BEHAVIOR AND PERSON SENSING SYSTEM BASED ON COMPUTER VISION

Mumtaz Zain Abdullah[1], Guan Yunchong[2], Yoppy Yunhasnawa[3]

[1,3] Department of Information Technology, Politeknik Negeri Malang, Soekarno-Hatta Street No. 9, Malang 65141, Indonesia
[2] College of Computer Science and Technology, Shenyang Aerospace University, No.37 Daoyi South Street, Shenbei New District, Shenyang, Liaoning, China

[1]zainabdullah2126@gmail.com, [2] y.c.guan@foxmail.com, [3] yunhasnawa@polinema.ac.id

Abstract

In modern industrial environments, effective supervision of worker attendance and activity is essential to ensure safety and optimize productivity. Inadequate monitoring can lead to increased workplace accidents and reduced operational efficiency. To address these challenges, we propose an intelligent monitoring system leveraging computer vision techniques, implemented using PyTorch and the YOLO (You Only Look Once) object detection framework. The system utilizes video input from CCTV or cameras installed in the workspace and performs real-time analysis through several key modules: (i) Worker Counting, which detects and counts the number of employees present; (ii) Activity Detection, which classifies workers as actively working or idle (e.g., sleeping or using mobile phones); (iii) Fall Detection, which identifies and alerts the server in case of worker falls; and (iv) Absent Detection, which tracks and flags workers who leave the monitored area. Detection results and relevant video segments are automatically archived in local storage, enabling retrospective review and continuous monitoring. By integrating these modules, the proposed system aims to enhance workplace safety, support real-time remote supervision, and improve overall operational effectiveness in industrial settings.

Keywords: worker situation, computer vision, monitoring system, YOLO, real-time detection, local storage.

1. Introduction

In the modern industrial landscape, ensuring operational efficiency and maintaining worker safety are critical priorities for enterprises (Ajerla et al., 2019). Workers represent essential assets in production, and their absence, inefficiency, or unsafe behavior can significantly impact productivity and profitability (Bo, 2023). Manual monitoring methods, often subjective and inconsistent, struggle to provide accurate, real-time insights and can lead to delays in responding to workplace incidents (Parasuraman et al., 2018). To overcome these limitations, this study proposes an intelligent worker monitoring system based on computer vision and deep learning. The system leverages the YOLO (You Only Look Once) object detection model within the PyTorch framework to automate the recognition of worker activities from live video streams captured via CCTV or IP cameras (Redmon & Farhadi, 2018; Singh et al., 2021; Zhao et al., 2024).

The proposed system is designed to identify and classify worker behaviors, including active working, sleeping, using mobile phones, falling, and leaving the workspace without authorization (Pereira, 2024; Raza et al., 2022). The integration of a Flask-based web interface allows real-time visualization of detection results, while detected events are logged and stored locally as videos and images for audit purposes (Mokayed et al., 2023). This approach not only reduces dependency on human supervision but also enables continuous, objective, and non-intrusive monitoring (Muheidat et al., 2018; Khekan et al., 2024). The system architecture is modular, supporting future extensions such as recognizing new types of activities, monitoring additional zones, or integrating more advanced predictive algorithms (Gowsikhaa et al., 2014; Sanjalawe et al., 2025).

Beyond operational monitoring, the system aims to strengthen workplace safety by including automatic fall detection that triggers alerts to a central server for immediate intervention (Yuan et al., 2006; Vishwakarma et al., 2007). This capability is crucial for minimizing response time to accidents, thereby reducing potential injuries. Additionally, unauthorized absence detection helps maintain discipline by identifying workers who leave designated areas during work hours (Chen et al., 2022). By generating reliable data on worker presence and activity, the system also supports better decision-making and workflow optimization (Fan et al., 2017; Khekan et al., 2024; Mok et al., 2016).

From a technical perspective, the system utilizes Python, PyTorch, and OpenCV for video processing and detection logic (Vasavi et al., 2023). Hardware requirements include standard CCTV or IP cameras connected to a local computer, optionally supported by GPUs for accelerated inference (Sturman et al., 2020). A lightweight web interface built with Flask displays key metrics, including worker counts, fall alerts, and historical logs (Zivkovic et al., 2025). Detected incidents are saved automatically with structured filenames, simplifying future audits.

As industries move toward smart manufacturing, adopting real-time, data-driven monitoring solutions becomes increasingly strategic (Li & Song, 2023). Future enhancements could explore boundary testing and robust error handling to improve system reliability (Wijaya et al., 2022), integrate predictive models to anticipate hazardous behaviors, and add personalized alert features to enhance user experience. Strengthening data security will also be critical to protect sensitive monitoring data. Overall, this project contributes an adaptable, automated approach to worker supervision, aligning technological advancement with workplace safety and operational excellence.

2. Method

The research methodology was structured to develop an intelligent worker monitoring system, as illustrated in Figure 1. The process begins with CCTV-based data collection, followed by preprocessing stages including frame extraction, labeling, and augmentation. A custom YOLOv8 model is then trained and validated. Finally, the trained model is deployed into a real-time monitoring system to evaluate its performance in detecting, tracking, and classifying worker activities within the industrial environment.

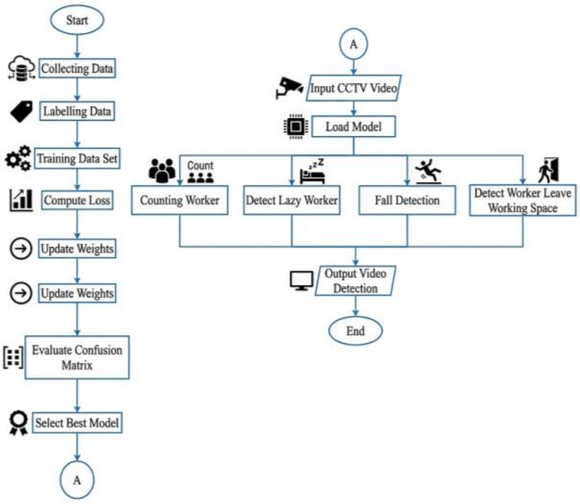

Figure 1. Research Methodology and System Workflow

2.1 Data Collection and Acquisition Technique

The process of collecting and getting data is very important for this research because the quality and variety of the data directly affect how well the YOLOv8 model works and how well it can be used in different situations. The goal of this part is to get good visual information about workers in an industrial setting so that we can build a system that can monitor workers in real time.

Most of the data came from CCTV cameras placed in a controlled industrial area. These cameras were set up at fixed positions to record workers doing different things under various lighting and environmental conditions. Each video shows several different actions, which are used for labelling and training the model. The activities of the workers were grouped into four main types:

- Working: when workers are doing their jobs or using machines.
- Sleeping: when workers are resting or lying down.
- Using Phone: when workers are looking at their phones.
- Idle: when workers are standing or sitting but not doing anything.

In addition to the videos, some data from open-source sets like COCO and OpenImages, which include images of people, were used to help train the YOLOv8 model earlier. This helps the model better understand features and prevents it from learning too much from just the specific data we collected.

The acquisition and preparation of data were conducted through several systematic steps:

- Video Recording: Continuous video footage was captured from CCTV cameras at a frame rate of 25–30 frames per second (fps). Each video covered a duration of several minutes, representing diverse worker behaviours.
- Frame Extraction: The recorded video data were decomposed into individual image frames, which were used as the primary dataset. The total number of frames extracted can be formulated as Equation (1).

$$Nf = Fr \times Td \quad (1)$$

where:
Nf = total number of frames extracted,
Fr = frame rate (frames per second),
Td = duration of the video in seconds.

- Annotation and Labeling: Each extracted frame was manually annotated using *LabelImg* following the YOLO format. The annotation file (.txt) contains normalized coordinates for each bounding box and the corresponding class label as expressed by Equation (2).

$$(class, x_{center}, y_{center}, width, height) \quad (2)$$

- Data Preprocessing: Before model training, preprocessing was applied to improve dataset quality and diversity. This included image resizing to 640×640640 \times 640640×640 pixels, pixel normalization, and several augmentation techniques such as horizontal flipping, brightness adjustment, and rotation. The augmentation transformation can be represented mathematically as Equation (3).

$$I_{aug} = f(I_{orig,}\alpha, \beta, \theta) \quad (3)$$

where I_{aug} is the original image and α, β, and θ represent parameters for brightness, contrast, and rotation, respectively.

2.2 Compute Loss

The YOLOv8-based Intelligent Worker Monitoring System uses a thorough loss function during training in order to maximize the model's ability to identify and categorize worker behaviors. During training, the model's internal parameters are adjusted to minimize prediction errors based on the mathematical foundation provided by the loss function.

Bounding box regression loss, objectness confidence loss, and classification loss are the three primary components of YOLOv8's total loss function, which is stated in Equation (4).

$$L = \lambda_{box}L_{box} + \lambda_{obj}L_{obj} + \lambda_{cls}L_{cls} \quad (4)$$

where:
L is the total loss, L_{box} represents the error in bounding box localization, L_{obj} measures the confidence of object detection, and L_{cls} evaluates the accuracy of class prediction. The coefficients λ_{box}, λ_{obj}, and λ_{cls} are weighting factors used to balance the contribution of each term.

The bounding box loss (L_{box}) ensures that the predicted bounding boxes are spatially aligned with the ground truth. YOLOv8 uses the Complete Intersection over Union (CIoU) to evaluate both the overlap and geometric distance between predicted and actual boxes.

The resulting precision value of 1.00 at 0.954 confidence demonstrates excellent performance of the proposed system. This means that the trained YOLOv8 model can accurately detect and classify worker behaviors with minimal errors, making it highly suitable for real-time industrial monitoring applications where accuracy and reliability are crucial.

$$L = 1 - IoU + \frac{\rho^2(c_p c_g)}{d^2} + \alpha v \quad (5)$$

Here, IoU denotes the intersection over union, $\rho^2(c_p c_g)$ is the Euclidean distance between the centers of the predicted and ground truth boxes, d is the diagonal length of the smallest enclosing box, and αvpenalizes aspect-ratio inconsistencies. Minimizing this term helps the model generate bounding boxes that closely match the actual position of workers in the frame.

The objectness loss (L_{obj}) determines how well the model can distinguish between the presence and absence of a worker in a given region. This component uses Binary Cross-Entropy (BCE) as shown in Equation (6).

$$L_{obj} = -[y \log(\hat{o}) + (1 - y) \log(1 - \hat{o})] \quad (6)$$

where:
$y \in \{0,1\}$ indicates whether an object is present (1) or not (0), and $\hat{o}$ is the predicted probability that an object exists in that region. A lower objectness loss reflects the model's increasing confidence in accurately detecting workers.

The classification loss (L_{cls}) evaluates how accurately the model predicts the correct activity class Working, *Sleeping*, *Using Phone*, *Idle*, or *Fall*. It is calculated using Binary Cross-Entropy or Focal Loss to handle class imbalance, as expressed in Equation (7).

$$L_{cls} = -\sum_{c=1}^{C} [y_c \log(\hat{p}_c) + (1 - y_c) \log(1 - \hat{p}_c)] \quad (7)$$

C represents the number of activity classes, y_c is the actual label, and $\hat{p}$is the predicted probability for each class. This term encourages the model to produce more accurate activity classifications across all worker behaviors.

In the context of this project, the decreasing total loss value demonstrates that the model effectively learns spatial and semantic patterns of worker behaviors captured by CCTV cameras. The compute loss function, therefore, plays a crucial role in achieving a robust and accurate real-time monitoring system capable of enhancing workplace safety and productivity in industrial environments.

2.3 Worker Counting

The Worker Counting module is a key component of the YOLOv8-based Intelligent Worker Monitoring System. Its main function is to detect and count the number of workers present in the monitored area using video input from CCTV cameras.

The counting process begins with object detection by the YOLOv8 model, which identifies workers and marks them with bounding boxes. Each detected worker is then assigned a unique ID using a tracking algorithm (DeepSORT) to prevent duplicate counts when the worker moves across frames. The number of workers in each frame is calculated as Equation (8).

$$N_t = \sum_{i=1}^{k} \delta_i \quad (8)$$

where:
N_t is the total number of detected workers, and δ_i=1 if the i^{th} detection is classified as "Worker."

To evaluate counting performance, the accuracy is measured by comparing predicted and actual worker counts using the following Equation (9).

$$A_c = 1 - \frac{C_{pred} C_{true}}{C_{true}} \quad (9)$$

A high A_c value indicates that the system's counting result is close to the actual number of workers. In conclusion, the Worker Counting module enables real-time and accurate supervision of worker presence, supporting both safety monitoring and productivity analysis in industrial environments.

2.4 Lazy Worker Detection

The Lazy Worker Detection module is designed to identify workers who are inactive or performing non-productive activities within the monitored area. This feature is essential for maintaining productivity and ensuring efficient work performance in industrial environments.

Using the YOLOv8 model, the system detects and classifies worker activities into categories such as Working, Sleeping, Using Phone, and Idle. Detections belonging to *Sleeping* or *Using Phone* are categorized as lazy behavior. The classification process is supported by a confidence score generated by the model, where the final decision is determined by the highest probability value, as formulated in Equation (10).

$$L_{Lazy} = [P(S|x) \geq \tau] \vee [P(UP|x) \geq \tau] \quad (10)$$

where:

- L_{Lazy}= lazy worker indicator (1 = detected as lazy, 0 = not lazy),
- $P(S|x)$ = probability that the detected worker is sleeping,
- $P(UP|x)$ = probability that the worker is using a phone,
- τ = confidence threshold (typically set to 0.8).

When L_{Lazy} = 1, the system automatically records the detection event, saves the worker ID, timestamp, and frame evidence into the database, and optionally sends an alert to the monitoring dashboard.

In conclusion, the Lazy Worker Detection module enables real-time supervision of worker productivity by identifying unproductive behaviors accurately and efficiently. This capability supports management decisions for improving discipline, efficiency, and overall performance in industrial operations.

2.5 Fall Detection

The Fall Detection module is developed to automatically identify incidents where a worker falls within the monitored area. This module is crucial for improving workplace safety and enabling a rapid response to potential accidents.

Using the trained YOLOv8 model, the system continuously analyses live video streams from CCTV cameras to detect abnormal postures or body orientations associated with a fall. The model distinguishes the "Fall" class from normal working postures based on spatial features, bounding box aspect ratios, and body alignment.

The detection decision is made when the model's confidence for the *Fall* class exceeds a predefined threshold, formulated as Equation (11).

$$F_{det} = \begin{cases} 1, if\ P(Fall|x) \geq \tau \\ 0, if\ P(Fall|x) \geq \tau \end{cases} \quad (11)$$

where $P(Fall|x)$ is the predicted probability of a fall and τ is the confidence threshold (typically 0.8).

Once a fall is detected, the system automatically captures the event frame, saves it to local storage, and sends an alert to the monitoring server. This ensures that supervisors are immediately informed of the incident, allowing quick medical or safety intervention.

In conclusion, the Fall Detection module enhances the overall reliability of the monitoring system by providing real-time accident detection and alerting, thereby increasing worker safety in industrial environments.

2.6 Worker Leave Detection

The Worker Leave Detection module is developed to monitor and identify workers who exit the designated working area during operation. This feature supports real-time attendance supervision and helps maintain safety compliance within the monitored industrial space.

Using the YOLOv8 detection framework integrated with the DeepSORT tracking algorithm, each detected worker is assigned a unique tracking ID. The system defines a fixed Region of Interest (ROI) that represents the valid working area. When a worker's tracked position moves outside the boundary of this ROI, the system registers the event as a leave action. The detection condition is defined as follows:

$$L_{det} = \begin{cases} 1, if\ (x_p, y_p) \notin ROI \\ 0, if\ (x_p, y_p) \in ROI \end{cases} \quad (12)$$

where:

(x_p, y_p)represents the centroid of the worker's bounding box in the frame, and ROIROIROI is the defined boundary of the working area.

Once a leave event ($L_{det} = 1$) is detected, the system logs the worker ID, timestamp, and last known position into the database, while also updating the dashboard and triggering a notification if necessary.

In conclusion, the Worker Leave Detection module ensures continuous supervision of worker presence and contributes to both safety assurance and disciplinary monitoring, helping management identify unauthorized movements or absences in real time.

3. Results and Discussion

The YOLOv8-based worker monitoring model was evaluated through four testing scenarios, each representing a different activity detection module using specific data ratios for training and testing. These scenarios correspond to the four worker activity classes Working, Sleeping, Playing Phone, and Falling that were trained and tested on separate data distributions. The resulting detection accuracy for each scenario is presented in Table 1.

The YOLOv8 model achieved stable and acceptable accuracy levels across all activity categories, ranging between 72% and 79%. The highest accuracy was recorded in the Working class (79.42%), which indicates that the model performs effectively when detecting normal working postures and movements in stable lighting and environmental conditions.

The Sleeping class achieved an accuracy of 76.35%, showing that the model can identify inactive behavior with a reasonable level of precision, although certain postures resembling resting or bending may lead to minor misclassifications.

The Playing Phone activity yielded the lowest accuracy (72.18%) due to visual similarity between hand and head movements during phone use and those of normal working behavior. This suggests that future model optimization could focus on temporal feature extraction or higher-resolution inputs to better distinguish such subtle differences.

Meanwhile, the Falling class reached an accuracy of 78.64%, demonstrating that the model can reliably detect potential accidents or abnormal postures in real-time monitoring scenarios a critical function for workplace safety systems.

Table 1 Accuracy Evaluation by Activity Class

Activity Class	Training: Testing Data Ratio	Acurracy
Working	80:20	79,42%
Sleeping	70:30	76.35%
Playing Phone	60:40	72.18%
Falling	75:25	78.64%

Overall, these results confirm that the YOLOv8-based system provides reliable multi-activity detection performance, even under varying data ratios. While current accuracy levels are moderate, they are sufficient for real-time safety and productivity monitoring. Further enhancement such as increasing dataset diversity, improving annotation consistency, and fine-tuning hyperparameters is expected to raise detection accuracy and system robustness in future implementations.

4. Conclusion and Suggestions

In this study, we successfully developed a computer vision-based worker monitoring system that can automatically detect and classify worker activities such as working, sleeping, using mobile phones, falling, and leaving the workspace. The system, built on the YOLOv8 object detection model and integrated with monitoring cameras, operates locally and presents real-time detection results through a Flask-based web interface. Important incidents, including falls and idle behaviors, are recorded and stored as images and videos for further review, ensuring a comprehensive record of workplace activities. The modular design adopted in this project offers strong potential for future development, including adding new monitoring zones, recognizing additional activity types, and improving detection accuracy through more advanced algorithms.

To enhance system reliability, it is also recommended to implement boundary testing and robust error handling. Future research directions could explore the integration of predictive machine learning models to anticipate unsafe behaviors, develop personalized alert systems to improve user experience, and strengthen data security measures to protect sensitive monitoring data. Overall, this project underscores the importance of a flexible, data-driven system that can adapt to evolving industrial needs for effective and secure worker supervision.

References

Ajerla, D., Mahfuz, S., & Zulkernine, F. (2019). A Real-Time Patient Monitoring Framework for Fall Detection. *Wireless Communications and Mobile Computing, 2019*, 9507938.

Bo, L. (2023). Human Fall Detection for Smart Home Caring Using YOLO Networks. *International Journal of Advanced Computer Science and Applications, 14*(4).

Chen, B., Wang, X., Bao, Q., Jia, B., Li, X., & Wang, Y. (2022). An Unsafe Behavior Detection Method Based on Improved YOLO Framework. *Electronics, 11*(12), 1912.

Fan, Y., Levine, M., Wen, G., & Qiu, S. (2017). A Deep Neural Network for Real-Time Detection of Falling Humans in Naturally Occurring Scenes. *Neurocomputing, 260*, 43-58.

Gowsikhaa, D., Abirami, S., & Baskaran, R. (2014). Automated Human Behavior Analysis from Surveillance Videos: A Survey. *Artificial Intelligence Review, 42*, 747-765.

Khekan, A., Aghdasi, H., & Salehpour, P. (2024). The Impact of YOLO Algorithms within Fall Detection Application: A Review. *IEEE Access*.

Li, S., & Song, X. (2023). Future Frame Prediction Network for Human Fall Detection in Surveillance Videos. *IEEE Sensors Journal, 23*(13), 14460-14470.

Mok, R., Chang, R., & Li, W. (2016). Detecting Low-Quality Workers in QoE Crowdtesting: A Worker Behavior-Based Approach. *IEEE Transactions on Multimedia, 19*(3), 530-543.

Mokayed, H., Quan, T., Alkhaled, L., & Sivakumar, V. (2023). Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques. *Artificial Intelligence and Applications, 1*(4), 205-213.

Muheidat, F., Tawalbeh, L., & Tyrer, H. (2018). Context-Aware, Accurate, and Real-Time

Fall Detection System for Elderly People. *IEEE 12th International Conference on Semantic Computing (ICSC)* (pp. 329-333). IEEE.

Parasuraman, R., Mouloua, M., Molloy, R., & Hilburn, B. (2018). Monitoring of Automated Systems. In *Automation and Human Performance* (pp. 91-115). CRC Press.

Pereira, G. (2024). Fall Detection for Industrial Setups Using YOLOv8 Variants.

Raza, A., Yousaf, M., & Velastin, S. (2022). Human Fall Detection Using YOLO: A Real-Time and AI-on-the-Edge Perspective. *12th International Conference on Pattern Recognition Systems (ICPRS)* (pp. 1-6). IEEE.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.

Sanjalawe, Y., Fraihat, S., Abualhaj, M., Al-E'Mari, S., & Alzubi, E. (2025). Hybrid Deep Learning for Human Fall Detection: A Synergistic Approach Using YOLOv8 and Time-Space Transformers. *IEEE Access, 13*, 41336-41366.

Singh, A., Singh, D., & Goyal, M. (2021). People Counting System Using Python. *5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1750-1754). IEEE.

Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., . . . Grewe, B. (2020). Deep Learning-Based Behavioral Analysis Reaches Human Accuracy and is Capable of Outperforming Commercial Solutions. *Neuropsychopharmacology, 45*(11), 1942-1952.

Vasavi, G., Diddi, H., Haritha, E., Kumar, G., & Varma, G. (2023). People Counting Based on YOLO. *4th IEEE Global Conference for Advancement in Technology (GCAT)* (pp. 1-6). IEEE.

Vishwakarma, V., Mandal, C., & Sural, S. (2007). Automatic Detection of Human Fall in Video. *Second International Conference on Pattern Recognition and Machine Intelligence (PReMI 2007)* (pp. 616-623). Springer.

Wijaya, J., Krisnanik, E., & Isnainiyah, I. (2022). Sistem Informasi Pemantauan Kinerja Pegawai Berbasis Web pada PT XYZ Indonesia. *Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, (pp. 245-256).

Yuan, Y., Miao, Z., & Hu, S. (2006). Real-Time Human Behavior Recognition in Intelligent Environment. *8th International Conference on Signal Processing.* IEEE.

Zhao, D., Song, T., Gao, J., Li, D., & Niu, Y. (2024). YOLO-Fall: A Novel Convolutional Neural Network Model for Fall Detection in Open Spaces. *IEEE Access, 12*, 26137-26149.

Zivkovic, T., Zivkovic, M., Jovanovic, L., Kaljevic, J., Dobrojevic, M., & Bacanin, N. (2025). YOLOv8 Model Architecture Selection for Human Fall Detection. *International Conference on Data Analytics and Management* (pp. 219-227). Springer.