IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# WORKERS SAFETY MONITORING SYSTEM

[1]Gangotri, [2]Bhavana S, [3]M Pooja, [4]Madhushree U, [5]Sreenivas A

[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor

[1]Dept of Computer Science,

[1]Navodaya Institute Of Technology, Raichur, India

*Abstract:* The Worker Safety Monitoring System is a computer vision–based application designed to monitor worker safety using image and video analysis. The system uses the YOLOv8 deep learning model along with OpenCV to detect workers and identify the use of personal protective equipment (PPE) in real time. Images or video streams are processed through a Flask-based backend, where object detection is performed and results are displayed on a web dashboard. The system operates without a database and focuses on real-time safety monitoring, making it efficient, lightweight, and suitable for industrial safety applications.

**Index Terms** - Worker Safety Monitoring, YOLOv8, Object Detection, Personal Protective Equipment (PPE), Computer Vision, OpenCV, Flask Framework, Image Processing, Video Processing, Real-Time Monitoring, Web Dashboard

## I. INTRODUCTION

Worker safety is a critical concern in industries such as construction, manufacturing, and mining, where workers are exposed to hazardous environments. Ensuring the use of proper Personal Protective Equipment (PPE) like helmets and safety vests is essential to reduce workplace accidents. Traditional safety monitoring methods rely on manual supervision, which is time-consuming and prone to human error.

The Worker Safety Monitoring System addresses this challenge by using computer vision and deep learning techniques to automatically detect worker safety compliance. The system uses the YOLOv8 object detection model along with OpenCV to analyze images and video streams in real time. A Flask-based web application is used to process the input and display detection results through a web interface. This system provides an efficient and automated approach for monitoring worker safety without the need for data storage or alert mechanisms.

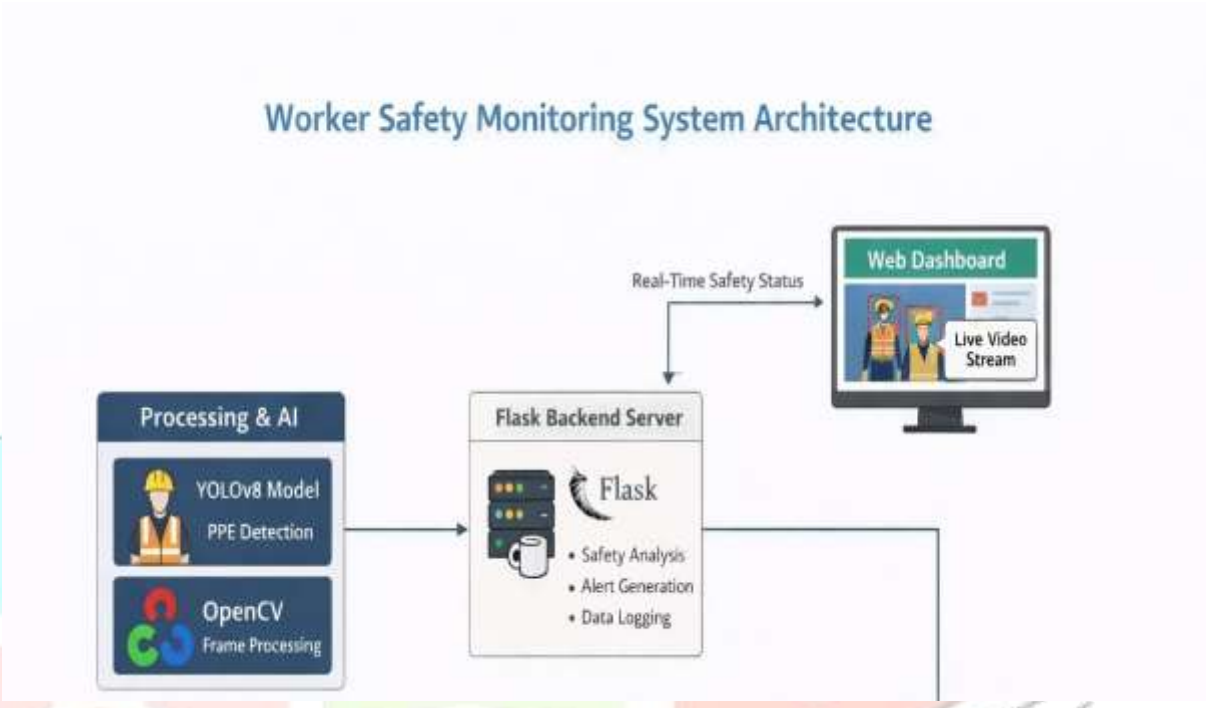
With the rapid advancement of **Artificial Intelligence and Computer Vision**, automated safety monitoring systems have become a reliable solution to enhance workplace safety. This project focuses on developing a **Worker Safety Monitoring System** using **YOLOv8 (You Only Look Once)** for real-time object detection and **Flask** as a web framework to deploy the system.

The proposed system automatically detects workers and verifies the presence of required PPE through live video streams or uploaded images. By integrating deep learning with a web-based interface, the system enables real-time monitoring, quick identification of safety violations, and improved compliance with safety regulations. This approach helps industries reduce accidents, improve safety standards, and ensure a safer working environment.

## II. DRAWBACKS OF EXISTING SYSTEM

- Limitations of the Existing Safety Monitoring System
- Problems in the Traditional Worker Safety Monitoring Approach
- Challenges of Manual Safety Monitoring Systems
- Issues with Conventional Worker Safety Practices

## III. METHOD




- The **Worker Safety Monitoring System** is designed using a **simple client–server architecture** that integrates computer vision and deep learning techniques to monitor worker safety in real time. The system focuses on detecting Personal Protective Equipment (PPE) such as helmets and safety vests using a trained YOLOv8 model and displaying the results through a web interface.

### 1. Processing & AI Layer

This layer performs the core safety detection tasks.

- **YOLOv8 Model**
  The YOLOv8 deep learning model is used to detect workers and identify whether required safety equipment (helmet, vest) is present or not. It processes individual frames and produces detection results with bounding boxes.
- **OpenCV (Frame Processing)**
  OpenCV is used to capture video frames from the input source (webcam or video file) and preprocess them before passing them to the YOLOv8 model for inference.

**2. Flask Backend Server**

The Flask server acts as the middleware between the AI processing layer and the frontend.

- Receives video frames from OpenCV
- Sends frames to the YOLOv8 model for PPE detection
- Processes detection results
- Streams the processed video with bounding boxes to the frontend
- Handles routing and real-time communication with the web interface

**3. Web Dashboard (Frontend)**

The web dashboard provides a visual interface for monitoring worker safety.

- Displays **live video stream** processed by the system
- Shows bounding boxes around detected workers and PPE
- Allows users to visually verify safety compliance in real time
- Built using **HTML, CSS, and JavaScript**

**Data Flow Description**

1. Video frames are captured using OpenCV.
2. Frames are passed to the YOLOv8 model for PPE detection.
3. Detection results are sent to the Flask backend.
4. Flask streams the processed video to the web dashboard.
5. The user views real-time safety monitoring through the browser.

**Architecture Advantages**

- Simple and lightweight design
- Real-time PPE detection
- No dependency on database or alert systems
- Easy to deploy and run locally
- Suitable for academic and demonstration purposes

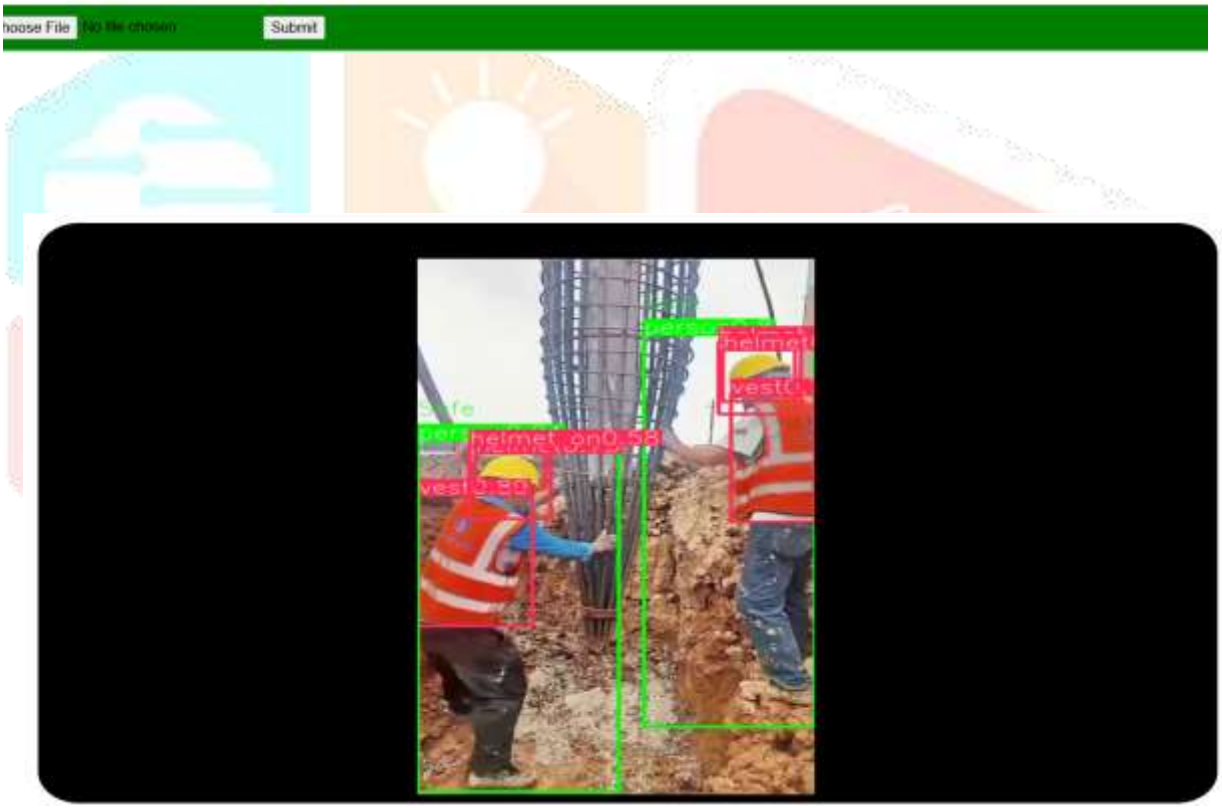
The proposed Worker Safety Monitoring System follows a systematic approach that integrates deep learning–based object detection with a web-based deployment framework. The overall methodology consists of the following steps:

1. **Data Collection and Preparation:** Images and video frames of workers with and without Personal Protective Equipment (PPE) are collected. The data is preprocessed and annotated to label different PPE items such as helmets, safety vests, gloves, and masks.

2. **Model Selection and Training;** The **YOLOv8 object detection model** is selected due to its high accuracy and real-time performance. The model is trained on the annotated dataset to detect workers and identify PPE compliance.
3. **PPE Detection Process:** The trained YOLOv8 model processes live video streams or uploaded images to detect workers and classify PPE items. Bounding boxes are drawn around detected objects, indicating whether safety equipment is present or missing.
4. **System Integration:** The trained model is integrated into a **Flask web application**, which acts as the backend server. Flask handles user requests, processes input data, and returns detection results.
5. **Web Interface and Visualization:** A web-based user interface displays the processed images or video streams with detection results in real time. This allows safety personnel to easily monitor compliance and identify violations.
6. **Testing and Evaluation:** The system is tested under different conditions such as varying lighting and camera angles to evaluate accuracy, speed, and reliability. Performance is analyzed to ensure effective real-time monitoring.

## III. RESULTS

The Worker Safety Monitoring System successfully detects workers and identifies the presence of personal protective equipment (PPE) such as helmets and safety vests using the YOLOv8 model. The system processes uploaded images and video streams in real time and displays detection results through a web interface. Bounding boxes are accurately drawn around detected objects, clearly indicating safety compliance. The system demonstrates reliable performance with minimal processing delay and stable operation through the Flask backend. Overall, the results show that the system effectively supports real-time visual monitoring of worker safety without the need for data storage or alert mechanisms.


## V.CONCLUSION

The Worker Safety Monitoring System provides an effective and automated approach for monitoring worker safety using computer vision and deep learning techniques. By utilizing the YOLOv8 object detection model along with OpenCV and a Flask-based web application, the system is able to detect workers and verify the use of personal protective equipment such as helmets and safety vests in real time. The system processes images and video streams efficiently and displays results through a simple web interface. This project demonstrates that real-time safety monitoring can be achieved without the need for data storage or alert mechanisms, making the system lightweight, practical, and suitable for industrial safety monitoring applications.

**ACKNOWLEDGEMENT**

We take this opportunity to present our thanks to all those who had really acted as lightening pillars to enlighten our way through this project that has led to successful and satisfactory completion of this study. Our sincere thanks to our beloved principal **Dr. M.V. Mallikarjuna**. Navodaya Institute of Technology for his kind co-operation during our entire course. We are really grateful to our **HOD DR. M.N. Faruk** for providing us with an opportunity to undertake this project in this university and providing us with all the facility. We extend our sincere and heartfelt thanks to our esteemed guide **Prof. Sreenivas A** for her exemplary guidance, monitoring and constant encouragement throughout the course at crucial junctures and for showing us the right way. We would like to thank other faculty members also. Lastly, We are thankful to all those, particularly the various friends who have been instrumental in creating proper, healthy and conductive environment and including new and fresh innovative ideas for me during the seminar. Your help, it would be extremely easy for us for preparing the project report in time bound frame work.

**REFERENCES**

[1] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.

[2] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLOv8: Ultralytics Object Detection Framework. *Ultralytics Documentation*.

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

[4] Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*, 740–755.

[5] Flask Development Team. (2023). *Flask Web Framework Documentation*. [Online]. Available: https://flask.palletsprojects.com/

[6] OpenCV Team. (2023). *Open Source Computer Vision Library*. [Online]. Available: https://opencv.org/

[7] Brownlee, J. (2019). *Deep Learning for Computer Vision*. Machine Learning Mastery.

[8] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.

[9] Occupational Safety and Health Administration (OSHA). (2022). *Personal Protective Equipment (PPE) Standards*. [Online]. Available: https://www.osha.gov/

[10] Zhang, Y., Wang, L., & Liu, H. (2021). Real-Time Safety Helmet Detection Based on Deep Learning. *Journal of Safety Science*, 134, 105–113.