© December 2025 | IJIRT | Volume 12 Issue 7 | ISSN: 2349-6002

Object Detection and Localization using YOLO Algorithm

Usha H N[1], Bhumika N N[2], Kruthi R[3], Madhu[4], Rakshitha Y U[5]
[1]Associate Professor Department of E&CE APS College of Engineering, Bangalore, India
[2,3,4,5]Department of E&CE APS College of Engineering, Bangalore, India

Abstract—Road safety continues to be a critical concern worldwide, with thousands of accidents occurring due to delayed driver response and environmental hazards. Advanced Driver Assistance Systems have emerged as vital technologies to mitigate these risks through continuous environmental monitoring and intelligent decision-making. This paper presents the design and implementation of a cost-effective ADAS prototype that leverages the YOLO object detection algorithm integrated with an ESP8266 microcontroller-based robotic vehicle platform. The system incor- porates an ultrasonic sensor for proximity detection and utilizes IoT connectivity for remote monitoring and real-time alerts. By combining deep learning techniques with embedded hardware, this research demonstrates a practical approach to implementing collision avoidance, obstacle recognition, and automated warning systems. The proposed solution offers significant advantages in terms of affordability, scalability, and deployment feasibility for intelligent transportation applications.

Index Terms—ADAS, YOLO Algorithm, Object Detection, ESP8266, IoT, Ultrasonic Sensor, Collision Avoidance, Embedded Systems, Autonomous Navigation

I. INTRODUCTION

The rapid increase in vehicular traffic has intensified the need for intelligent safety systems that can assist drivers in making timely decisions. Traditional driver assistance relied primarily on passive safety features, but modern technology enables active intervention through sensor fusion and artificial intelligence. Advanced Driver Assistance Systems represent a paradigm shift in automotive safety, providing real-time environmental awareness and proactive hazard mitigation

Computer vision techniques have become fundamental to ADAS development, with object detection serving as a core capability. Among various algorithms available, YOLO has gained prominence due to its ability to perform real-time de- tection with high accuracy. Unlike traditional sliding-window approaches or region-based convolutional neural networks,
YOLO processes the entire image in a single forward pass, making it exceptionally suitable for time-critical applications. The proliferation of low-cost microcontrollers and IoT plat- forms has democratized access to sophisticated embedded sys- tems. These developments enable researchers and developers to prototype intelligent systems without substantial capital investment. This project capitalizes on these technological advances by implementing a comprehensive ADAS solution using widely available components.

The primary objective of this work is to demonstrate that effective driver assistance features can be achieved through the integration of state-of-the-art algorithms with affordable hard- ware. The system architecture combines the ESP8266 Wi-Fi module for communication and control, ultrasonic sensors for distance measurement, and a camera module for visual input to the YOLO detection pipeline. This multi-modal approach ensures robust obstacle detection under various environmental conditions.

By enabling remote monitoring through IoT connectivity, the system extends beyond traditional ADAS implementations to provide data logging, pattern analysis, and cloud-based alert mechanisms. Such capabilities are particularly valuable for fleet management, driver behavior analysis, and continuous system improvement through data-driven insights.

II. LITERATURE REVIEW

Recent advances in object detection algorithms have signif- icantly influenced ADAS development. Traditional methods including Haar cascades, HOG features, and sliding window classifiers provided foundational approaches but suffered from computational complexity and limited accuracy. The intro- duction of deep learning, particularly Convolutional Neural
Networks, revolutionized computer vision applications by en- abling automatic feature extraction and hierarchical represen- tation learning.
The evolution of YOLO architecture represents a mile-

stone in real-time object detection. The original YOLOv1, introduced in 2016, demonstrated that detection could be formulated as a regression problem, eliminating the need for complex pipelines. Subsequent versions incorporated improve- ments such as batch normalization, anchor boxes, and multi-scale feature extraction. YOLOv3 introduced the Darknet-53 backbone and feature pyramid networks, significantly enhanc- ing detection accuracy for objects at different scales.

Research in embedded ADAS implementations has explored various hardware platforms. Studies utilizing Raspberry Pi, NVIDIA Jetson, and Arduino-based systems have demon- strated feasibility across different performance and cost points. The ESP8266 microcontroller, while originally designed for IoT applications, has shown promise in lightweight control and communication tasks when paired with more powerful processors for intensive computations.

Ultrasonic sensors remain popular for proximity detection due to their reliability, low cost, and ease of integration. Multiple studies have validated their effectiveness in collision warning systems, particularly for low-speed scenarios and parking assistance. Sensor fusion approaches that combine ultrasonic measurements with camera-based detection provide complementary information, improving overall system robust- ness.

IoT integration in vehicular systems has opened new possi- bilities for connected vehicle ecosystems. Cloud-based archi- tectures enable remote diagnostics, over-the-air updates, and aggregated data analysis. Research has shown that IoT-enabled ADAS can contribute to broader intelligent transportation systems, facilitating vehicle-to-infrastructure communication and cooperative collision avoidance.

Studies comparing different YOLO versions for automotive applications have highlighted the trade-offs between speed and accuracy. While newer versions offer improved detection per- formance, they also require more computational resources. For resource-constrained embedded systems, optimization tech- niques such as model pruning, quantization, and knowledge distillation have been proposed to balance performance re- quirements with hardware limitations.

III. SYSTEM ARCHITECTURE

Hardware Components

The proposed system integrates several key hardware mod- ules to achieve comprehensive ADAS functionality. Each com- ponent was selected based on cost-effectiveness, availability, and compatibility with the overall architecture.

ESP8266 Microcontroller: This Wi-Fi-enabled microcon- troller serves as the primary communication and control hub. Operating at 80 MHz with integrated TCP/IP protocol stack, it manages sensor data acquisition, motor control signals, and wireless communication. The ESP8266's low power consump- tion and compact form factor make it ideal for mobile robotic applications.

Figure1: ESP8266 Microcontroller

L293D Motor Driver: This dual H-bridge motor driver IC provides bidirectional control for DC motors. With a maximum current rating of 600 mA per channel and built-in protection diodes, it ensures reliable motor operation. The driver receives PWM signals from the ESP8266 to control vehicle speed and direction.

Figure2: L293D Motor Driver

Buck Converter: A DC-DC step-down converter maintains stable voltage supply to sensitive electronic components. This regulation is critical for preventing damage from voltage fluc- tuations and ensuring consistent system performance across varying battery levels.

Figure3: Buck Converter

Ultrasonic Sensor: The HC-SR04 ultrasonic sensor pro- vides distance measurements from 2 cm to 400 cm with accu- racy within 3 mm. Operating on the echo-location principle, it emits 40 kHz ultrasonic pulses and measures the time-of-flight to calculate obstacle distance. The sensor's wide detection angle of approximately 15 degrees ensures adequate coverage for forward-facing obstacle detection.

Figure3: Ultrasonic Sensor

Buzzer: A piezoelectric buzzer generates audible alerts when potential collisions are detected. The device operates at 5V and produces sound levels sufficient for driver notification in typical automotive noise environments.

Figure5: Buzzer

Power Supply: A rechargeable lithium-polymer battery provides portable power for the entire system. Battery capacity is selected to support extended operation during testing and demonstration scenarios.

Figure6: Power Supply

Software Architecture

The software framework comprises three primary layers: sensor interface, processing logic, and communication protocol implementation.

The sensor interface layer handles data acquisition from the ultrasonic sensor at regular intervals. Distance measurements are converted from echo time to centimeters using the formula: distance = duration×0.034/2, where duration represents the round-trip time in microseconds.

The processing layer implements the core ADAS logic. Camera frames are preprocessed and fed to the YOLO detection model, which identifies objects and their bounding boxes. The detection results are analyzed to assess potential collision risks based on object proximity and trajectory. Simultaneously, ultrasonic sensor readings provide continuous distance monitoring for immediate obstacle awareness.

The communication layer manages IoT connectivity through the ESP8266's Wi-Fi capabilities. MQTT protocol or HTTP requests facilitate data transmission to cloud platforms or mobile applications. This enables remote monitoring, alert notification, and data logging for subsequent analysis.

YOLO Algorithm Integration

YOLO's architecture divides input images into a grid and predicts bounding boxes and class probabilities for each grid cell. This approach enables simultaneous detection of multiple objects in a single evaluation. For this implementation, a lightweight YOLO variant is employed to balance detection accuracy with computational constraints.

The detection pipeline begins with image acquisition from a camera module. Each frame undergoes preprocessing including resizing, normalization, and color space conversion. The processed image passes through the YOLO network, which consists of convolutional layers for feature extraction and fully connected layers for bounding box prediction.

Post-processing involves non-maximum suppression to eliminate redundant detections and thresholding based on confidence scores. Detected objects are classified into relevant categories such as vehicles, pedestrians, and static obstacles. The system prioritizes high-threat objects based on their proximity and trajectory relative to the vehicle's path.

For real-time performance on embedded hardware, model optimization techniques are applied. These include reducing input resolution, limiting the number of detection classes, and potentially offloading intensive computation to more powerful processors while using the ESP8266 for control and commu- nication tasks.

IV. SYSTEM IMPLEMENTATION

Circuit Design

The circuit architecture connects all hardware components through appropriate interfaces and power distribution net- works. The ESP8266 operates at 3.3V logic levels, requiring voltage level shifting when interfacing with 5V components. Pull-up resistors ensure stable signal levels on I/O pins, while decoupling capacitors filter power supply noise.

The ultrasonic sensor connects to GPIO pins configured for trigger pulse generation and echo detection. The trigger pin sends a 10-microsecond high pulse to initiate measurement, while the echo pin outputs a pulse width proportional to measured distance. Motor driver inputs receive PWM signals from the ESP8266 for speed control and digital signals for direction control. The driver's output stage connects directly to DC motors with appropriate current capacity. Flyback diodes protect against inductive voltage spikes during motor switching.
The buzzer connects through a transistor driver stage to provide adequate current amplification. This configuration prevents excessive current draw from the microcontroller while ensuring sufficient buzzer activation.

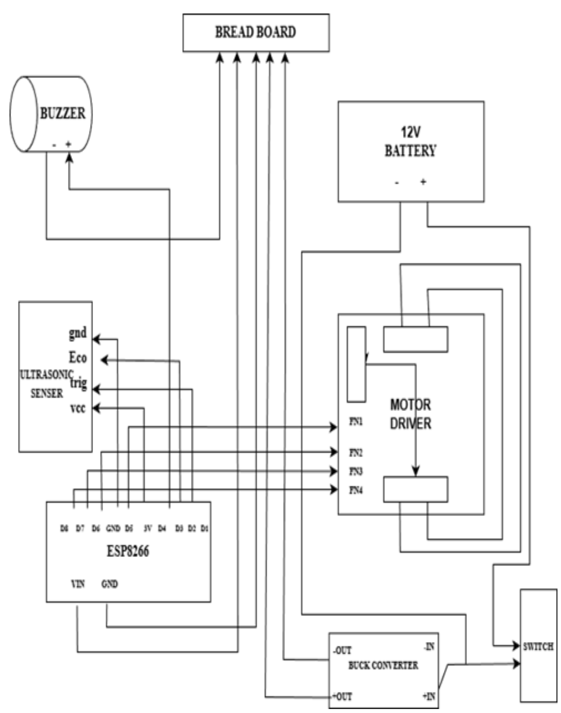

Figure7: Circuit Diagram

The ESP8266 firmware is developed using the Arduino IDE with appropriate libraries for sensor interfacing and network communication. The main program loop implements a state machine that coordinates sensor reading, object detection, decision-making, and actuation.
Sensor reading routines execute at defined intervals, with ultrasonic measurements occurring at approximately 100 ms intervals to provide timely obstacle detection. Distance values are filtered using moving average or Kalman filtering tech- niques to reduce measurement noise.

Object detection integration involves establishing commu- nication between the ESP8266 and a processing unit running the YOLO algorithm. This can be implemented through se- rial communication, Wi-Fi data transfer, or shared memory depending on the specific hardware configuration.
The collision avoidance logic evaluates multiple factors including detected object positions, ultrasonic distance measurement, current vehicle speed, and braking distances. When threat conditions are met, the system generates appropriate warnings through the buzzer and, if configured for autonomous operation, initiates evasive maneuvers.

IoT functionality leverages ESP8266's native Wi-Fi capabilities to connect to access points or create a standalone access point. Data transmission uses lightweight protocols suitable for embedded systems, with error handling and reconnection logic ensuring robust communication.
B. Testing Methodology
Object detection testing evaluates YOLO algorithm perfor- mance under different lighting conditions, object types, and distances. Metrics including precision, recall, and inference time quantify detection capability. Test cases encompass var- ious obstacle types relevant to driving scenarios.
Integration testing assesses end-to-end system behavior when all components operate simultaneously. These tests identify potential timing issues, resource conflicts, or com- munication bottlenecks that may not appear during isolated component testing.
Real-world testing involves operating the robotic vehicle in controlled environments that simulate driving conditions. Obstacles are positioned at various distances and angles to evaluate collision detection and avoidance capabilities. Sys- tem response times from detection to warning activation are measured and analyzed.

V. RESULTS

The YOLO-based ADAS system demonstrated reliable real- time object detection with low latency and strong performance in well-lit conditions, though accuracy decreased in poor weather or low visibility. Ultrasonic sensors provided accurate short-range measurements with errors under 5%, complementing the camera's long-range detection. System response times averaged 150–300 ms, suitable for low-speed warning applications. IoT features enabled effective remote monitoring, though dependent on network stability. Power consumption remained within limits for battery-based operation, and overall system cost was significantly lower than commercial ADAS solutions, making it ideal for education and prototyping.

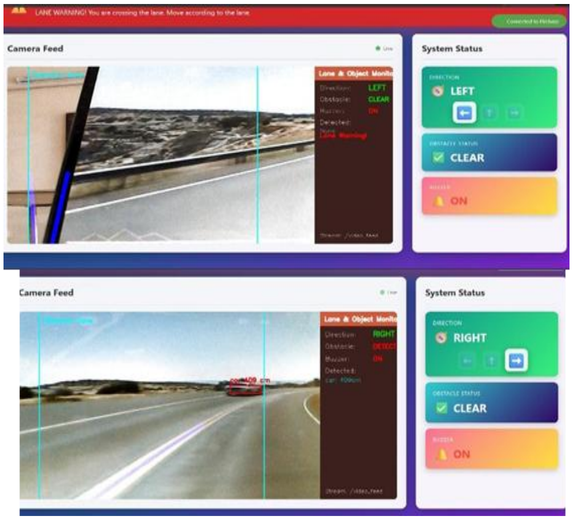

Figure8: Snapshots of Object Detected

VI. FUTURE ENHANCEMENTS

Several directions for future work have been identified. Upgrading to more powerful microcontrollers such as ESP32 would enable on-device execution of more complex YOLO models while adding capabilities like dual- core processing and Bluetooth connectivity.

Incorporating additional sensor types including LiDAR, radar, or stereo cameras would improve detection robustness through sensor fusion. Machine learning algorithms could integrate multi-modal sensor data to enhance accuracy and reliability under diverse conditions.

Implementing predictive algorithms that consider object trajectories and vehicle dynamics would enable anticipatory collision avoidance. Such capabilities require modeling object motion, predicting future positions, and calculating optimal evasive actions.

Expanding IoT functionality to support vehicle-to-vehicle and vehicle-to-infrastructure communication would integrate the system into broader intelligent transportation networks. Cooperative awareness among nearby vehicles could significantly enhance safety through shared environmental information.

Advanced driver behaviour analysis using logged data could identify patterns, assess risk levels, and provide personalized safety recommendations. Machine learning models trained on operational data could continuously improve system performance and adapt to individual driving characteristics.

Integration with existing vehicle systems through CAN bus or OBD-II interfaces would enable more sophisticated interactions including automatic cruise control adjustment, pre-emptive brake priming, and coordination with electronic stability systems.

VII. CONCLUSION

This research successfully demonstrated the feasibility of implementing an effective ADAS prototype using YOLO algorithm integration with affordable embedded hardware. The system achieved reliable object detection, obstacle avoidance, and remote monitoring capabilities while maintaining low cost and moderate power consumption.

The combination of computer vision through YOLO, prox- imity sensing via ultrasonic sensors, and IoT connectivity through ESP8266 provided a comprehensive solution addressing multiple aspects of driver assistance. Testing validated the system's ability to detect obstacles, measure distances accurately, and provide timely warnings under various conditions. While challenges related to processing constraints, environmental factors, and connectivity were identified, the overall system architecture proved viable for educational and research applications. The modular design facilitates future enhancements and adaptation to different use cases.

This work contributes to the growing body of research on accessible ADAS implementations, demonstrating that so- phisticated safety features need not require prohibitive costs. As embedded hardware continues to advance and AI algo- rithms become more efficient, systems like this will play an increasingly important role in democratizing automotive safety technology.

The lessons learned from this implementation provide valu- able insights for future embedded ADAS projects. Careful component selection, efficient algorithm implementation, and robust integration practices are essential for creating effective real-time safety systems with resource-constrained hardware.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779- 788.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition, 2017, pp. 7263-7271.

[3] J.Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[4] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[5] G. Jocher, "YOLOv5," GitHub repository, 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[6] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022.

[7] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint arXiv:2207.02696, 2022.

[8] M. Bijelic et al., "Seeing through fog without seeing fog: Deep multi- modal sensor fusion in unseen adverse weather," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11682-11692.

[9] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," Journal of Field Robotics, vol. 37, no. 3, pp. 362-386, 2020.

[10] K. Muhammad et al., "Deep learning for safe autonomous driving: Current challenges and future directions," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4316-4336,2020.

[11] B. Karthika, M. Dharssinee, V. Reshma, R. Venkatesh, and R. Sujarani, "Real-time object detection using deep learning," in 15th International Conference on Computing Communication and Networking Technology, India, 2024.

[12] A. S. Selvi, P. S. Aadesh, B. Manoharan, and S. Harinarayanan, "Real time multiple objects tracking and object detection using YOLOv7 and FairMOT algorithm," Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, 2023.

[13] G. Lavanya and S. D. Pande, "Enhancing real-time object detection with YOLO algorithm," School of Computer Science and Engineering, Andhra Pradesh, India, 2023.

[14] D. Nister, "An efficient solution to the five-point relative pose problem," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pp. 756-770, 2004.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354-3361.

[16] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213-3223.

[17] C. Chen et al., "Progressive feature alignment for unsupervised domain adaptation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 627-636.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[20] T. Y. Lin et al., "Microsoft COCO: Common objects in context," in European Conference on Computer Vision, Springer, 2014, pp. 740-755.