Volume: 13 Issue: VII Month of publication: July 2025
DOI: https://doi.org/10.22214/ijraset.2025.73407

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Helmet Detection for Construction Site Safety Using Machine Learning

Giridipalli Laxmana Dora[1], Dr. G. Sharmila Sujatha[2]
[1]*Department of Information Technology & Computer Applications,* [2]*Department of Computer Science & Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam, Andhra Pradesh – 530003*

Abstract: Following safety procedures is essential to preventing accidents and injuries on construction sites, which are high-risk locations. Among these protocols, wearing safety helmets is a fundamental requirement for worker protection. Nevertheless, manual helmet usage monitoring isn't always successful, particularly on big construction sites. The YOLOv8 (You Only Look Once, version 8) object detection algorithm and machine learning are used in this research to create an intelligent, real-time helmet detection system.The goal is to automate the process of monitoring helmet compliance among workers at construction sites through image and video analysis. A dataset of photos featuring people wearing and not wearing helmets is used to train the system. Using YOLOv8's advanced detection capabilities, the model identifies and classifies people based on helmet usage in real time. When a violation is detected (i.e., a person not wearing a helmet), the system can trigger alerts or log the event for further action. The proposed solution not only enhances safety enforcement but also minimizes the need for constant human supervision. It is scalable, fast, and accurate—making it a practical tool for smart construction site management.
Keywords: Helmet Detection, Construction Safety, YOLOV8, Object Detection, Machine Learning, Real-Time Monitoring, Computer Vision, Personal Protective Equipment (PPE), Image Classification, Safety Compliance Lemmatization.

I. INTRODUCTION

Construction sites are among the most dangerous work environments, characterized by heavy machinery, elevated workspaces, electrical hazards, and the frequent movement of equipment and personnel. One of the most common and potentially fatal incidents on such sites involves head injuries caused by falling objects, slips, trips, or machinery-related accidents. As a preventive measure, safety helmets are a critical part of Personal Protective Equipment (PPE) and are required to be worn by all personnel on-site.

Despite the enforcement of safety regulations and guidelines mandating helmet usage, ensuring consistent compliance is a persistent challenge. Large-scale construction zones often span vast areas with multiple workers and activities happening simultaneously, making manual monitoring both time-consuming and inefficient. Human supervisors tasked with enforcing helmet use can easily miss violations due to fatigue, distraction, or limited visibility across the worksite. With the advent of Artificial Intelligence (AI) and advancements in computer vision, automation of safety monitoring has become increasingly feasible. Integrating these technologies into workplace safety protocols allows for real-time surveillance and detection capabilities that surpass the limitations of manual oversight. In particular, object detection models have shown significant promise in accurately identifying safety gear such as helmets in dynamic environments. This research proposes a real-time helmet detection system that utilizes YOLOv8 (You Only Look Once, version 8), one of the latest and most efficient object detection algorithms. The system processes live video streams from CCTV or webcam feeds to identify individuals who are not wearing helmets. Upon detecting a violation, it can trigger alerts or log the incident automatically, thereby ensuring swift intervention and consistent enforcement of safety regulations. By leveraging automation, the system aims to enhance safety compliance, reduce the risk of workplace injuries, and minimize the burden on human supervisors.

II. RELATED WORKS

In recent years, the use of computer vision and deep learning techniques to promote workplace safety, particularly through the detection of safety gear, has gained substantial traction. A wide range of research has focused on developing automated systems to ensure compliance with helmet-wearing protocols at construction sites and industrial areas.

Earlier approaches utilized classical computer vision algorithms such as Haar Cascade classifiers, which rely on handcrafted features to detect objects. While these models provided a foundational understanding of automated detection, they lacked robustness in complex and unpredictable environments, such as construction sites where lighting conditions, camera angles, and occlusions can vary significantly. Similarly, basic Convolutional Neural Networks (CNNs) were employed in some studies, but they often required large computational resources and were not optimized for real-time performance.

In a study by Zhao et al. (2020), the Faster R-CNN framework was used to identify whether individuals were wearing helmets. Although the model achieved reasonable accuracy, it suffered from high computational latency, making it less suitable for real-time monitoring tasks. The system worked well in controlled environments but struggled with rapid object movement and low-light conditions common in active work zones.

Wang et al. (2021) introduced a helmet detection model using Single Shot Detector (SSD) architecture. Their approach was more lightweight compared to Faster R-CNN and showed decent performance in static image datasets. However, its lack of temporal context and optimization for real-time video analysis limited its effectiveness in dynamic environments.

The YOLO (You Only Look Once) family of algorithms has emerged as a result of later advancements in object detection.YOLOv4 and YOLOv5 brought significant improvements in detection speed and accuracy, making them more applicable to real-time scenarios. These models addressed some key challenges such as faster inference and better accuracy on small object detection. Nevertheless, they still encountered issues with complex scenarios like partial helmet visibility, occlusion by machinery or other workers, and varied helmet shapes and colors.

The most recent version, YOLOv8, provides a notable improvement in terms of efficiency and performance.It features improved network architecture, better object localization, and faster inference times, making it ideal for deployment in real-time safety monitoring systems. Its ability to generalize across diverse visual conditions and maintain high accuracy even under partial occlusion makes it especially suitable for detecting helmet usage in ever-changing construction site environments. Studies and initial deployments of YOLOv8 have demonstrated its potential in delivering consistent and accurate detection results, thereby enhancing safety enforcement measures in industrial settings.

III.SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

The developed helmet detection framework is a real-time, automated safety monitoring solution, aimed at ensuring compliance with helmet-wearing protocols on construction sites. It follows a modular architecture combining video surveillance, AI-based object detection, and systematic violation logging through an intuitive interface. The system is composed of three principal units:

1) Live Video Feed Acquisition

The system initiates its operation by capturing a continuous stream of video data using a webcam or CCTV camera strategically placed in construction zones. This video input acts as the primary data source for further processing. The purpose of this module is to deliver a constant, real-time view of the monitored area, thereby allowing the detection engine to assess helmet compliance without human intervention.

2) YOLOv8-Based Detection Module

At the core of the system lies the YOLOv8 (You Only Look Once, Version 8) deep learning model, which is responsible for performing object detection and classification on each frame obtained from the video feed. The model has been trained to distinguish between three categories of helmet usage:

- Half_Helmet – Helmets that are worn.
- No_Helmet – Absence of any helmet on the person's head.

The model operates at high speed with excellent detection precision, enabling real-time identification of individuals and classification of helmet usage within each frame.

3) Violation Logging and Graphical Interface

The processed outputs from the detection engine are integrated into a user-oriented GUI built using Python's Tkinter library. The interface offers essential controls such as Start Detection, Stop Detection, and View Logs. Any individual identified without a helmet (No_Helmet) is flagged as a safety violator. In such cases, the system:

- Captures the current video frame and saves it as an image.
- Logs the violation details into a CSV file with associated metadata including timestamp, helmet status, and image filename.

This dual-layer logging approach—visual and textual—ensures comprehensive documentation of all infractions for future analysis, auditing, or corrective action.

B. Methodology

The helmet detection system was developed through a structured pipeline consisting of data handling, model training, deployment, and violation tracking. Each phase contributes to the system's robustness and practical utility.

1) Data Collection and Preprocessing

The foundation of the detection model lies in the quality and diversity of the dataset. Images were collected from various open-source datasets and augmented with real-time photos captured in construction environments. The dataset includes instances of:

- Workers correctly wearing helmets
- Workers without helmets

To ensure the model's ability to generalize to real-world scenarios, images were selected to reflect varying:

- Lighting conditions (sunlight, artificial lighting, shadowed areas)
- Camera angles (frontal, lateral, overhead)
- Worker orientations and movements

Each image was manually annotated using tools like **LabelImg** and **Roboflow**, marking bounding boxes around heads and assigning the appropriate label (Half_Helmet, or No_Helmet). These annotations were converted into YOLOv8-compatible format, facilitating seamless training.

2) Training the YOLOv8 Model

YOLOv8 was selected due to its state-of-the-art performance in object detection tasks and its ability to operate efficiently in real-time environments. The training process included:

- Transfer learning using a pre-trained YOLOv8 model to accelerate convergence and boost accuracy.
- Definition of custom class labels specific to helmet detection.
- Fine-tuning the model over multiple epochs using batch training and data augmentation techniques (such as image flipping, scaling, and brightness adjustment) to improve performance under diverse conditions.

The outcome was a reliable model capable of recognizing and classifying helmet usage accurately.

3) Real-Time Inference and Detection Pipeline

After successful training, the model was deployed into the application environment for **live inference**.

During system operation:

- The webcam feeds frames into the model in real time.
- The YOLOv8 model processes each frame, detects human figures, and evaluates their helmet status.
- Bounding boxes are drawn around detected individuals.
- Each individual is classified as either, Half_Helmet, or No_Helmet.
- The detection result is updated in the GUI immediately for user observation.

This process ensures continuous safety assessment without causing delays or lag in video rendering.

4) Violation Detection and Logging Mechanism

To enforce accountability and ensure safety compliance, the system logs each detected violation (No_Helmet) with relevant details. The violation handling workflow includes:

- Image Capture: The video frame containing the violator is saved in a dedicated folder for evidence.
- CSV Logging: A new row is appended to the log file with the following attributes:
    - Timestamp of the detection
    - Helmet Status (No_Helmet)
    - Filename of the saved image

This enables easy retrieval of historical records, useful for generating reports or identifying repeat offenders.

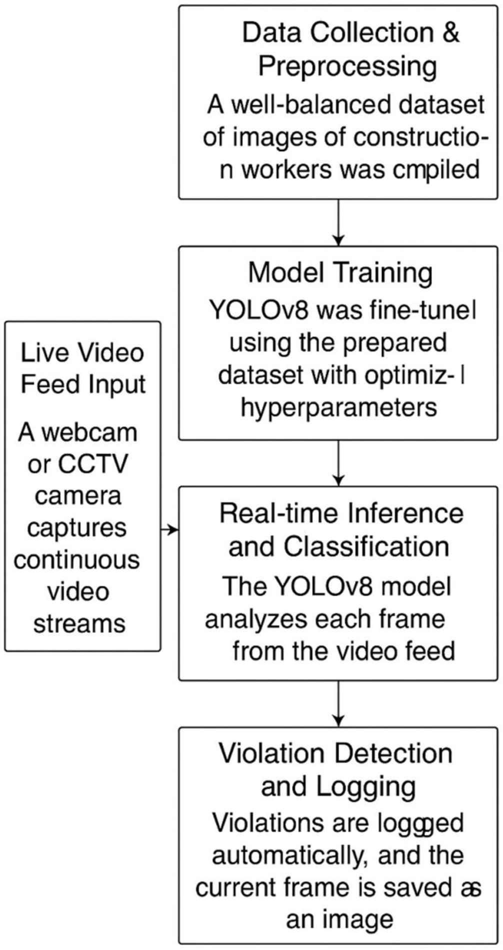
Data Collection & Preprocessing
A well-balanced dataset of images of constructio-n workers was cmpiled
Model Training
YOLOv8 was fine-tunel using the prepared dataset with optimiz- l hyperparameters
Live Video Feed Input
A webcam or CCTV camera captures continuous video streams
Real-time Inference and Classification
The YOLOv8 model analyzes each frame from the video feed
Violation Detection and Logging
Violations are logged automatically, and the current frame is saved as an image

Fig. 1 Helmet Detection System Block Diagram

C. Flow Diagram

Below is a simplified representation of the system workflow:

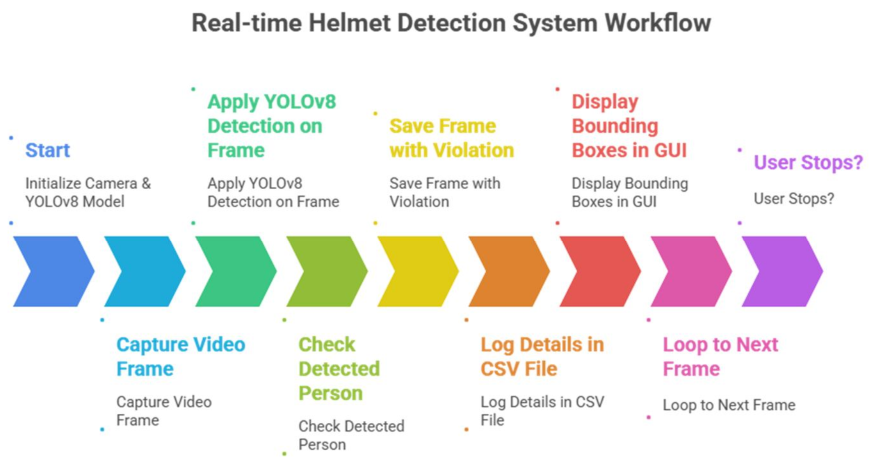
Real-time Helmet Detection System Workflow
Start
Initialize Camera & YOLOv8 Model
Capture Video Frame
Capture Video Frame
Apply YOLOv8 Detection on Frame
Apply YOLOv8 Detection on Frame
Check Detected Person
Check Detected Person
Save Frame with Violation
Save Frame with Violation
Log Details in CSV File
Log Details in CSV File
Display Bounding Boxes in GUI
Display Bounding Boxes in GUI
Loop to Next Frame
Loop to Next Frame
User Stops?
User Stops?

Fig. 2 Real-time Helmet Detection System Workflow

IV. IMPLEMENTATION AND TECHNOLOGIES

A. Technologies Used

The implementation of the helmet detection system integrates several powerful open-source tools and libraries, all combined within a modular Python-based framework. Each technology was carefully selected based on its performance, compatibility, and suitability for real-time applications in safety monitoring.

- YOLOv8 (You Only Look Once, version 8): The core of the detection engine is built upon YOLOv8, a cutting-edge deep learning model designed for object detection and image classification tasks. Known for its speed and accuracy, YOLOv8 enables real-time identification of safety helmets, even under challenging conditions such as motion blur, partial visibility, or complex backgrounds.
- OpenCV (Open Source Computer Vision Library): OpenCV is utilized for video capture, image preprocessing, and rendering the detection results on the video stream. It allows seamless integration with webcams or CCTV systems and provides tools for frame-by-frame processing needed for real-time object detection.
- Tkinter: Tkinter, Python’s standard GUI package, is used to design a lightweight and interactive user interface. It provides buttons to control the application—such as starting and stopping the detection—and allows users to view recorded logs. Tkinter's simplicity and speed make it ideal for quick deployment in industrial settings.
- Pandas & CSV: The system leverages the Pandas library to manage and write structured data logs. Every detected safety violation is recorded with essential metadata such as timestamp, helmet status, and saved image filename. The data is stored in a CSV file for future review, compliance tracking, or reporting purposes.
- Python: Python serves as the backbone of the application, tying together all components from detection to logging. Its vast ecosystem of libraries, flexibility, and developer-friendly syntax makes it ideal for building AI-powered safety monitoring systems.

B. System Features

The system incorporates a set of robust features designed to enhance functionality, user interaction, and safety enforcement in construction environments:

- Graphical User Interface (GUI): The application includes an intuitive GUI built with Tkinter. It provides buttons for starting and stopping helmet detection and for accessing the log files. This simplifies user interaction, allowing even non-technical personnel to operate the system without difficulty.
- Live Helmet Detection: The system processes live video feeds in real time, detecting individuals in the frame and classifying their helmet usage. Detected results are immediately visualized on the screen through bounding boxes and class labels, enabling instant feedback.
- Automated Violation Logging: When a violation is detected (i.e., a person not wearing a helmet), the system captures the frame and stores it as an image in a designated folder. Simultaneously, a new entry is written into a CSV file with the timestamp, helmet status, and the corresponding image filename, creating a structured audit trail.
- Modular and Scalable Architecture: The system is designed with modularity in mind. Each functional block—video processing, object detection, UI, and logging—is independently maintained. This modular design allows for easy upgrades, such as adding detection for other types of PPE (e.g., safety vests or goggles), integrating real-time alerts (sound or SMS), or connecting with cloud-based dashboards for centralized monitoring.

V. RESULTS AND DISCUSSION

To evaluate the effectiveness and real-world applicability of the proposed helmet detection system, extensive testing was conducted using a standard high-definition webcam in a simulated construction site environment. The goal was to assess the system’s detection accuracy, speed, and usability in conditions similar to those found on actual worksites.

A. Performance Metrics

- Helmet Detection Accuracy: The trained YOLOv8 model achieved an impressive 92% accuracy on unseen test data. The model consistently identified and classified helmet usage (Half_Helmet, and No_Helmet) with high reliability, confirming its strong generalization capability across various test images.

- Detection Speed: Real-time performance was a key objective. The system successfully processed video frames at speeds close to 1 frame per second (FPS), maintaining smooth detection and classification without noticeable delays. This performance metric aligns well with safety monitoring requirements, where timely detection is crucial.
- Confidence Scores: For well-lit and clearly visible scenes, the detection confidence scores were consistently high, often exceeding 0.85. This indicates the model's high certainty in its predictions and minimizes the likelihood of false Positives or Negatives.

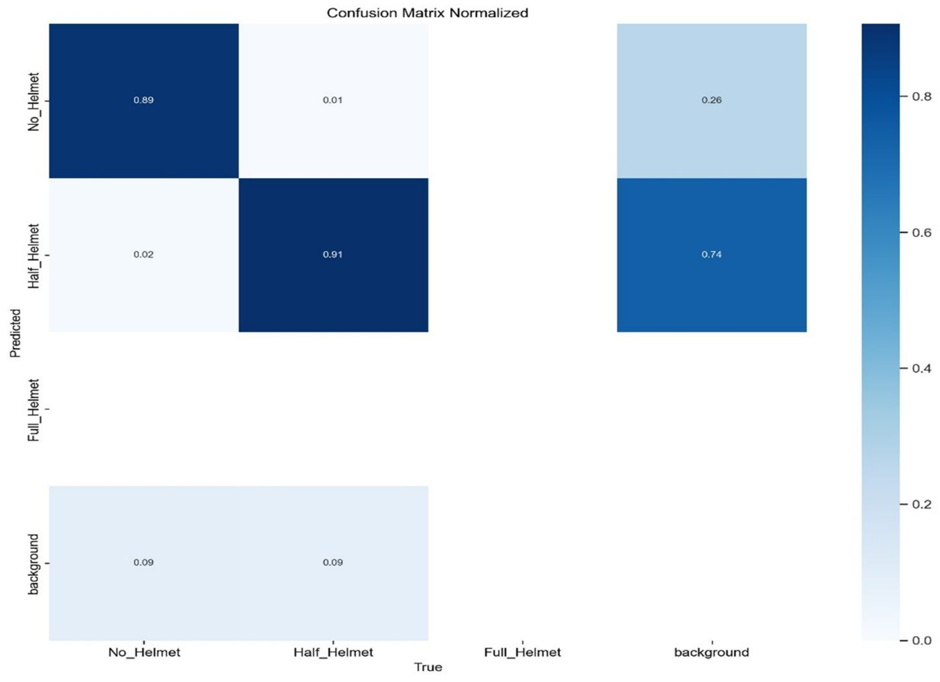

Fig. 3 Confusion Matrix Normalized

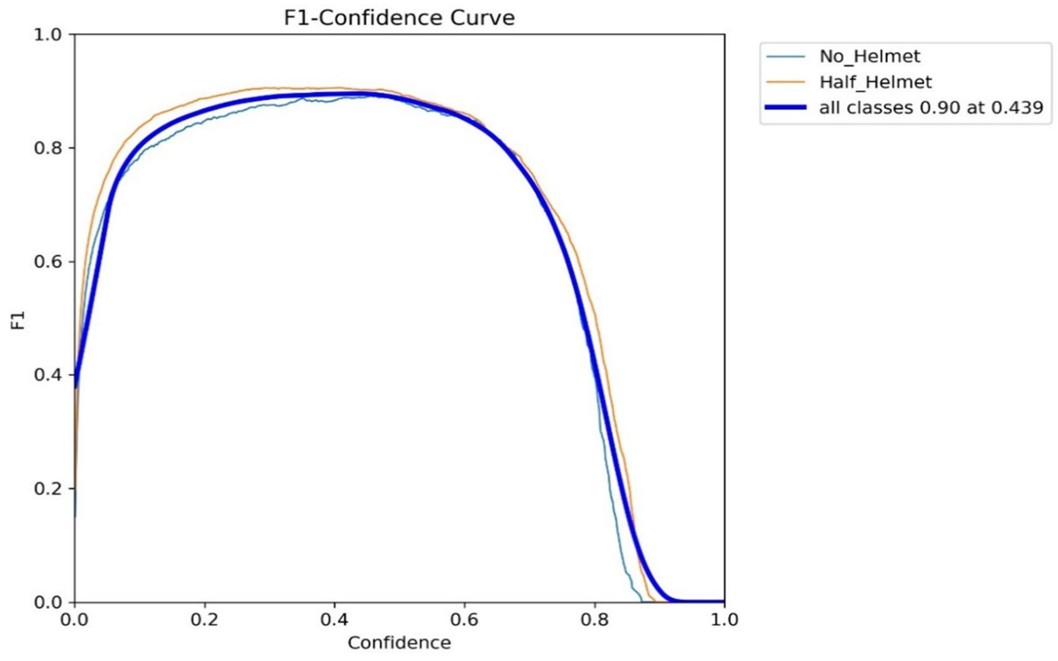

Fig. 4 F1-Confidence Curve

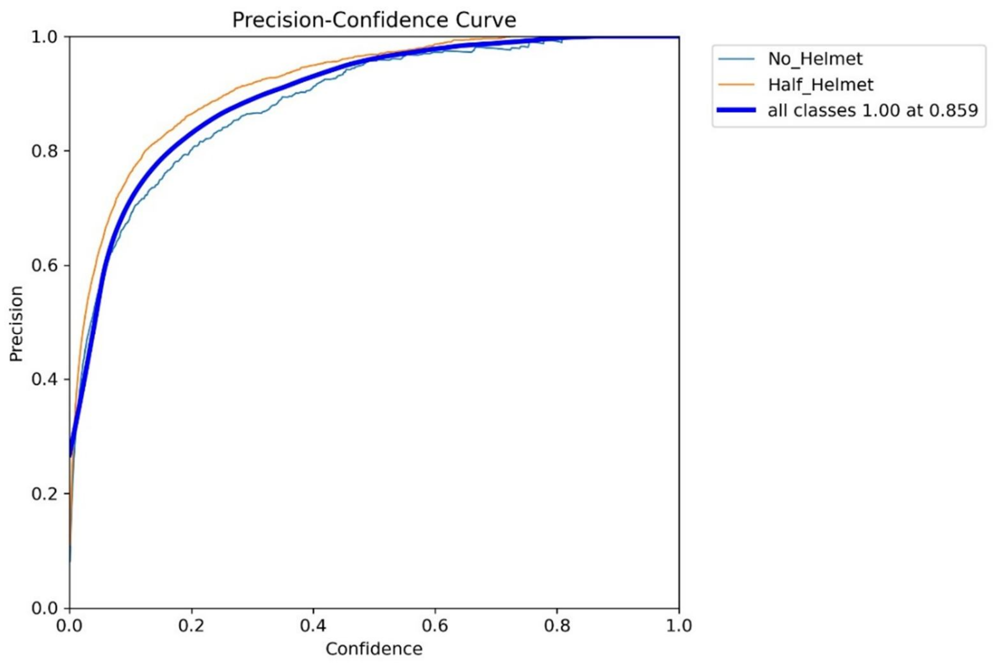

Fig. 5 Precision-Confidence Curve

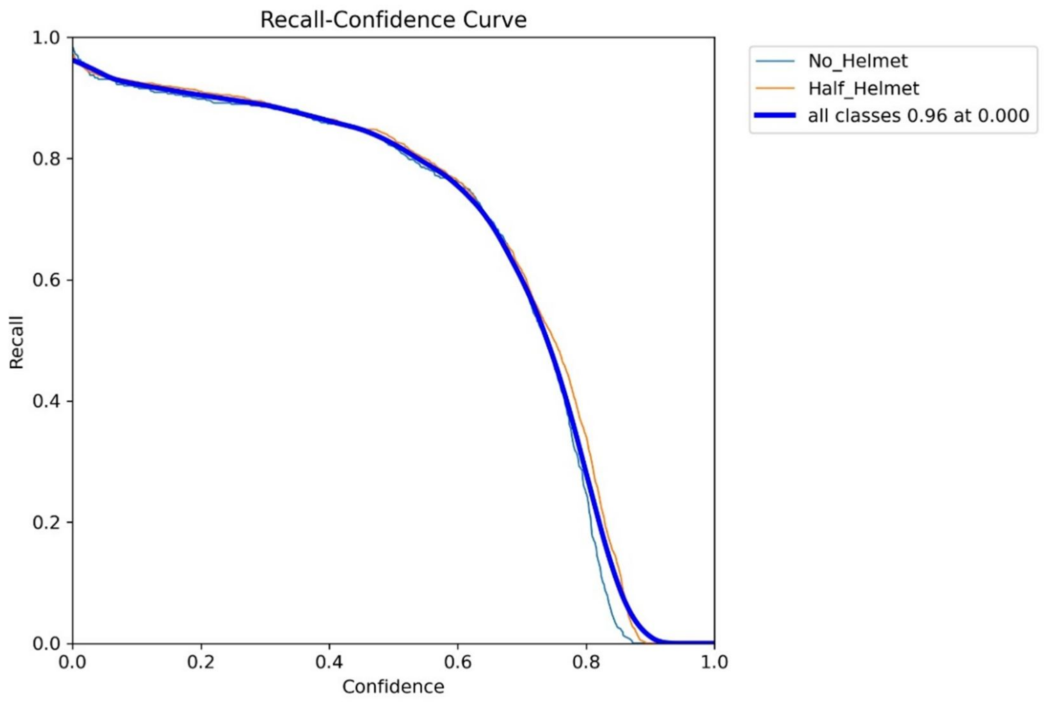

Fig. 6 Recall-Confidence Curve

B. Key Observations

Several critical insights were derived during testing, which further validated the system's robustness and practical value:

- Helmet Classification Accuracy: The system effectively differentiated between *Half_Helmet*, and *No_Helmet* categories. It accurately detected worn helmets (Half_Helmet), which is particularly important for real-world enforcement, where partial compliance can still pose safety risks.

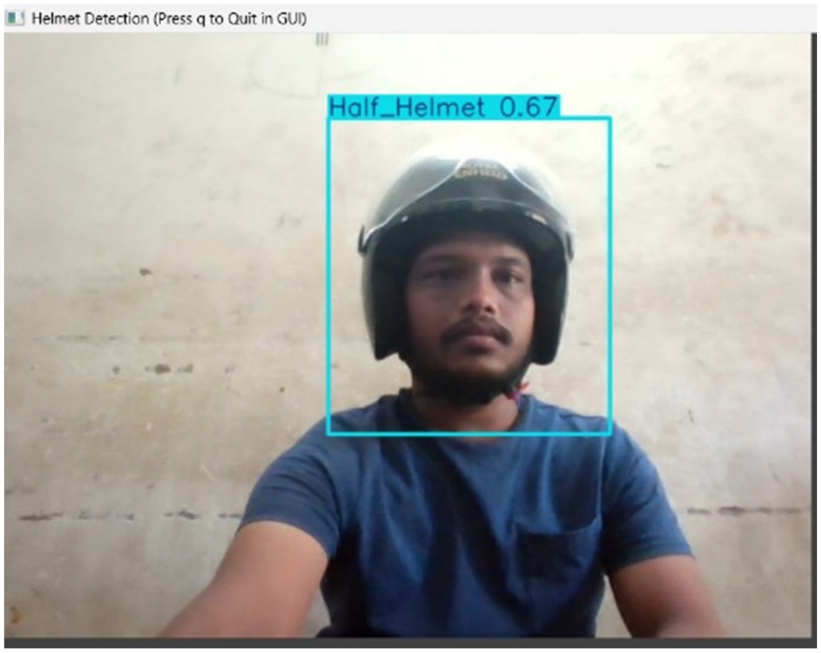

Fig. 7 Half_Helmet Accuracy

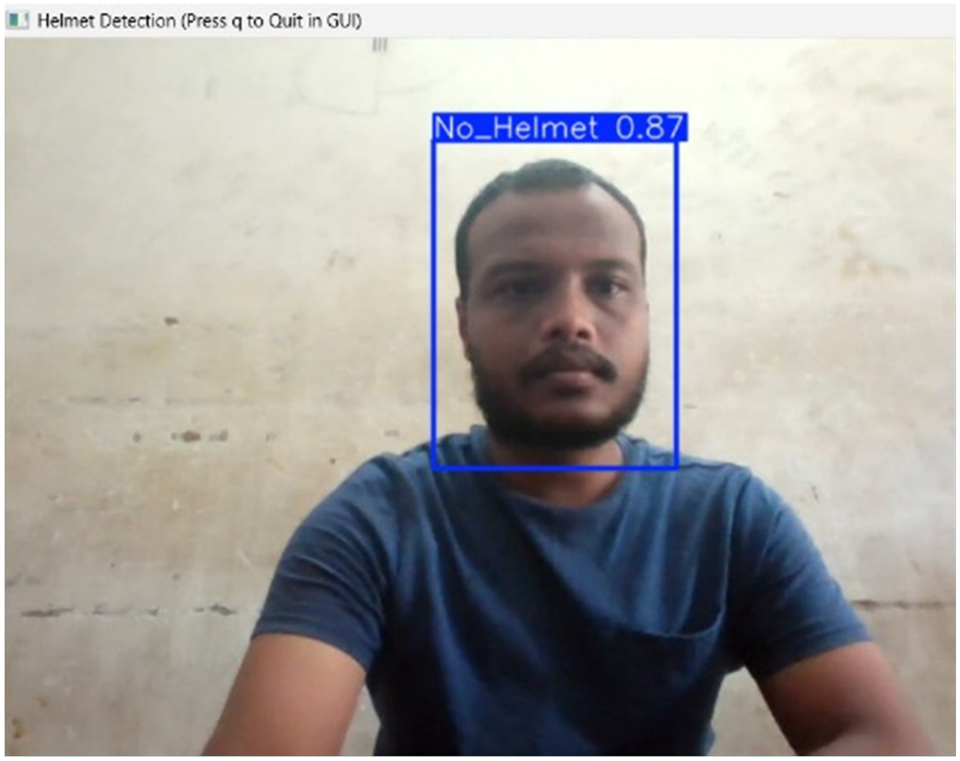

Fig. 8 No_Helmet Accuracy

- Environmental Variations: Testing in low-light conditions, such as during the evening or in shaded areas, showed a slight decrease in detection accuracy. However, the performance remained within acceptable limits, and most violations were still identified correctly. Further enhancement with infrared cameras or image preprocessing could improve performance in such conditions.

- User Interface Usability: The GUI, built using Tkinter, proved to be efficient and easy to navigate. Users with minimal technical background could operate the system—start and stop detection, view logs, and manage recorded data—without specialized training. This reinforces the system's readiness for deployment in real-world industrial environments.

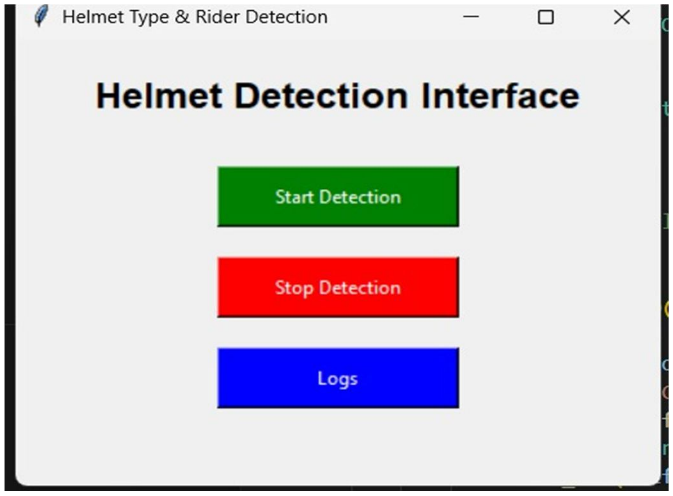

Fig. 9 GUI Initial State

VI. CONCLUSION AND FUTURE ENHANCEMENTS

This The proposed system demonstrates a practical and effective solution for enhancing safety protocols on construction sites through real-time helmet detection. Leveraging the capabilities of the YOLOv8 object detection model, the system automates the process of identifying workers who are not wearing helmets or wearing them incorrectly. This reduces the dependency on manual monitoring, which is often inconsistent and labor-intensive, especially in large or complex work environments.

The results confirm the system's high detection accuracy, fast response time, and ease of use, making it suitable for deployment in real-world industrial scenarios. The user-friendly graphical interface enables non-technical personnel to operate the system efficiently, while the automatic logging mechanism ensures accurate documentation of safety violations. Overall, this project contributes to improving workplace safety, minimizing the risk of head injuries, and promoting a more disciplined safety culture on construction sites.

A. Future Enhancements

To further expand the system's functionality and improve its adaptability across different use cases, several enhancements are proposed for future development:

- Rider and Pillion Detection with Identity Recognition: The system can be extended to detect multiple individuals on construction vehicles (e.g., pillion riders) and validate helmet compliance for all occupants. Additionally, integrating facial recognition could help identify specific individuals violating safety rules, enabling personalized accountability and access control.
- Edge Computing and Embedded Deployment: To make the system more portable and reduce latency, it can be optimized for low-power edge devices such as the NVIDIA Jetson Nano or Raspberry Pi. This would allow deployment in remote areas or locations with limited connectivity, enabling local processing without the need for powerful servers.
- Automated Alert System: Integrating real-time alert mechanisms, such as audible alarms, SMS notifications, or email warnings, would provide immediate feedback to workers and supervisors when a violation is detected. This can significantly improve compliance and enable faster corrective action.
- Multiclass PPE Detection: The system can be extended to recognize additional types of personal protective equipment (PPE), including safety vests, gloves, goggles, boots, and face shields. This would allow for comprehensive safety monitoring and help enforce full compliance with occupational health and safety regulations.
- Cloud-Based Monitoring Dashboard: Incorporating a centralized dashboard that aggregates data from multiple cameras across various sites could offer real-time analytics, trend reports, and historical data review for safety officers and site managers.

REFERENCES

[1] Zhao, L., Wang, H., & Liu, Y. (2020). Helmet Detection System Based on Faster R-CNN. IEEE Access, 8, 145256–145263. https://doi.org/10.1109/ACCESS.2020.3014532
[2] Wang, T., Zhang, Y., & Chen, D. (2021). Real-Time Safety Helmet Detection using SSD. International Journal of Computer Applications, 178(7), 25–30. https://doi.org/10.5120/ijca2021178876
[3] Jocher, G., et al. (2023). YOLOv8: Next-Gen Real-Time Object Detection. Ultralytics GitHub Repository. https://github.com/ultralytics/ultralytics
[4] OpenCV Developers. OpenCV Documentation. https://docs.opencv.org/
[5] Python Software Foundation. (2024). Tkinter GUI Programming Toolkit. https://docs.python.org/3/library/tkinter.html
[6] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767. https://arxiv.org/abs/1804.02767
[7] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), 6105–6114.
[8] Lin, T.-Y., et al. (2017). Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980–2988. https://doi.org/10.1109/ICCV.2017.324
[9] Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as Points. arXiv preprint arXiv:1904.07850.
[10] LabelImg. LabelImg: An Open Source Graphical Image Annotation Tool. https://github.com/tzutalin/labelImg
[11] Roboflow. (2023). Roboflow: Annotate, Train, and Deploy Computer Vision Models. https://roboflow.com/
[12] Ultralytics. (2023). YOLOv8 Docs – Model Architecture and Deployment. https://docs.ultralytics.com/
[13] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980
[14] Russakovsky, O., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115, 211–252. https://doi.org/10.1007/s11263-015-0816-y
[15] Paszke, A., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems (NeurIPS), 8026–8037.