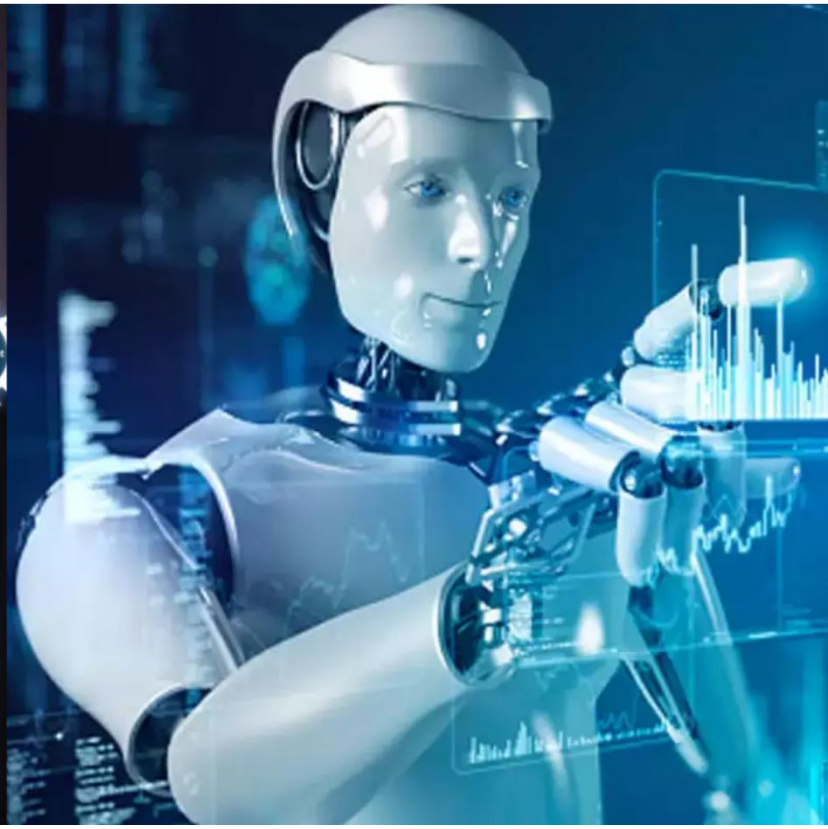
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 3, March 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503175|

Real-Time PPE Detection on Edge Devices using YOLO-Based API for Industrial Safety

Mrs. Janani G[1], Ashwin Sathappan V[2], Bala M[3], Bhuvan Kalyan G V[4], Giridharan P[5]

Associate Professor, Department of Information Technology, R.M.D Engineering College, Chennai, Tamil Nadu, India[1]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Chennai, Tamil Nadu, India[2,3,4,5]

ABSTRACT: Ensuring worker safety in industrial environments is a critical concern, particularly in enforcing the use of Personal Protective Equipment (PPE). Traditional cloud-based monitoring systems introduce latency and raise privacy concerns. This paper presents a real-time PPE detection system deployed on edge devices using a lightweight YOLO-based model integrated with a RESTful API. The system leverages a Raspberry Pi-based architecture to perform on-device inference, reducing latency and bandwidth usage. Experimental results demonstrate improved response time and efficient resource utilization, making the proposed system suitable for real-time industrial applications.

KEYWORDS: PPE Detection, Edge Computing, YOLO, Raspberry Pi, API Integration, Industrial Safety.

I. INTRODUCTION

Industrial safety compliance is essential to prevent workplace accidents. Monitoring the use of PPE such as helmets, gloves, and safety vests is often performed manually, which is inefficient and prone to human error. Recent advancements in computer vision and deep learning have enabled automated PPE detection systems.

However, most existing solutions rely on cloud-based inference, which introduces latency and potential privacy risks. To address these challenges, this paper proposes an edge-based PPE detection system that performs real-time inference using a lightweight YOLO model deployed on a Raspberry Pi device. Additionally, a RESTful API is developed to allow seamless integration with existing surveillance systems.

The remainder of this paper is organized as follows. Section II presents the related work on PPE detection and edge computing. Section III describes the proposed edge-based PPE detection system, including system architecture, model deployment, and API integration. The overall workflow of the system is illustrated using an architectural diagram. Section IV presents the experimental results and performance analysis of the proposed system. Finally, Section V concludes the paper and discusses future work.

II. LITERATURE SURVEY

Object detection models such as YOLO have been widely used for real-time applications due to their speed and accuracy. Redmon et al. [1] introduced the YOLO framework for unified object detection. Later improvements such as YOLOv4 and YOLOv8 enhanced detection performance and made them more suitable for real-time and edge deployment scenarios [2], [3].

Edge computing has gained popularity in industrial IoT applications due to reduced latency and improved data privacy [4]. Several studies have explored deploying lightweight deep learning models on embedded systems such as Raspberry Pi for real-time applications [5], [6]. Additionally, hardware platforms like Raspberry Pi have been widely used for prototyping intelligent edge solutions [7]. However, integrating such systems with scalable APIs for industrial deployment remains an area of active research.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503175|

III. PROPOSED SYSTEM

A. SYSTEM OVERVIEW

The proposed system is designed to perform real-time PPE detection using an edge computing approach. It consists of four primary components: a video input source, an edge device based on Raspberry Pi, a YOLO-based object detection model, and an API layer for communication. The system captures live video frames from a camera and processes them locally on the edge device to detect PPE compliance. By performing inference directly on the device, the system reduces dependency on cloud infrastructure, minimizes latency, and ensures faster decision-making in industrial environments.

B. SYSTEM ARCHITECTURE

The architecture of the proposed system is shown in Figure 1.

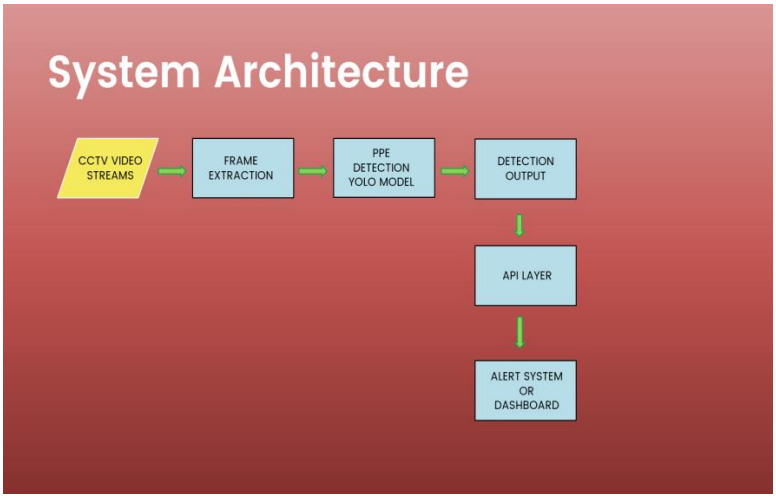


Figure 1. System Architecture of Edge-Based PPE Detection

As illustrated in Figure 1., the system follows a pipeline where the input video stream is first converted into frames and then passed to the edge device for processing. The YOLO model deployed on the edge device analyzes each frame and detects the presence or absence of PPE by generating bounding boxes and class labels. The detection output is then transmitted to the API layer, which facilitates communication with external systems such as dashboards or alert mechanisms. This architecture ensures efficient real-time processing while maintaining scalability and modularity.

C. DATA PROCESSING AND PREPROCESSING

The input video stream is processed frame-by-frame to ensure real-time performance. Each frame undergoes preprocessing steps such as resizing to a fixed resolution, typically 640×640 pixels, to match the input requirements of the YOLO model. The pixel values are normalized to improve model performance, and the frames are converted into tensor format for efficient computation. These preprocessing steps standardize the input data and enable consistent and optimized inference on the edge device.

D. DETECTION PIPELINE

The detection pipeline operates continuously in real time:

1. Frame capture from video stream
2. Preprocessing of input frame
3. Inference using YOLO model
4. Generation of bounding boxes and class labels
5. Filtering based on confidence threshold

The detection pipeline operates continuously to process incoming video frames in real time. Initially, frames are captured from the video stream and passed through the preprocessing stage. The processed frames are then fed into the

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503175|

YOLO model, which performs inference to detect objects related to PPE such as helmets and safety vests. The model generates bounding boxes, confidence scores, and class labels for each detected object. These detections are filtered using a predefined confidence threshold to eliminate low-confidence predictions. The final results are then forwarded to the API layer for further processing and response generation.

E. MODEL DEPLOYMENT

To ensure efficient execution on resource-constrained hardware, a lightweight version of the YOLO model is deployed on the Raspberry Pi. The model is optimized by reducing input resolution, using a compact architecture such as YOLOv8n, and streamlining the inference pipeline [3]. These optimizations help achieve a balance between detection accuracy and computational efficiency. As a result, the system is capable of performing real-time detection while maintaining low resource consumption on the edge device.

F. API INTEGRATION

A RESTful API is implemented to enable seamless communication between the edge device and external applications. The API is developed using a lightweight framework such as Flask and provides endpoints for accessing detection results and triggering alerts. For instance, the /detect endpoint returns the detected objects along with their confidence scores, while the /alert endpoint can be used to generate notifications in case of safety violations. This API-based design allows easy integration with existing surveillance systems and industrial monitoring platforms.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

The proposed system was implemented on a Raspberry Pi device and evaluated using real-time video input captured from a camera. The experiments were conducted under various environmental conditions to assess the system's robustness and performance. The evaluation focused on analyzing the system's ability to process frames efficiently and detect PPE accurately in real-world scenarios.

B. PERFORMANCE METRICS

The performance of the system was evaluated using key metrics such as frames per second (FPS), latency, CPU utilization, and memory usage. FPS measures the number of frames processed per second, indicating the system's real-time capability. Latency represents the time taken to process each frame, while CPU utilization and memory usage reflect the computational efficiency of the system. These metrics provide a comprehensive understanding of the system's performance on edge devices.

V. RESULTS

The experimental results demonstrate that the system achieves an average processing speed of 8–12 FPS with a latency of approximately 120 milliseconds. The CPU utilization remains around 65%, while memory usage is approximately 450 MB. These results indicate that the system is capable of performing real-time PPE detection with acceptable performance on resource-constrained hardware.

Parameters	Values
FPS	8-12
Latency	~120 ms
CPU Usage	65%
Memory Usage	450 MB

D. COMPARATIVE ANALYSIS

A comparison between cloud-based and edge-based approaches highlights the advantages of the proposed system. Cloud-based systems typically exhibit higher latency due to network communication and require continuous internet connectivity. In contrast, the edge-based system processes data locally, resulting in significantly lower latency and improved response time. Additionally, the edge approach enhances data privacy and reduces bandwidth usage, making it more suitable for industrial applications [4].

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503175|

Parameters	Cloud-Based System	Edge-Based System
Latency	High (~400 ms)	Low (~120 ms)
Dependency	Internet Required	No Internet Required
Privacy	Lower	Higher
Scalability	Moderate	High

E. ACCURACY EVALUATION

The accuracy of the detection model was evaluated using standard metrics such as precision, recall, and F1-score. The model achieved a precision of 0.89, recall of 0.85, and an F1-score of 0.87, indicating reliable detection performance. These results demonstrate that the system maintains a good balance between accuracy and efficiency, making it suitable for real-time industrial safety monitoring.

VI. CONCLUSION

This paper presented a real-time PPE detection system using a YOLO-based model deployed on an edge device. The integration of a RESTful API enables efficient communication with monitoring systems. Experimental results demonstrate that the proposed system achieves a balance between performance and efficiency.

The study highlights the potential of edge computing in enhancing workplace safety and provides a foundation for future research in intelligent industrial monitoring systems.

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 (CVPR), 2016, pp. 779–788.
[2] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv preprint arXiv:2004.10934, 2020.
[3] Ultralytics, “YOLOv8 Documentation,” 2023. [Online]. Available: https://docs.ultralytics.com
[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[5] S. Mittal, “A Survey of Techniques for Optimizing Deep Learning on Embedded Systems,” Journal of Systems Architecture, vol. 89, pp. 28–40, 2018.
[6] A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv preprint arXiv:1704.04861, 2017.
[7] Raspberry Pi Foundation, “Raspberry Pi Documentation,” 2023. [Online]. Available: https://www.raspberrypi.org/documentation/

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details