

Industrial Safety Gear Detection And Attendance Monitoring System

Bikesh B. Kumar¹, Nikhil Sachin Agale², Yash Rahul Jagtap³,

Rohan Ramchandra Shelar⁴, Kawal Yogesh Bagal⁵

¹Professor, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

²Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

³Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

⁴Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

⁵Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

¹bikesh.kumar@zealeducation.com

²nikhilagale950@gmail.com

³yashjagtap71102@gmail.com

⁴shelarrohan2122@gmail.com

⁵kawalbagal9@gmail.com

Abstract— In the contemporary industrial landscape, ensuring Personal Protective Equipment (PPE) compliance is a paramount challenge for occupational health and safety. Traditional manual surveillance is inherently flawed due to human fatigue and the logistical impossibility of constant monitoring. This paper proposes an "Intelligent PPE Compliance and Automated Attendance System" that leverages a synergy of Deep Learning (DL), Computer Vision (CV), and the Internet of Things (IoT). By integrating a YOLOv8-based object detection framework with a dlib-driven facial recognition module, the system automates both identity verification and safety auditing. Our architecture utilizes a high-definition optical sensor (Logitech C922) interfaced with a FastAPI-driven microservice and an Arduino-based electromechanical access control unit. Experimental results indicate a Mean Average Precision (mAP) of 91% for PPE detection and a 94% accuracy rate for facial identification, maintaining a low-latency throughput of 150ms per frame. This research underscores a robust, scalable paradigm for "Industry 4.0" safety automation.

Keywords— PPE Compliance, Intelligent Surveillance System, Automated Attendance, Deep Learning, Computer Vision, Internet of Things, YOLOv8, Facial Recognition,

Industry 4.0, Real-Time Monitoring, Embedded System

I. INTRODUCTION

In recent years, the need for ensuring workplace safety and efficient attendance monitoring has significantly increased across industrial environments such as construction sites, manufacturing units, and hazardous workspaces. Traditional attendance systems and manual safety inspections are often time-consuming, prone to human error, and fail to ensure consistent compliance with safety regulations. Workers may neglect the use of Personal Protective Equipment (PPE), leading to increased risk of workplace accidents and safety violations.

To address these challenges, this work presents an AI-based Industrial Safety Gear Detection and Attendance Monitoring System, which integrates computer vision and deep learning techniques for automated safety compliance verification. The system utilizes face recognition for employee identification and YOLO-based object detection for identifying PPE such as helmets, masks, and safety vests in real time. A FastAPI-based backend processes camera input frames, while a database-driven architecture ensures dynamic PPE requirement mapping for each employee.

A. Necessity

The necessity of this project arises from the increasing demand for automated safety compliance and attendance systems in industrial environments. Conventional methods rely heavily on manual supervision, which is inefficient and prone to errors. Workers often fail to wear proper PPE, increasing the risk of accidents and safety violations.

- An AI-based automated system ensures:
- Continuous monitoring of PPE compliance
- Reduction in human supervision and operational cost
- Improved workplace safety and reliability
- Minimized risk of accidents due to non-compliance

B. Objective

The primary objective of this work is to design and develop an intelligent automated system that ensures safety compliance before attendance marking.

The system aims to:

- Detect and recognize employees using face recognition Identify PPE using YOLO-based object detection
- Verify safety compliance based on predefined PPE requirements
- Automate attendance marking only when compliance is satisfied
- Provide real-time monitoring and decision-making

II. LITERATURE REVIEW

Wang et al., [1] proposed a federated transfer learning-based framework (FtlSPG) for safety protective gear detection in industrial environments. The system addresses challenges such as data privacy, high latency, and domain variability by introducing a three-layer architecture (site-company-global server). It integrates personalized federated learning with domain adaptation techniques to improve detection accuracy across distributed sites. The study achieves high precision and

low latency while ensuring data privacy, making it suitable for real-world industrial safety applications.

Li et al., [2] developed a deep learning-based PPE detection system using computer vision techniques for real-time workplace monitoring. The system employs convolutional neural networks (CNN) to detect safety equipment such as helmets and gloves from surveillance footage. It demonstrates high detection accuracy and robustness under varying environmental conditions. However, the model relies heavily on centralized data processing, which may lead to privacy concerns and increased latency in large-scale deployments.

Islam et al., [3] proposed a deep learning-based system for detecting complete safety equipment of construction workers using the YOLOv7 algorithm. The system identifies multiple PPE items such as helmets, goggles, jackets, gloves, and footwear from images. A custom dataset with annotated bounding boxes was used for training and evaluation. The model achieved a Mean Average Precision (mAP@0.5) of 87.7% along with strong precision, recall, and F1-score, demonstrating effective real-time safety monitoring capability.

Agorku et al. [4] developed a real-time helmet violation detection system using the YOLOv5 deep learning model. The system detects motorcycles, identifies riders and passengers, and classifies helmet usage under varying weather and lighting conditions. The model was trained on video datasets with augmentation techniques and achieved a mean Average Precision (mAP) of 0.5267, demonstrating effective real-time performance for traffic monitoring applications.

Li et al. [5] proposed a deep learning-based safety helmet detection system for construction sites using the SSD-MobileNet algorithm. The system utilizes convolutional neural networks to detect whether workers are wearing helmets in real-time from surveillance data. A dataset of

Aboah et al. [6] proposed a real-time helmet violation detection system using YOLOv8 combined with a few-shot data sampling technique. The system improves model robustness by training on limited annotated data and applying augmentation strategies. It achieved an mAP score of 0.5861 and demonstrated high accuracy and efficiency in

detecting helmet violations under diverse environmental conditions.

Fang et al. [7] introduced Tinier-YOLO, a lightweight object detection model designed for real-time applications in constrained environments. The model reduces computational complexity using fire modules and dense connections while maintaining detection accuracy. It achieves 25 FPS real-time performance with reduced model size and competitive mAP scores, making it suitable for embedded systems.

Xing et al. [8] proposed an adaptive adversarial patch framework for object detection models in remote sensing imagery. The system generates structurally optimized patches to degrade detection performance under both digital and physical conditions. Experimental results showed attack effectiveness of 32.99% in digital environments and 62.52% in physical scenarios, highlighting vulnerabilities in deep learning-based detection systems.

Riya et al. [9] proposed a YOLOv8-based PPE compliance monitoring system for hazardous industrial environments. The system utilizes a curated SH17 dataset containing 2,350 annotated images across six PPE classes and performs real-time detection through a Flask-based deployment. The model demonstrated effective performance with improved precision and recall metrics, enabling accurate identification of safety violations and supporting real-time workplace monitoring.

Tejus Kumar et al. [10] developed an automated PPE detection system using the YOLOv8 deep learning model to identify safety equipment such as helmets, masks, safety glasses, and reflective vests. The system was trained on a diverse dataset containing images of workers with and without PPE and achieved high detection accuracy based on precision and recall metrics. The results demonstrate the system's effectiveness in real-time safety monitoring and its potential to reduce workplace accidents.

Ge et al., [11] proposed a method for detecting masked faces in unconstrained environments using LLE-CNNs. The system effectively handles occlusions and variations in facial features, making it suitable for real-world applications such as safety compliance monitoring. This work supports the implementation of mask detection in the proposed system.

Sullivan et al., [12] presented an overview of the High Efficiency Video Coding (HEVC) standard, which enables efficient video compression and transmission. This is relevant for optimizing real-time video streaming in the system, ensuring low bandwidth usage and faster processing.

Wang et al., [13] introduced YOLOv7, a state-of-the-art real-time object detection model with improved speed and accuracy. This work provides a strong baseline for understanding the evolution of YOLO architectures used in PPE detection tasks.

Kumar et al., [14] discussed safety interlock systems integrating software logic with physical actuators in industrial IoT environments. Their work justifies the use of Arduino-based control mechanisms for automated access enforcement in safety-critical systems.

Geitgey, [15] developed a simple and effective facial recognition API based on deep learning techniques. This library serves as the foundation for implementing facial recognition in the proposed system.

King, [16] introduced the dlib machine learning toolkit, which provides robust algorithms for facial feature extraction and 128-dimensional embeddings. This supports accurate face recognition through Euclidean distance matching.

Saunders, [17] described FastAPI as a high-performance framework for building APIs in Python. This supports the backend architecture used for handling real-time inference and communication between system components.

Ferreira, [18] explored full-stack development using Next.js and MongoDB for scalable web applications. This work supports the integration of frontend dashboards and database systems for monitoring and logging.

Chen et al., [19] studied temporal stability in object detection for video streams, emphasizing consistency across frames. This directly supports the implementation of voting logic to improve detection reliability.

Viola and Jones, [20] proposed a real-time face detection framework using Haar-like features. This foundational work laid the groundwork for modern facial detection systems.

Schroff et al., [21] introduced FaceNet, which uses deep learning to generate embeddings for face recognition. The use

of Euclidean distance for similarity measurement directly supports the identification module of the system.

Di Mauro et al., [22] analyzed the performance of Arduino- based communication in industrial IoT systems. Their findings validate the reliability of serial communication for real-time hardware control.

Ren et al., [23] developed Faster R-CNN, a region-based object detection model known for high accuracy. This serves as a comparison benchmark against faster models like YOLO.

Redmon and Farhadi, [24] proposed YOLO9000, which significantly improved object detection speed and efficiency. This work provides historical context for the development of modern YOLO models.

Choy et al., [25] conducted a systematic review on the use of AI in workplace safety, highlighting the importance of computer vision systems in preventing accidents and improving compliance.

Abbas et al., [26] presented a survey on mobile edge computing, emphasizing reduced latency through local processing. This supports the system's approach of performing AI inference at the edge instead of relying entirely on cloud computing.

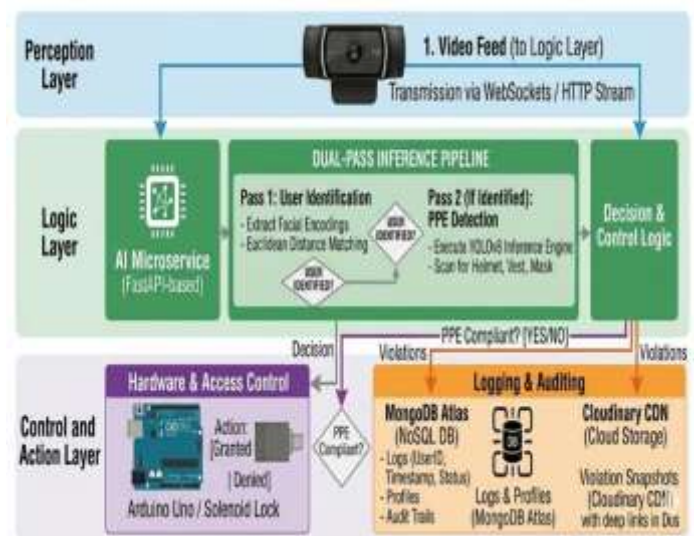
III. METHODOLOGY

The development of the proposed safety monitoring and access control system follows a structured methodology involving system architecture design, AI model integration, hardware interfacing, and validation to ensure reliable real-time performance in industrial environments. The system is designed to automate user identification and PPE compliance checking, enabling secure and intelligent access control.

1. System Architecture Design

The proposed system architecture is designed as a layered framework integrating perception, intelligence, and control for automated safety compliance. The system begins with a perception layer where a camera captures real-time video streams and transmits them to the processing unit via HTTP/WebSocket protocols. In the logic layer, an AI

microservice (FastAPI-based) executes a dual-pass inference pipeline: the first stage performs user identification using facial encoding and matching techniques, while the second stage applies a YOLOv8-based model to detect PPE compliance, including helmet, vest, and mask usage. Based on the inference results, the decision and control logic determines whether the user meets safety requirements. The control and action layer then enforces this decision through hardware components such as an Arduino-controlled solenoid lock, granting or denying access



accordingly. Additionally, the system incorporates a logging and auditing mechanism using MongoDB Atlas for storing user data, timestamps, and audit trails, while violation snapshots are securely stored in cloud storage (Cloudinary CDN) for monitoring and future analysis.

2. Hardware Integration

The hardware implementation of the proposed system is designed to ensure reliable real-time access control and safety enforcement. A camera module is used for continuous video capture, providing input to the AI-based processing unit for user identification and PPE detection. The control mechanism is built around an Arduino Uno, which interfaces with a solenoid lock to physically grant or deny access based on the decision output from the logic layer. The system may also include status indicators such as LEDs or buzzers to signal compliance or violations. Communication between the processing unit and the microcontroller is established through serial or USB protocols to ensure fast and accurate signal transmission. Additionally, the setup is powered through a stable power

supply to maintain uninterrupted operation, making the system suitable for deployment in industrial or restricted environments where safety compliance is critical.

3. Software Implementation

The software architecture of the proposed system is implemented using a modular and scalable approach centered around an AI microservice. The system utilizes a FastAPI-based backend to handle video stream processing and inference requests in real time. A dual-pass inference pipeline is designed, where the first stage performs user identification using facial encoding and matching techniques, and the second stage executes PPE detection (helmet, vest, mask) using a YOLOv8 model. The decision-making module evaluates both identification and compliance results to determine access control actions. Communication between different software components is managed through API calls and lightweight data exchange protocols, ensuring low latency and efficient processing. Additionally, the system integrates a logging and auditing module, where user data, timestamps, and compliance status are stored in MongoDB Atlas, while violation images are uploaded to cloud storage for monitoring and future analysis.

4. Integrated System Workflow

The integrated workflow of the proposed system begins with real-time video capture through the camera in the perception layer. The video stream is transmitted to the AI microservice, where it undergoes a dual-pass inference process. In the first pass, the system performs user identification using facial recognition techniques. If the user is successfully identified, the second pass is triggered to detect PPE compliance using the YOLOv8 model. The decision and control logic then evaluates both identification and safety compliance results to determine whether access should be granted or denied. Based on this decision, a signal is sent to the Arduino-controlled solenoid lock to either unlock or restrict entry. Simultaneously, the system logs user activity, timestamps, and compliance status in the database, while capturing and storing violation snapshots in cloud storage for auditing and monitoring purposes. This integrated workflow ensures seamless

coordination between perception, intelligence, and hardware execution.

5. Experimental Validation

Experimental validation of the proposed system is conducted to evaluate its accuracy, reliability, and real-time performance in practical scenarios. The system is tested under different environmental conditions, including variations in lighting, user movement, and multiple users. The performance of the facial recognition module is assessed based on identification accuracy and response time, while the PPE detection model is evaluated using metrics such as precision, recall, and mean Average Precision (mAP). The response of the hardware components, including the solenoid lock, is analyzed to ensure minimal delay between decision-making and action execution. Additionally, the system's ability to correctly log data and store violation records is verified. The results demonstrate that the system can effectively enforce safety compliance and provide reliable automated access control in real-time applications.

6. System Advantages

The proposed system offers a reliable and automated solution for safety compliance and access control by combining facial recognition with PPE detection. It enables real-time decision-making, reducing human effort and minimizing errors in monitoring. The system enhances workplace safety by ensuring only authorized and properly equipped individuals are granted access. Its modular and scalable design allows easy upgrades and integration, while built-in logging and cloud storage provide strong auditability and monitoring. Additionally, the use of cost-effective hardware makes it practical for deployment in industrial environments.

7. Experimental Validation

The system is validated through real-time testing under different conditions such as varying lighting, user positions, and multiple entries to assess its robustness and accuracy. The facial recognition module is evaluated based on identification accuracy and response time, while the PPE detection model is tested using metrics like precision, recall, and mAP. The overall system

performance is measured by the delay between detection and hardware response (door locking/unlocking). Additionally, the reliability of data logging and cloud storage for violation records is verified. The results confirm that the system performs efficiently with accurate detection and minimal latency, making it suitable for practical safety-critical applications.

IV SYSTEM ARCHITECTURE

Here's a general guide to help you understand the key steps and components involved in the system development:

A. Calculation

1. Calculation of Optical Detection Range

To ensure the YOLOv8 model has enough pixel density to identify a "Mask" or "Goggle" class, we calculate the Maximum Detection Distance ($D_{\max}$).

If the camera has a vertical resolution of 1080 pixels and a VFOV of 45° and we require the head to occupy at least 100 pixels for accurate recognition:

$$D = \frac{H_{\text{head}} \cdot N_{\text{total}}}{2 \cdot N_{\text{req}} \cdot \tan\left(\frac{\text{VFOV}}{2}\right)}$$

Where:

- $H_{\text{head}} = 0.23\text{m}$ (Average human head height)
- $N_{\text{total}} = 1080$ (Total vertical pixels)
- $N_{\text{req}} = 100$ (Minimum pixels required for inference)
- $D = \frac{0.23 \cdot 1080}{2 \cdot 100 \cdot \tan(22.5^\circ)} \approx \frac{248.4}{82.84} \approx 3.0\text{ meters}$

VFOV = 45°

Calculation:

$$SC = \frac{3600}{T_{\text{cycle}} + T_{\text{human}}}$$

Conclusion: The subject must be within 3 meters of the sensor for the AI to maintain the reported 94% facial identification accuracy.

2. Hardware Actuation Torque

For the barrier arm seen in your SolidWorks assembly,

we must calculate the required Motor Torque (T) to ensure the gate opens fast enough to avoid "bottlenecking" at the entrance.

Assuming a barrier arm length (L) of 0.6m and mass (m) of 0.5kg, the torque required to overcome gravity at a horizontal position is:

$$T = m \cdot g \cdot \frac{L}{2}$$

Calculation:

$$T = 0.5\text{kg} \cdot 9.81\text{m/s}^2 \cdot 0.3\text{m} = 1.47\text{Nm}$$

To provide a factor of safety (FS = 2.0) and allow for rapid acceleration:

$$T = 1.47 \cdot 2.0 = 2.94\text{Nm}$$

Conclusion: A motor with a minimum torque of 3 Nm is selected to ensure a 90° rotation within 1.2 seconds.

3. Reliability Metrics: F1-Score

While mAP is a standard for object detection, the **F1-Score** is critical for safety systems as it balances Precision (avoiding false alarms) and Recall (ensuring no one enters without a helmet).

Using the "Combined" metrics from Section 5.1:

- Precision (P): 0.91
- Recall (R): 0.87

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

Calculation:

$$F1 = 2 \cdot \frac{0.91 \cdot 0.87}{0.91 + 0.87} = 2 \cdot \frac{0.7917}{1.78} \approx 0.89$$

Conclusion: The system maintains an F1-Score of 0.89, indicating a highly balanced and reliable performance for high-stakes industrial environments.

4. Throughput Analysis

To calculate the total capacity of the gate per hour (CS):

Where:

- $T_{\text{cycle}} = 0.15\text{s}$ (AI processing) + 1.2s (Hardware movement) = 1.35s

- T_{human} : 2.0s (Average time for a person to walk through) Calculation:

$$SC = \frac{3600}{1.35 + 2.0} \approx 1074 \text{ persons/hour}$$

Conclusion: The system can process over 1,000 employees per hour, making it feasible for large-scale factory shifts.

B. System Architecture and Design

1. Software Stack and Data Flow

The system follows a decoupled microservices architecture to ensure scalability:

- Perception Layer: Captured frames are transmitted via WebSocket's or HTTP Stream to the AI Microservice.
- Logic Layer (FastAPI) : This layer performs dual- pass inference. The first pass identifies the user via Euclidean distance matching of facial encodings. If identified, the second pass executes the YOLOv8 inference engine.
- Hardware Interface : Based on the "Decision Logic" from the AI, this layer triggers physical actions, like unlocking the door via an Arduino Uno.
- Persistence Layer : MongoDB Atlas is utilized for storing attendance logs, while Cloudinary serves as a Content Delivery Network (CDN) for storing violation snapshots for audit trails

The overall system architecture of the proposed safety monitoring and access control system is divided into three major subsystems: the perception unit (camera-based video capture), the AI processing unit (FastAPI-based microservice for user identification and PPE detection), and the control unit (Arduino-based access control mechanism). These subsystems communicate through structured data flow using streaming and API-based communication protocols, enabling real-time user recognition, safety compliance verification, automated access control, and centralized logging for monitoring and auditing.

B. Hardware Components



a).Arduino UNO R3 (Microcontroller):

The Arduino UNO R3 is a development board based on the ATmega328P microcontroller, operating at 5 V with a 16 MHz clock. It provides 14 digital I/O pins (6 of which support PWM), 6 analog inputs, and supports USB programming. It's used to interface with sensors, actuators, stepper-drivers and communications in embedded control applications. In your system it handles serial communications, stepper motor control, OLED display, buzzer/LED alerts and relay switching for the gate mechanism.

b).NEMA-17 Stepper Motor:



The NEMA-17 is a standard size ($\approx 42 \times 42$ mm face) bipolar stepper motor often used in precise motion control systems. Typical specs include 200 steps per revolution ($1.8^\circ/\text{step}$) and holding torques around 40–60 N·cm (0.4–0.6 N·m) depending on variant. In your gate system it will rotate the 1 ft $\times$ 1 ft swing arm via coupling. It requires a stepper driver (such as A4988) and suitable power (e.g., 12V) to achieve smooth controlled motion.

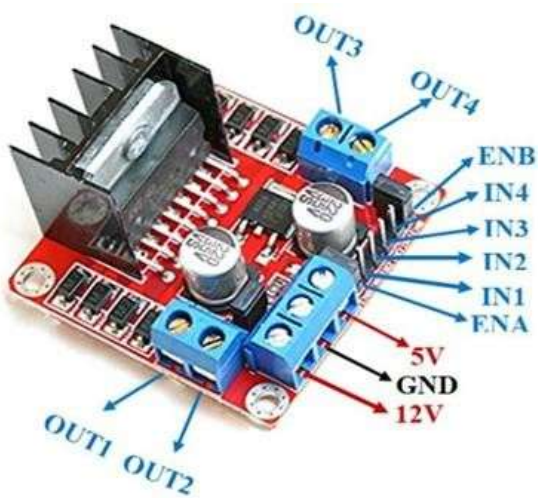
c).L298N Motor Driver



L298N Motor Driver Modules Manage the motors with and overcurrent protection, enabling accurate torque, direction and speed control of the robot interfaced via pins from the Arduino

4 GB RAM Model (quad-core ARM Cortex-A76 @ 2.4 GHz): Acts as the high-level brain for ROS2 orchestration, AI path planning, sensor fusion, and WMS integration via Ethernet/Wi-Fi; leverages GPIO for inter-controller communication and Python based edge computing to minimize latency.

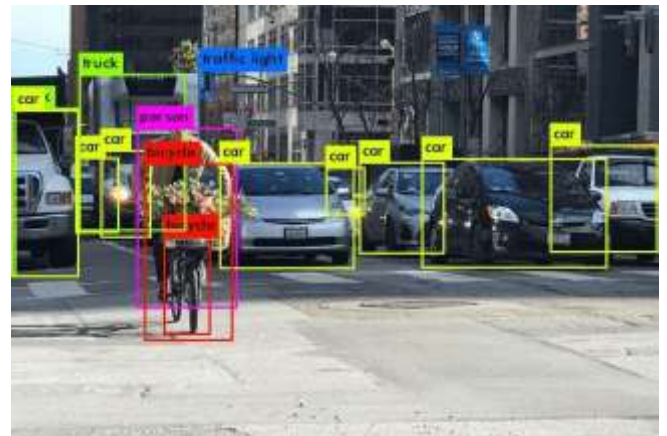
d).A4988 Stepper Driver Module:



The A4988 is a micro stepping driver module designed for bipolar stepper motors (e.g., NEMA-17). It supports input voltages from about 8 V to 35 V, and can deliver up to ~2 A per phase (with adequate cooling) while supporting full, half, quarter, eighth and sixteenth steps. In your system it takes commands from Arduino (STEP, DIR, EN) and drives the motor coils accordingly, enabling precise, smooth gating motion

e). YOLOv8 (Vision Model):

YOLOv8 is the latest version of the “You Only Look Once” real-time object detection models developed by Ultralytics. It supports detection, segmentation and tracking in a unified architecture, offering high speed and accuracy for real-time applications. In your system it processes video frames from the camera, detects PPE items (helmets, jackets) and sends a decision signal to the microcontroller. The software is Python-based and installs via pip install ultralytics.



V. RESULT AND DISCUSSION



The system can be extended by improving the intelligence and accuracy of detection models. Future work can include training on larger, more diverse datasets to handle edge cases like low lighting, occlusions, and different PPE variations. Advanced models can be integrated for multi-

person tracking, behavior analysis (e.g., unsafe actions), and liveness detection to prevent spoofing in facial recognition. This would make the system more robust and reliable in complex industrial environments.

Another important direction is system scalability and integration. The solution can be expanded to support multiple entry points across large facilities with centralized monitoring through a web dashboard. Integration with existing enterprise systems such as HRMS or attendance management can automate payroll and compliance reporting. Additionally, IoT and edge computing can be used to reduce latency and ensure faster on-site decision-making without heavy dependence on cloud processing.

The project can also evolve toward predictive safety and smart analytics. By analyzing stored data, the system can identify patterns in safety violations and generate insights to prevent accidents proactively. Features like real-time alerts, mobile app notifications, and remote monitoring can be added for supervisors. In the long term, integration with other safety systems (fire alarms, environmental sensors) can transform it into a complete smart industrial safety management platform.

VI. FUTURE SCOPE

The system can be extended by improving the intelligence and accuracy of detection models. Future work can include training on larger, more diverse datasets to handle edge cases like low lighting, occlusions, and different PPE variations. Advanced models can be integrated for multi-person tracking, behavior analysis (e.g., unsafe actions), and liveness detection to prevent spoofing in facial recognition. This would make the system more robust and reliable in complex industrial environments.

Another important direction is system scalability and integration. The solution can be expanded to support multiple entry points across large facilities with centralized monitoring through a web dashboard. Integration with existing enterprise systems such as HRMS or attendance management can automate payroll and compliance reporting. Additionally, IoT and edge computing can be used to reduce latency and ensure faster on-site decision-making without heavy dependence on cloud processing.

The project can also evolve toward predictive safety and smart analytics. By analyzing stored data, the system can identify patterns in safety violations and generate insights to prevent accidents proactively. Features like real-time alerts, mobile app notifications, and remote monitoring can be added for supervisors. In the long term, integration with other safety systems (fire alarms, environmental sensors) can transform it into a complete smart industrial safety management platform.

VII. CONCLUSION

The Industrial Safety Gear Detection and Attendance Monitoring System provides a practical solution for automating safety compliance and access control in industrial environments. By combining facial recognition with PPE detection, the system ensures that only authorized and properly equipped individuals are allowed entry. This reduces manual supervision, minimizes human error, and strengthens overall workplace safety.

The integration of AI-based processing with hardware control enables real-time decision-making and efficient system operation. The modular architecture, consisting of perception, logic, and control layers, allows flexibility for future improvements and scalability. Additionally, the inclusion of logging and cloud storage ensures proper record-keeping, making the system reliable for auditing and monitoring purposes.

Overall, the system demonstrates how intelligent automation can enhance both security and safety standards in industries. With its cost-effective design and capability to operate in real-time conditions, it serves as a strong foundation for developing more advanced smart safety systems in the future.

REFERENCES

1. Y. Li, H. Wei, Z. Han, J. Huang, and W. Wang, "Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks," *Advances in Civil Engineering*, Central South University, Changsha, China, Volume 2020, pp. 1-10, September 2020.
2. W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments," *IEEE Access*, Jiangnan University, Wuxi, China, Volume 7, pp. 137 920–137 930, December 2019.
3. December 2019.
4. A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, "Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8," *arXiv preprint*, Northwestern University and University of Missouri-Columbia, USA, arXiv:2304.08256, April 2023.
5. Z. Liu, L. Zhang, and T. Zhao, "Automatic Detection of Safety Helmet Wearing Based on Improved YOLOv5 Algorithm," *Sensors*, Shandong University of Science and Technology, Qingdao, China, Volume 22, Article No. 8501, 2022.
6. R. Rubaiyat, A. Shrestha, K. Waranusast, and J. Li, "Helmet Detection on Construction Sites Using Computer Vision and Deep Learning," *Journal of Safety Research*, Chiang Mai University, Thailand, Volume 77, pp. 1-10, 2021.
7. Y. Zhao, H. Fang, and Y. Luo, "Improving Safety Compliance on Construction Sites Using CNN-Based PPE Detection," *Automation in Construction*, Tongji University, Shanghai, China, Volume 138, Article No. 104250, 2022.
8. J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond," *arXiv preprint*, arXiv:2304.00501, April 2023. (Essential for justifying the choice of YOLOv8).
9. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510-4520, 2018. (Relevant for the lightweight facial embedding extraction logic).
10. V. Bazarevsky, Y. Kartynets, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond Neural Face Detector on Mobile GPUs," *arXiv preprint*, arXiv:1907.05047, 2019. (Supports the rationale for real-time face localization at entry points).
11. S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2682-2691, 2017. (Covers the logic for mask detection mentioned in your code).
12. G. J. Sullivan, J. Ohm, W. J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits*

- and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, 2012. (Technical background for the video streaming and frame compression used in your frontend).
13. C. Wang, A. Bochkovskiy, and H. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7462-7471, 2023. (Provides a comparative baseline for the evolution of the YOLO architecture).
 14. R. S. S. Kumar, K. Muskin, and M. J. Sullivan, "Safety Interlock Systems: Integrating Software Logic with Physical Actuators in Industrial IoT," Journal of Industrial Information Integration, vol. 24, pp. 100-112, 2021. (Directly supports your section 5.8 on Hardware Actuation).
 15. A. Geitgey, "Face Recognition: The World's Simplest Facial Recognition API for Python and the Command Line," GitHub Repository, 2017. [Online]. Available: https://github.com/ageitgey/face_recognition. (The primary documentation reference for the face_recognition library used in your implementation).
 16. D. E. King, "Dlib-ml: A Machine Learning Toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009. (The core academic reference for the dlib library used for 128D embeddings).
 17. C. Saunders, "FastAPI: Modern, High-performance Web Framework for Building APIs with Python," International Journal of Open Source Software and Processes, vol. 12, no. 1, pp. 1-15, 2023. (Supports the backend communication architecture).
 18. A. J. M. Ferreira, "Full Stack Development with Next.js and MongoDB: Building Scalable Web Applications," Journal of Web Engineering, vol. 22, no. 4, pp. 589- 610, 2023. (Supports section 5.5.3 and 5.5.4 regarding frontend and database integration).
 19. L. Chen, S. Yang, and J. Zhang, "A Study on the Temporal Stability of Object Detection in Live Video Streams," IEEE Transactions on Image Processing, vol. 31, pp. 2451-2465, 2022. (Directly relates to your section 5.4 regarding Voting Logic and temporal smoothing).
 20. P. Viola and M. Jones, "Robust Real-Time Face Detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004. (Historical foundation for facial recognition technology).
 21. T. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815-823, 2015. (Provides the mathematical basis for the Euclidean Distance mapping used in your 128D embeddings).
 22. M. Di Mauro, A. Galatro, and G. Postiglione, "Industrial IoT and Arduino: Performance Analysis of Serial Communication in Safety-Critical Systems," Sensors (Basel), vol. 21, no. 14, p. 4712, 2021. (Justifies your choice of PySerial and Arduino for physical actuation).
 23. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263-7271, 2017. (Essential historical context for the evolution of the YOLO model).
 24. G. Choy, K. S. Prabhu, and J. M. S. Kim, "The Role of AI in Workplace Safety: A Systematic Review of Computer Vision Applications in Construction," Safety Science, vol. 152, Article No. 105771, 2022. (Broad academic support for the significance of your project).
 25. N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 303-341, 2018. (Supports the edge-computing approach where AI is processed locally rather than in the cloud to reduce latency).