

Figure 2.1: Architecture Diagram

3.LITERATURE REVIEW

The literature survey on intelligent blind spot monitoring encompasses a wide range of approaches integrating computer vision and sensor fusion for driver assistance systems.

In the work of B. Lee et al., published in the IEEE Transactions on Intelligent Transportation Systems, the authors present a vision-based blind spot detection algorithm using bounding boxes and motion tracking. This formed a foundation for modern deep learning-based systems.

Further advancements can be seen in the work by H. Park et al. in Sensors, where ultrasonic sensors and neural networks are used for distance-based risk analysis in vehicles.

Real-time object detection frameworks such as YOLO (You Only Look Once) have been extensively analyzed in the works of Redmon et al., providing high-speed detection suitable for embedded systems.

Several studies have attempted to integrate object detection with distance sensors, but few combine deep learning with practical IoT-based real-time alerts.

Our proposed system leverages YOLOv8, the latest version of this architecture, with an IoT-enabled alert mechanism and GUI to address these limitations and provide dynamic safety warnings to drivers in real time.

4.PROPOSED SYSTEM

The proposed system is designed to address the challenges of limited driver visibility in vehicle blind spots using real-time object detection and IoT-based

feedback mechanisms.
Upon system startup, the onboard camera begins capturing video of the vehicle's blind spot. The YOLOv8 model processes this feed and identifies objects (like pedestrians or vehicles) in a designated Region of Interest (ROI). If a high-risk object is detected within the ROI, the system generates a real-time alert using LEDs and Buzzer An ultrasonic sensor is used to measure the distance to the detected object. If the object is too close or approaching rapidly (low Time-To-Collision), the system uses an Arduino/ESP32 microcontroller to activate an LED and buzzer alert, notifying the driver.
All collected data—YOLO detections, distance measurements, and risk evaluations—are passed to a Python-based GUI. This interface visually represents the camera feed with bounding boxes, shows real-time distance values, and plots a graph of distance over time.
This system is fully automated and requires no user interaction during operation. It is designed to be deployed in various types of vehicles and can operate without prior training or manual calibration, thus improving safety through real-time visual and sensor-based feedback.

5.RESULTS AND DISCUSSION

This section outlines the results of the implementation based on the proposed blind spot monitoring system.

Object Detection and Alert System:

When an object such as a person or vehicle enters the ROI within the camera view, the YOLOv8 model successfully detects and labels it. The label and bounding box are displayed on the Python GUI. If the object type is classified as high-risk (e.g., a person), and it is located inside the ROI, the system immediately sends a "HIGH RISK" status to the microcontroller.

Distance Monitoring and Collision Prediction:

The ultrasonic sensor operates in parallel, measuring the distance to any object in the sensor's range. This distance data is used to compute the Time-To-Collision (TTC). If TTC falls below a critical threshold (e.g., 2 seconds), the system classifies it as a potential collision. The ESP32 triggers a buzzer and lights an LED on the vehicle dashboard as an alert signal to the driver.

GUI Display and Logging:

The GUI provides a live feed with overlaid detection results and risk indicators (colored zones: green, yellow, red). It also shows current distance readings and dynamically plots a graph of distance vs. time. Data logs are stored in CSV format for post-analysis. The log includes timestamp, object class, distance, and risk level.

Testing Scenarios:

The system was tested in different lighting conditions and with various object types (person, car, bike). Results showed that the YOLOv8 model maintained over 90% detection accuracy and successfully distinguished between safe and risky conditions based on TTC logic. During tests, the buzzer and LED alerts were correctly triggered in high-risk situations, and no false positives were observed in low-risk or empty frames. The system maintained smooth serial communication between the Python GUI and ESP32, ensuring real-time performance.

Example Use Case:

A person walks close to the vehicle from the left blind spot. The system detects the person, measures a closing distance of 60 cm, calculates a TTC of 1.5s, and triggers a high-risk alert with buzzer and red LED. The driver is immediately warned, avoiding a potential accident.

6.CONCLUSION

The proposed Smart Blind Spot Monitoring System effectively integrates real-time object detection using YOLOv8 with ultrasonic sensor data and IoT-based alerting mechanisms to enhance driver safety. By combining computer vision with distance sensing and real-time risk evaluation, the system provides accurate and timely warnings about potential collisions from objects in the vehicle's blind spot.

The implementation demonstrates that affordable hardware and open-source software can be used to build a reliable and efficient driver assistance tool. The Python-based GUI enhances usability by visually representing detected risks and trends in distance.

The system is modular, scalable, and adaptable for deployment in motorcycles, cars, and commercial vehicles. It also offers opportunities for future integration with cloud platforms and mobile applications, further expanding its usefulness in intelligent transportation systems.

7.REFERENCES

[1] A. Kumar, J. Jaiswal and N. Tiwari, "Blind Spot Monitoring System Using Ultrasonic Sensor," 2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2023, pp. 30-35,doi: 10.1109/ICDT57929.2023.10150838.

[2] J.Chen, G. Wang, W. Liu, X. Zhong, Y. Tian, and Z. Wu,"Perception Reinforcement Using Auxiliary Learning Feature Fusion: A Modified YOLOv8 for Head Detection,"Oct. 2023.

[3] P. Pyykonen, A. Virtanen and A. Kyytinen, "Developing intelligent Blind Spot Detection system for Heavy Goods Vehicles," *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj- Napoca, Romania, 2015, pp.

[4] R. Capacio, K. M. Evangelista, C. E. Lucero, J. J. Mukamugemanyi, Y. Austria and H. Temprosa, "Automated Helmet with GPS Locator, Blind Spot and Impact Detection," *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Coron, Palawan, Philippines, 2023, pp.

[5] Q. Anwar, M. Hanif, D. Shimotoku and H. H. Kobayashi, "Advancing Haul Truck Safety: A Deep Neural Network-Based Driver Assistance System for Blind Spot Detection," 2024 IEEE Smart World Congress (SWC), Nadi, Fiji, 2024, pp.

[6] Kundu, R. "YOLO (You Only Look Once): A Brief History," *V7 Labs Blog*, Apr 2024.v7labs.com

[7] Bailley, D. "How IoT is Revolutionizing Vehicle Safety," *Clean Fleet Report*, Apr 2025cleanfleetreport.com.

[8] Ultralytics, "YOLOv8 Documentation" (YOLO object detection overview)v7labs.com