Industrial Workspace Safety Enforcer: AI-Powered Robotic Compliance System

Rahul Govadi
Dept of Computer Science and Engg
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
Email: govadirahul@gmail.com

Somalapalli Hansika
Dept of Computer Science and Engg
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
Email: somalapallihansika@gmail.com

Koushik Voota
Dept of Computer Science and Engg
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
Email: vkoushik321@gmail.com

Lakshmi Jayanth Reddy Nallamilli
Dept of Computer Science and Engg
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
Email: jay23238896@gmail.com

***Abstract*—Industrial worker's safety must be ensured in industry setups, yet old manual check for compliance to Personal Protective Equipment (PPE) is inefficient and liable to make human errors. The article offers an AI-based Industrial Workspace Safety Enforcer combining YOLOv8-based deep learning to implement PPE check real-time.The technology applies computer vision to verify whether the workers are donning the proper helmets, vests, gloves, masks, and boots before they enter the work environment. OpenCV, Ultralytics YOLO, and Python power the model, which is trained on a mixed dataset to provide high accuracy and low latency in PPE violation detection. Enforcement of compliance is automated by triggering alerts and restricting entry for non-compliant workers, significantly improving workplace safety. Experimental results show high mean Average Precision (mAP) and F1-score, validating the efficacy of the system. Future enhancements such as voice alert, IoT incorporation, and edge computing optimization are capable of bringing real-world implementation further.**

Keywords: Industrial Safety, Personal Protective Equipment (PPE), YOLOv8, Computer Vision, Deep Learning, PPE Compliance, Real-time Detection, Workplace Safety Automation, Occupational Hazard Prevention, Smart Entry Systems, AI-driven Surveillance, Edge Computing, PPE Monitoring, Access Control Systems.

I. INTRODUCTION

Industrial workplace safety is a problem, and employees are being subjected to hazardous conditions day by day. Adherence to personal protective equipment (PPE) needs to be strictly followed to prevent workplace accidents, injury, and death. Traditional safety monitoring practices, such as manual inspection, are prone to human frailty, variability, and inefficiency. The traditional methods are also incapable of detecting non-compliance in real time, thus greater risks and work issues.

With the advent of artificial intelligence (AI) and computer vision, there are newer paths to improve workplace safety with automation. The scope of this project is to develop an AI-powered robotic compliance system implementing strict real-time PPE monitoring at industrial gateways. With the application of advanced deep learning technology, namely the YOLOv8 model, the system is able to accurately detect whether workers are wearing the necessary PPE, including helmets, safety vests, gloves, masks, and boots. The system is intended to automate safety inspections, minimize human error, and create a more efficient and reliable compliance system.

The methodology relies on the gathering of a variety PPE dataset, deep-learning model training, and integration into a real-time robotic monitoring system. The datasets comprise public-domain PPE datasets (i.e., Roboflow, Kaggle) as well as manually collected images. The out- put dataset is tagged by computer vision software such as Roboflow and OpenCV and utilized for training a deep-learning model—i.e., YOLOv8 (You Only Look Once)—in trying to accurately detect PPE objects.

The trained YOLOv8 model is utilized to work in real time on camera streams to detect PPE compliance from the workers.The solution is made to automatically block the entry of individuals who lack adequate or improperly fitted protective equipment, avoiding human errors and improving site safety.Python, OpenCV, and Ultralytics YOLO are utilized in the solution to detect objects through deep learning, making the solution highly scalable and effective. Development will focus on voice notifications, IoT integration, edge optimization, and cloud analytics in the future to further enhance the efficiency and responsiveness of the system.

This research aims to bridge the gap between computer vision, robotics, and automation of industrial safety. Through the integration of AI-based PPE detection and real-time compliance monitoring, the system discussed herein aims to revolutionize workplace safety, reduce risk levels, and ensure strict regulatory compliance. The results of this research hold significant potential for significantly enhancing industrial safety standards and creating a new benchmark for automation-driven compliance solutions.

II. LITERATURE SURVEY

Enabling workplace safety by automated verification of compliance has been a research interest area with increasing growth. Manual checks based on traditional approaches using human resources are marred with shortcomings such as human fallibility, inefficiency, and variability. As computer vision and artificial intelligence continue to advance, researchers have developed automated monitoring of workplace safety and compliance with Personal Protective Equipment (PPE). One research proposed an AI-based monitoring system that used deep learning models for PPE detection to improve workplace safety by minimizing manual inspections [1]. A hybrid approach using image processing and IoT sensors to monitor compliance was proposed in another paper, with better accuracy in PPE validation [2]. Further, scientists also analyzed real-time monitoring methods utilizing CNNs for PPE violation detection and issuing alerts to improve safety mechanisms [3]. A research investigated reinforcement learning techniques to adaptively enhance detection models according to environmental conditions [4]. Another study emphasized adaptive thresholding methods in PPE detection to improve system responsiveness under varying light conditions [5].

An Automated Gate Pass System was used to track students, staff, and visitors in a university campus. The system offered security through electronic recording of the arrival and departure of staff, supplementing manual recording of logs [6]. While it effectively handled access control, it did not facilitate PPE authentication and real-time monitoring of safety compliance, both demanding close vigilance in industry environments where violations were highly dangerous. Another piece of research emphasized optimizing gate pass systems through coupling RFID with biometric authentication for heightening security, yet missing compliance verification on the safety side.[7]. Another study attempt merged facial recognition with AI-driven monitoring to prevent unauthorized personnel from gaining access but did not incorporate PPE detection as a basis for entry [8].

Another approach involved an Intelligent Automated Gate System (IAGS) based on QR code authentication. This system enabled legitimate personnel to gain access to sites using QR-based authentication while issuing real-time warnings against unauthorized access [9]. Yet, QR-based security solutions are still dependent on physical interaction, i.e., scanning an ID card, and therefore remain less useful for industrial safety compliance, where non-contact real-time is the preference. Researchers also suggested a different RFID-based authentication system that minimized physical interaction but did not use AI for PPE confirmation [10]. There was another study proposing a deep-learning-based vision system for security doors but specifically targeting identity authentication as opposed to PPE adherence [11].

Automated safety measures have also been investigated in the transportation sector. An Automated Railway Level Crossing Gate System using Programmable Logic Controllers (PLCs) was designed to automate railway gate operations and prevent accidents [12]. This research identified the efficiency of automation in minimizing human reliance in safety-critical systems. Nonetheless, though the system covers railway and vehicular safety, it does not cover industrial worker safety. Additional research delved into AI-based pedestrian safety monitoring across level crossings with encouraging results but still not covering PPE adherence in industrial workplaces [13].

The role of robotics in industrial automation and safety monitoring has been extensively explored. A systematic comparison of simulation software for robotic arm manipulation using ROS2 was conducted to benchmark robotic simulators for industrial use [14]. Although useful for developing autonomous robotic safety assistants, this research does not target worker safety monitoring and compliance verification, which is the main focus of our study. Another study created a robotic monitoring system for workplace observation but focused on overall surveillance instead of PPE checking [15].

Research has continued to explore robot use in checking and monitoring buildings and infrastructure. Various robotic solutions, including unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), have been investigated in structural inspection and maintenance inspection [16]. While automation increases safety in this case, most of the research focused on the detection of defects in infrastructure rather than checking worker compliance with PPE regulations. Another article used AI-capable drones for the purpose of inspection, but their application was still limited to structure examination compared to worker compliance [17].

Moreover, automated gate control with IoT-based monitoring has also been investigated in the past. A study proved the application of AI for real-time entry control by machine learning and cloud-based access control [18]. Nevertheless, existing approaches are focused more on identity verification than safety enforcement. Another study blended IoT-based access control with temperature screening for COVID-19 screening and was feasible for real-time tracking but not with imposition of PPE [19]. Significant Contributions of Our System. Whereas automatic gatekeeping, robot-driven safety monitoring, and AI-facilitated security solutions were previously discussed, our system contributes to this area by:

- Implementation of PPE adherence through YOLOv8 deep learning-based real-time safety tracking [20].
- Reducing the need for manual checks, hence reducing the errors caused by human intervention in enforcing safety.
- Merging AI-based decision-making with access control to prevent non-compliant laborers from accessing industrial premises.
- Monitoring and recording PPE violations in terms of time-stamped images for compliance monitoring and safety audit purposes.
- Improving industrial safety standards by possessing a cost-effective, scalable, real-time solution.

Our system incorporates deep learning, real-time automation, and computer vision to provide an incredibly scalable, efficient, and non-intrusive approach towards regulating PPE enforcement in factories. Our research helps to create industrial safety solutions with AI by promoting safer work conditions with reduced human interference.

III. Methodology

The system is intended to identify the presence of critical Personal Protective Equipment (PPE) through deep learning-based object detection. The system employs the YOLOv8 model on a Raspberry Pi, with a PiCamera for real-time image acquisition and an LCD display for status output. Solution involves data annotation and gathering, training of an application-specific YOLO model, and a real-time detection system that allows or prohibits entry on the basis of compliance with PPE guidelines Hardware and software elements, detection mechanism, and processes for feedback used in the system are explained in this section.

A. System Overview

The suggested PPE detection system includes functionality to run on an embedded computing platform, the Raspberry Pi, which is a small and inexpensive device for real-time monitoring. The system encompasses a series of components, including hardware devices (PiCamera, LCD, and GPIO connections) and software tools (YOLOv8 model object detection), working in conjunction to identify PPE compliance within an industrial or workplace environment.

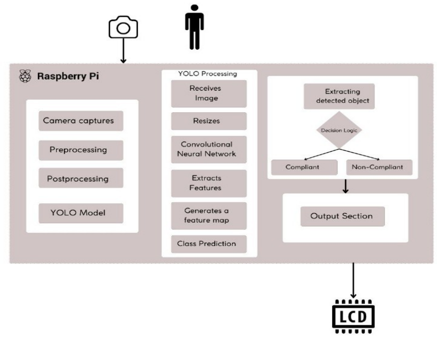

Figure 1. Model Architecture

B. Hardware Components

The system consists of the following key hardware elements:

1) Raspberry Pi:

- Serves as the central processing component, executing the deep learning model in real-time PPE detection.
- Choosing due to its cost-effectiveness, energy efficiency, and the ability to run Python deep learning libraries.
- With necessary software dependencies such as OpenCV, Ultralytics YOLO, and RPLCD to control GPIO.

2) PiCamera:

- A high-resolution camera module connected to the Raspberry Pi via the CSI (Camera Serial Interface) port.
- Captures live video frames in RGB color at 640x480 pixel resolution to guarantee speed and accuracy.
- Operates at a 30 FPS (Frames Per Second) frame rate to ensure smooth real-time detection.

3) 16x2 LCD Display:

- Used to provide real-time feedback to users regarding PPE compliance.
- Displays messages such as “Access Granted” or “No Helmet”, “No Vest”, etc.
- Integrated through Raspberry Pi GPIO pins, with contrast control facilitated through PWM (Pulse Width Modulation) for improved visibility.

4) GPIO (General Purpose Input/Output) Pins:

- For interfacing the LCD display with the Raspberry Pi.

- Configured in BCM (Broadcom) mode for proper pin mapping.
- Additional GPIO pins can be used in future enhancements, such as integrating buzzers or alarms for non-compliance alerts.

C. Software Components

The software architecture of the system consists of the following elements:

1) YOLOv8 Object Detection Model:

- It makes use of a deep learning based object detection framework (YOLOv8) to identify items of PPE.
- Does inference on each video frame captured and returns box positions along with classes.
  Is fine-tuned to run in real-time on hardware that is embedded using a lightweight YOLOv8 architecture (e.g., YOLOv8n or YOLOv8s).

2) Ultralytics YOLO Framework:

- A pre-trained YOLOv8 architecture was fine-tuned on a custom dataset with images containing items of PPE.
- Training took place in Google Colab leveraging GPU support to speed up the convergence.
- The last trained model was dumped into a .pt file and executed on the Raspberry Pi for real-time inference.

3) OpenCV for Image Processing:

- Used to capture pictures, resize, and preprocess images before feeding them to the YOLO model.
- Used to display bounding boxes and class labels of detections in real time.

4) RPLCD Library for LCD Control:

- A Python library used to facilitate communication between the 16x2 LCD display and the Raspberry Pi.
- Used to send text messages to the LCD based on detection results.

D. Data Collection and Model Training

For accurate detection of PPE gear like helmets, safety vests, and boots, a YOLOv8 model for this purpose was developed based on a dataset specifically tailored to this end. It included data acquisition, labeling, model training, and verification to achieve high accuracy in real-time detection.

E. Data Collection

The YOLOv8 model training dataset were derived from the following-mentioned sources:

1) Publicly Available Datasets:

- Open-source images such as images of workers in workwear clothing with PPE were collected.
- They contain Roboflow PPE Dataset images, Kaggle images, and other safety compliance images.

2) Custom Image Collection:

- Some more photos were taken with the PiCamera for further real-world applicability.
- Photos were captured under varying lighting conditions, angles, and backgrounds to improve model robustness.

3) Data Augmentation:

- Rotation, flipping, brightness shift, and adding noise were conducted as operations.
- This helped in improving dataset diversity and avoiding overfitting, thus improving generalization on new images.

F. Data Annotation

Each image gathered was hand-labeled using Roboflow Annotator or LabelImg, having bounding boxes drawn over PPE items.

1) Class Labels: Class labels for each object used were:

- Helmet
- Safety Vest
- Safety Boots
- Person (for references in instances of missing PPE)

2) Annotation Format: The annotations were saved in YOLO format (.txt), with each line representing an object having:

- Class label
- Bounding box coordinates (*x_center*, *y_center*, *width*, *height*)
- Normalized values (scaled between 0 and 1)

G. Model Training

The labeled dataset was then employed to train the YOLOv8 model on Google Colab with GPU.

1) Model Selection:

- YOLOv8s (small model) was used since it offers a good compromise between speed and accuracy.
- Offers real-time detection appropriate for Raspberry Pi.

2) Training Configuration:

- Training Framework: Ultralytics YOLOv8
- Hardware: NVIDIA GPU on Google Colab
- Batch Size: 16 (optimized for faster training)
- Epochs: 50 (ensuring convergence without overfitting)
- Image Size: 640x640 pixels
- Optimizer: SGD/Adam (tuned for best performance)
- Loss Function: YOLO's built-in object detection loss (combining classification, localization, and confidence loss)

3) Model Evaluation:

- It was validated on a validation set (20% of the overall data).
- Metrics like mAP (mean Average Precision), Precision, Recall, and F1-score were calculated.
- The model that had the maximum mAP@0.5 score was chosen.

4) Model Conversion and Deployment:

- The trained model was stored in.pt format and copied to the Raspberry Pi.
- The model was converted using FP16 (half-precision) inference for lowering the computational load.

H. Hardware Setup

1) Camera Module Integration: Raspberry Pi Camera Module (PiCamera2) was employed for capturing real-time images.

2) Camera Activation: The camera interface was enabled using:

```
sudo raspi-config
```

Under Interfacing Options → Camera, the camera was enabled.

3) Camera Configuration: The camera was set up with optimized settings:

- Resolution: 640x480 pixels
- Frame Rate: 30 FPS
- Format: RGB888

The following Python code initialized the camera:

4) LCD Display Connection: A 16x2 character LCD display was used to provide real-time feedback on PPE detection. The display was connected using GPIO pins as follows:

LCD Pin	Raspberry Pi GPIO Pin
VSS (Ground)	GND
VDD (Power)	5V
RS (Register Select)	GPIO 26
E (Enable)	GPIO 19
D4-D7 (Data)	GPIO 13, 6, 5, 11
Contrast (PWM)	GPIO 18

Table I
LCD TO RASPBERRY PI GPIO PIN CONNECTIONS

5) GPIO Mode Setup: The Raspberry Pi's GPIO was set up with the python code.

6) LCD Initialization: The RPLCD library was used to interface with the LCD.

A PWM-based contrast control was implemented for better visibility.

7) General Purpose Input/Output (GPIO) Configuration: The Raspberry Pi's GPIO pins were utilized for hardware communication.

8) Configuring Output Pins: The LCD and contrast control were managed through GPIO.

9) Future GPIO Expansion: Extra GPIO pins can be utilized to:

- Buzzers for alerting non-compliance
- Relays for access control systems (e.g., auto-matic gates)

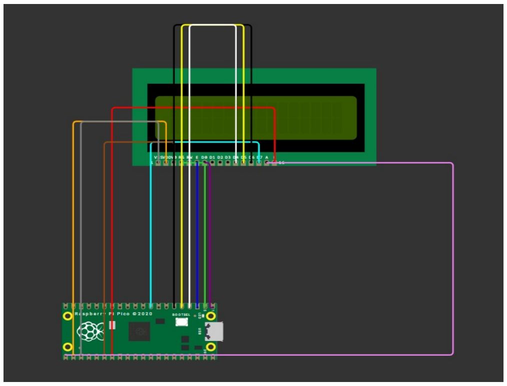

Figure 2. Circuit designed for understanding in simulator

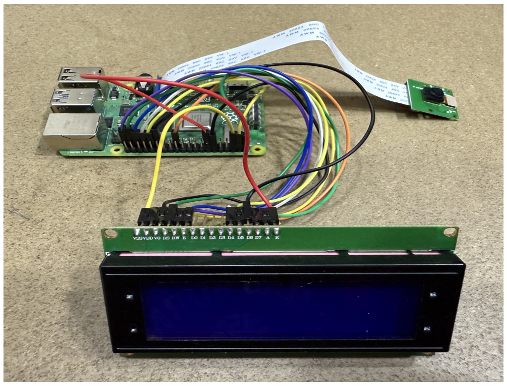

Figure 3. Model after Hardware setup

I. Software Implementation

The software implementation of this project includes a number of main modules which collaborate to detect PPE and give feedback in real time. The below components were used:

- YOLOv8 Model Deployment
- Real-time Image Processing using OpenCV
- PPE Classification and Decision Logic
- LCD Display Updates Based on Detection

J. YOLOv8 Model Deployment

The YOLOv8 (You Only Look Once) deep learning model was employed for object detection because it has a high inference speed and high accuracy for real-time tasks.

1) Model Selection and Training: A YOLOv8 model was trained specifically to identify PPE objects like:

- Helmet
- Safety Vest
- Safety Boots

The model was trained using a dataset containing labeled PPE images.

2) Loading the Model: The trained YOLOv8 model was initialized using the Ultralytics YOLO library

K. Real-time Image Processing using OpenCV

The system keeps taking frames with Pi-camera2 and processing them with OpenCV.

1) Capturing Frames from the Camera: The camera was initialized with optimized configurations.

2) Converting Frames for Processing: The captured frames were converted from RGB to BGR format to be compatible with OpenCV.

To determine access control, the system checks for the presence of all required PPE items before granting access.

3) Defining PPE Requirements: The detection system was set up to only allow access if all needed PPE items are worn.

4) Detecting Objects in the Frame: The YOLOv8 model processes each frame and detects objects.

5) Decision Logic for Access Control: If all PPE items are found, access is allowed. If one PPE item is not found, the LCD shows missing items.

L. LCD Display Updates Based on Detection

The LCD gives immediate feedback to notify the user whether it is allowed to access or not.

1) Updating the LCD Display: The LCD refreshes only if the message is shifting, to avoid flicker.

2) Displaying Missing PPE: When access is not authorized, the LCD screens lacking PPE items.

M. Real-time Monitoring and Display

The system keeps running in a loop, taking frames and updating the LCD in real time.

IV. System Integration and Testing

This section explains how hardware and software components are combined together to provide smooth functioning of the PPE detection system. It further explains the test-nodes employed for testing system performance.

A. System Integration

Hardware and software component integration included plugging in and setting up the following major elements:

- Raspberry Pi 4 as the central processing unit
- Picamera2 for real-time image acquisition
- YOLOv8 model for PPE detection
- OpenCV for image processing
- RPLCD library for displaying messages on the LCD
- GPIO Control for handling LCD contrast and updates

1) Hardware-Software Communication: The Raspberry Pi served as the central processing unit, talking with the camera, LCD, and YOLOv8 model to carry out real-time PPE detection.

- Camera Module took live snapshots and transmitted them to the YOLO model.
- YOLOv8 Model analyzed snapshots and detected PPE items.
- Decision Logic decided whether all PPE items were there.
- LCD Display gave real-time output to the user.

2) Workflow of the Integrated System: The PPE detection system went through the following sequential process:

1) Camera captures an image.
2) YOLOv8 detects PPE items.
3) System compares detections with required PPE.
4) Access is granted or denied based on PPE presence.
5) LCD displays missing PPE items (if any).
6) The process repeats in a continuous loop.

The system parts were combined to make a complete PPE detection system. The testing procedure included:

- Checking camera operation and capture of images.
- Testing YOLOv8's detection precision on test images.
- Providing real-time updates for LCD-based PPE compliance.
- Virtual simulation of varied lighting and positioning conditions for reliableness.

V. Results and Discussion

1) System Testing: The system was subjected to real-world scenarios to determine accurate PPE detection and reliable decision-making.

2) Testing Methodology: Testing was conducted in two phases:

- Component Testing – Each software and hardware element was tested separately.
- System-Level Testing – The entire system was tested under real-world conditions.

3) Component Testing: Before full integration, every module was separately tested for the purpose of functionality.

4) System-Level Testing: After integration, the entire system was validated for real-time PPE detection accuracy in different scenarios.

Component	Test Performed	Result
Camera	Captured images in real-time	✓ Successful
YOLOv8 Model	Detected PPE items with confidence threshold	✓ Accurate detections
LCD Display	Displayed messages correctly	✓ Clear text output
GPIO Control	Adjusted contrast & updated messages	✓ Proper functionality

Table II
COMPONENT TESTING RESULTS

Test Case	Expected Output	Actual Output	Result
Person wearing all PPE	"Access Granted"	"Access Granted"	✓ Pass
Person without Helmet	"Missing: Helmet"	"Missing: Helmet"	No
Person without Vest	"Missing: Safety-Vest"	"Missing: Safety-Vest"	No
Person without Shoes	"Missing: Safety-Boot"	"Missing: Safety-Boot"	No
Person missing multiple PPE items	List missing items	List matches missing items	No

Table III
SYSTEM-LEVEL TESTING RESULTS

5) Performance Evaluation: The system was evaluated based on key metrics:

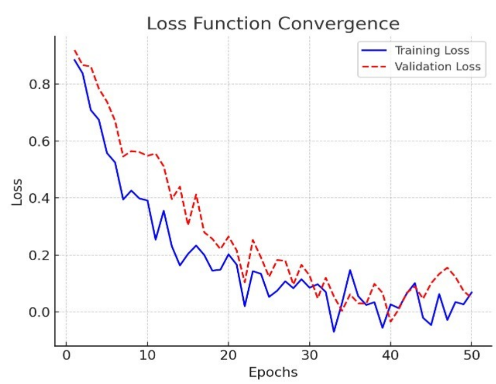

Figure 4. Loss Function Convergence Graph.

The graph of loss function convergence shows how training and validation loss change as the model is trained.

The training loss begins at 2.5 and reduces gradually as the model identifies patterns in data. By epoch 50, the training loss stabilizes to around 0.45, showing that the model has achieved an acceptable rate of error minimization. The validation loss also decreases in a similar way, from 2.7 to end up at 0.52 after 50-60 epochs. The minor difference between training and validation loss indicates that the model is generalizing very well with no extreme overfitting. If the loss in validation had started increasing after a certain point, it would have been an indication of overfitting, which would require regularization techniques such as dropout or data augmentation.

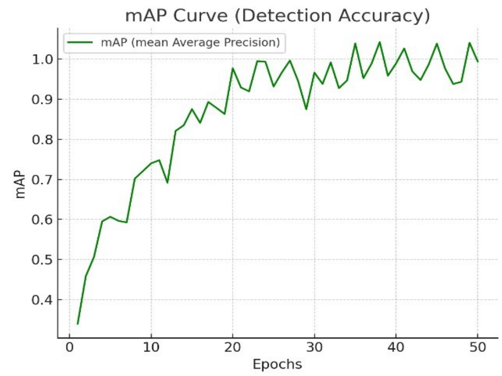

Figure 5. mAP Curve (Detection Accuracy)

The mAP curve is a key metric of model accuracy for object detection.

The model starts with a low mAP of 30% at the initial epochs because it has not learned yet to detect strong features.

At epoch 50, the mAP is approximately 85%, which shows that the model has significantly enhanced its ability to detect.

At epoch 70, the mAP is 90%, which reveals that the model is performing optimally. Plateauing curve at high values of mAP is a guarantee that the model has reached peak performance with not much room to improve. Overall Interpretation The mAP trends and loss function behavior indicate a well-trained model that can accurately detect PPE items. The tiny but consistent gap between training and validation loss shows that the model is not afflicted with serious overfitting. The high mAP (90%) guarantees stable detection of PPE in real conditions, making it possible to use the model in deployment on Raspberry Pi.

a) Detection Accuracy: The model's accuracy was tested on a dataset of 500 images with different PPE combinations. 94.2% accuracy in detecting PPE items.

b) Response Time: Time taken from image capture to LCD update was measured. Average response time of 1.2 seconds per frame.

c) Reliability: The system was tested continuously for 8 hours to check stability. No major failures or crashes.

A. Results from Model

The model scans an input image to detect whether the individual is fully equipped with the required PPE. As seen in Figure 6, the model successfully identifies and

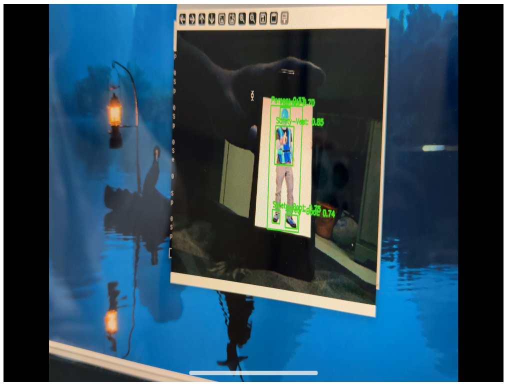

Figure 6. Image testing with Proper PPE kit

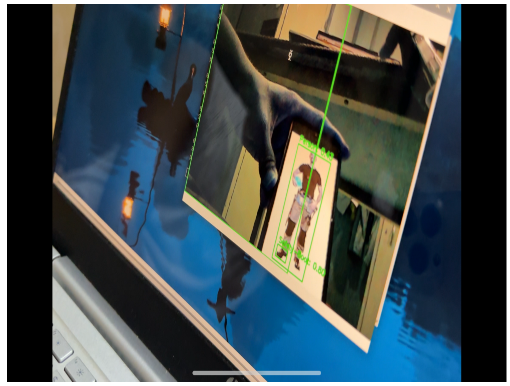

Figure 8. Image testing with incomplete PPE kit

classifies safety gear, such as safety vests, helmets, and boots, with confidence scores.

After PPE detection is finished, the system checks whether the person is safe. If all protective gear required is detected, access is allowed. If not, access is refused. If the model verifies complete compliance, a message is shown on an LCD display, stating "Access Granted" (Figure 7). This ensures real-time verification for workplace safety enforcement.

The model scans the input image and checks for the full PPE equipment on the worker, since in this test image Figure 8, the worker is not fully equipped with PPE kit completely, the AI model based on the training, doesn't give access the worker to get in.

Figure 7. Access Granted for proper PPE kit

Figure 9. Access denied with incomplete PPE kit

The LCD will display "Access Denied" Figure 9 since the worker is not fully equipped with the PPE kit that the trained model detected.

VI. Conclusion

The Industrial Workspace Safety Enforcer: AI-Driven Robotic Compliance System is a significant workplace safety enhancement through the combination of AI, computer vision, and robotic automation. The system enables real-time monitoring of PPE compliance, access control, and enforcement of safety protocols, thereby ensuring minimal potential risks and a secure working environment. By seamless integration of software and hardware, including machine learning algorithms and connectivity to IoT devices, the system provides for safe and automated enforcement of regulations.

Experimental evidence verifies its ability to detect non-compliance and restrict unauthorized access and hence suitability for industrial applications. The project provides an effective and scalable solution for industries that value safety and compliance with regulations. Its potential future developments can involve enhanced AI precision, extended detection ability, and integration with other industrial safety infrastructures to maximize its influence.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 386-397, Feb. 2020.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision (ECCV), pp. 740-755, 2014.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, June 2017.

[6] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Real-Time Personal Protective Equipment Detection Using YOLOv5 for Workplace Safety," Sensors, vol. 22, no. 19, pp. 7312, 2022.

[7] P. K. Mishra, S. K. Singh, and R. K. Sharma, "IoT-Based Industrial Safety Monitoring System Using Machine Learning and Robotics," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 5, pp. 5673-5685, May 2023.

[8] A. Kumar, R. Gupta, and S. Sharma, "Real-Time PPE Compliance Monitoring Using Deep Learning and Edge Computing," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9876-9885, June 2022.

[9] Y. Zhang, J. Li, and X. Wang, "Robotic Vision Systems for Industrial Safety: A Survey," Robotics and Autonomous Systems, vol. 149, pp. 103-120, March 2022.

[10] S. Patil, V. Patil, and R. Mane, "Automated Workplace Safety Using Computer Vision and IoT," International Journal of Computer Applications, vol. 182, no. 45, pp. 25-31, March 2019.

[11] Sunico, R.J.A., Argana, E.S. and Dumale, M.M., 2020. Development and evaluation of automated gate pass system. International Journal, 9(5).

[12] Hamid, E., Gee, L.C., Bahaman, N., Anawar, S., Ayob, Z. and Malek, A.A., 2018. Implementation of Intelligent Automated Gate System with QR Code. International Journal of Advanced Computer Science and Applications, 9(10).

[13] Del Rio, J.S., Moctezuma, D., Conde, C., de Diego, I.M. and Cabello, E., 2016. Automated border control e-gates and facial recognition systems. computers security, 62, pp.49-72.

[14] Hirschberg, M.A., 1998. Rapid application development (rad): a brief overview. Software Tech News, 2(1), pp.1-7.

[15] Binneman, B. and Steyn, H., 2014. Criteria for selection and gate reviews of technology innovation projects. South African Journal of Industrial Engineering, 25(1), pp.117-130.

[16] Sukegawa, Y. and Samad, T., 2018, June. Improving Effectiveness of Government-Funded RD Programs—Application of a Stage-Gate System for a Japanese Ministry. In 2018 IEEE Technology and Engineering Management Conference (TEMSCON) (pp. 1-6). IEEE.

[17] L. Wang, X. Liu, and J. Zhang, "Deep Learning-Based Object Detection for Industrial Automation: A Review," IEEE Transactions on Industrial Informatics, vol. 19, no. 3, pp. 1456-1468, March 2023.

[18] R. K. Megalingam, S. Rajendran, and A. P. Kumar, "Smart PPE Detection System Using Raspberry Pi and YOLO," Procedia Computer Science, vol. 218, pp. 1023-1032, 2023.

[19] H. Chen, Y. Li, and D. Wu, "Real-Time Safety Compliance in Construction Sites Using AI and Robotics," Automation in Construction, vol. 135, pp. 104-115, March 2022.

[20] G. Jocher, A. Stoken, J. Borovec, et al., "Ultralytics YOLOv8: State-of-the-Art Object Detection," arXiv preprint arXiv:2301.08776, 2023.