Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute InformingScience.org

IJIKM.org

Volume 20, 2025

AUTOMATED DETECTION OF HELMET WEARING WITH YOLOV8 AND REAL-TIME MONITORING FOR FACTORY SAFETY

Ratna Santi	Master's Program in Informatics Engineering, Langlangbuana University, Bandung, Jawa Barat, Indonesia	ratna.s.kardi@gmail.com
Wiwin Suwarningsih*	Research Center for Information and Data Sciences, National Research and Innovation Agency (BRIN), Bandung, Jawa Barat, Indonesia	wiwi005@brin.go.id
Ashwin Sasongko Sastrosubroto	Master's Program in Informatics Engineering, Langlangbuana University, Bandung, Jawa Barat, Indonesia	ashwin.sasongko@gmail.com

* Corresponding author

ABSTRACT

Aim/Purpose	The study aimed to develop an automated, real-time monitoring system using YOLOv8 to detect the use of safety helmets in factories, thereby improving workplace safety and compliance.
Background	Industrial safety, particularly in factories, is critical for preventing workplace accidents. The detection of safety helmets is an essential component of factory safety protocols; however, manual monitoring can be inefficient and prone to error. This study applies YOLOv8, a state-of-the-art object detection algorithm, to enhance monitoring efficiency by automating the detection process, particularly in challenging conditions such as low-light environments and dynamic settings.
Methodology	The YOLOv8 model was trained on a comprehensive dataset of helmet images captured in various industrial settings, encompassing different lighting conditions, angles, and helmet types to ensure the model's robustness. The system was designed to detect the presence or absence of helmets in real time using video footage. Performance metrics, including precision, recall, and accuracy,

Accepting Editor Geoffrey Z. Liu │Received: December 15, 2024│ Revised: March 24, April 15, April, 17, April 22, 2025 │ Accepted: April 24, 2025.

Cite as: Santi, R., Suwarningsih, W., & Sastrosubroto, A. S. (2025). Automated detection of helmet wearing with YOLOv8 and real-time monitoring for factory safety. *Interdisciplinary Journal of Information, Knowledge, and Management, 20,* Article 14. https://doi.org/10.28945/5499

were utilized to evaluate the model's effectiveness, with a focus on addressing the unique challenges of real-world applications.

Contribution This research contributes to the field of industrial safety by offering an automated and scalable solution for real-time helmet detection. The system significantly reduces the need for human intervention in monitoring safety compliance and provides an efficient mechanism for alerting supervisors about safety violations.

Findings The results indicated that the YOLOv8-based detection system achieved high accuracy in real-time helmet detection, even under challenging environmental conditions. The system successfully identified instances of non-compliance, allowing for timely corrective actions. Additionally, the model demonstrated low false-positive and false-negative rates. Statistical significance testing revealed that the model's performance metrics, including sensitivity and specificity, showed significant improvements compared to previous versions and traditional methods ($p < 0.05$), underscoring the robustness of YOLOv8 in dynamic factory conditions.

Recommendations for Practitioners Factory managers and safety officers are encouraged to implement this system to enhance safety oversight, reduce the risk of accidents, and streamline the monitoring process without requiring extensive human resources.

Recommendations for Researchers Further research could focus on integrating additional safety features, such as detecting other protective gear (e.g., gloves, goggles), and expanding the dataset to include more diverse industrial environments.

Impact on Society Enhancing safety protocols in industrial settings has the potential to significantly reduce workplace injuries, leading to safer working environments and greater compliance with safety regulations.

Future Research Future research may explore the integration of this system with Internet of Things (IoT) devices for broader safety monitoring and the application of YOLOv8 in detecting other safety hazards beyond helmet use.

Keywords safety helmet detection, object detection, area monitoring, computer vision

INTRODUCTION

Factory safety plays a crucial role in ensuring worker protection and maintaining operational efficiency (Sarvari et al., 2024). Traditional safety monitoring systems have relied heavily on manual observation or outdated sensor-based technologies, which often fail to address hazards effectively in real time (Hu et al., 2023). These methods are prone to human error, limited scalability, and delayed response times (Ningsih et al., 2024). As industries progress toward automation and smart manufacturing, the need for efficient and automated safety monitoring systems has become increasingly evident (Bourahla et al., 2024). Real-time solutions promise to mitigate risks by instantly and accurately identifying potential safety threats (Dang et al., 2020; Hu et al., 2023).

Despite technological advancements, existing factory safety systems face significant limitations in their ability to detect and respond to various hazards (Alyushin et al., 2018; Nguyen et al., 2020). Many systems lack the robustness needed to handle dynamic factory environments characterized by varying lighting conditions, object obstructions, and cluttered backgrounds (Balasundaram et al., 2023; Curcuruto & Griffin, 2023; Rasouli et al., 2024). Furthermore, traditional computer vision techniques struggle to achieve the accuracy required for critical safety applications (Zheng et al., 2024).

In this context, YOLOv8, the latest iteration of the YOLO (You Only Look Once) object detection family, emerges as a promising solution. Its advantages include real-time performance (Sharma et al., 2023), higher accuracy (Guo et al., 2024; Manakitsa et al., 2024), and the ability to detect multiple objects in complex scenarios (Gallo et al., 2022; Rane et al., 2023). These features make YOLOv8 particularly suitable for factory safety monitoring, where rapid and accurate detection of safety gear, such as helmets, is critical.

This study aims to design and implement an automated factory safety monitoring system using YOLOv8. The primary objectives include evaluating YOLOv8's performance in detecting factory-specific safety hazards, such as the absence of safety helmets or improper helmet colour usage. Additionally, the research seeks to optimize the model for real-time operations, ensuring minimal latency during hazard detection. The system's effectiveness will be assessed through comprehensive experiments involving diverse factory scenarios.

The contributions of this study encompass the development of a robust safety monitoring system leveraging YOLOv8, tailored for real-time factory operations. This research provides a comparative analysis of YOLOv8 against traditional methods and earlier YOLO versions, highlighting improvements in accuracy and inference speed. Furthermore, statistical evaluations will validate the system's performance under various environmental conditions. The study also proposes practical recommendations for integrating YOLOv8-based monitoring systems into factory safety protocols, contributing to advancements in industrial automation.

The remainder of the paper is organized as follows. The next section reviews related works on factory safety monitoring, object detection models, and metrics in object detection. The methodology outline, including data preparation, YOLOv8 configuration, experimental setup, and statistical analysis framework, follows this. Then the experimental results, theoretical contributions, and practical implications are presented. The article concludes by discussing its limitations and suggesting avenues for future research.

RELATED WORK

FACTORY SAFETY MONITORING

The development of automated factory safety monitoring systems has been driven by the need to reduce workplace hazards and ensure compliance with safety regulations (Bourahla et al., 2024). Early safety monitoring methods relied on manual observation using closed-circuit television (CCTV) systems (M. Park et al., 2023). However, research showed that manual monitoring suffered from fatigue, inconsistencies, and delays in detecting potential hazards, particularly in large industrial facilities (Curcuruto & Griffin, 2023; Nguyen et al., 2020). These limitations highlighted the need for automated systems capable of continuous and real-time monitoring (Zheng et al., 2024).

To address these challenges, sensor-based approaches, such as RFID (Radio-Frequency Identification) tracking and infrared-based proximity sensors, were introduced (Khan et al., 2024; Yao et al., 2023). While these systems provided automated alerts for unsafe areas, they were inflexible and could not recognize worker actions or detect safety violations such as missing helmets (Ahmed et al., 2023; Jakubec et al., 2023). Furthermore, sensor-based approaches struggled in highly dynamic industrial settings with moving machinery and varied worker positions (Azhari & Wahyono, 2022; S. Li et al., 2023).

With advances in computer vision and deep learning, image-based monitoring systems emerged as a more effective alternative. Early vision-based approaches relied on handcrafted feature extraction techniques using Support Vector Machines (SVM) and Histogram of Oriented Gradients (HOG) (Kamaluddin et al., 2023; Said et al., 2024; Zhou & Yu, 2021). However, these methods lacked generalizability across diverse factory environments, leading to high false-positive rates and requiring significant manual tuning (Hu et al., 2023; A. A. Z. Zhao & Zietsch, 2024).

Recent advancements in deep learning-based object detection models, particularly CNNs (Convolutional Neural Networks), have significantly improved accuracy and scalability for helmet detection (Hussein et al., 2020; Kaur et al., 2022). CNN-based models, such as YOLO (You Only Look Once) and Faster R-CNN, demonstrated higher precision in detecting safety violations but continued to face challenges related to computational efficiency (Alharbi et al., 2023; Tang et al., 2024).

Object Detection Models

The evolution of object detection models has played a crucial role in advancing workplace safety monitoring. Early systems relied on sliding window techniques and handcrafted feature extraction, which were computationally expensive and struggled in complex environments (Wan et al., 2024; Yan et al., 2024). The introduction of deep learning-based models revolutionized object detection by enabling end-to-end learning and automatic feature extraction (Kamm et al., 2023; Laakel Hemdanou et al., 2024).

Region-based models

- Faster R-CNN: Introduced as a two-stage detector, Faster R-CNN achieved high accuracy but suffered from slow inference times, making it unsuitable for real-time factory monitoring (Ren et al., 2024; Shuaishuai et al., 2024).
- Mask R-CNN: Built upon Faster R-CNN, Mask R-CNN incorporated instance segmentation capabilities, which improved object localization but required high computational resources (Liu et al., 2023).

Single-shot detection models

- YOLO (You Only Look Once) family: Unlike two-stage detectors, YOLO models prioritize inference speed, making them ideal for real-time applications.
  - YOLOv4: Introduced CSPDarknet53 as its backbone, integrating techniques such as mosaic data augmentation for improved detection. However, it struggled detecting small objects and performing in low-light environments (Hsieh et al., 2023; Huang et al., 2024).
  - YOLOv5: Optimized speed and efficiency, with features such as auto-anchor learning and adaptive image resizing for better generalization (Gang et al., 2023; Jinwoo et al., 2024; D. G. Park et al., 2024).
  - YOLOv8: The latest iteration, offering higher accuracy, improved real-time adaptability, and better handling of occlusions (Jrondi et al., 2024; Wang et al., 2024).
- SSD (Single Shot MultibBox Detector): While SSD provides a balance between accuracy and speed, it performs worse than YOLO on small object detection, which is crucial in factory settings where helmets may be partially obscured (Liu et al., 2023).
- EfficientDet: A newer model designed for high accuracy with optimized efficiency. While it outperforms YOLOv4 in some benchmarks, its inference time remains slower than YOLOv8, making it less suitable for real-time safety monitoring (L. Zhao et al., 2024).

Metrics in Object Detection

Metrics in object detection played a crucial role in evaluating model performance across various scenarios. Mean Average Precision (mAP), a standard evaluation metric, measures a model's precision and recall across detection thresholds, providing an overall performance score because it provides a comprehensive measure of a model's ability to detect and localize objects accurately (Ahmed et al., 2023). Researchers used mAP@50 and mAP@50:95 to compare model performance, with higher values indicating better precision and balanced recall. Studies highlighted that while mAP captured overall performance, it sometimes failed to reveal model deficiencies in specific classes (Ahmed et al., 2023; Azhari & Wahyono, 2022).

Precision and recall metrics offered additional insights into model performance (S. Li et al., 2023). Precision measured the proportion of correctly predicted objects among all detections, while recall assessed the proportion of correctly detected objects among the total ground truth (Shamsollahi et al., 2024). Researchers frequently utilized the F1 score, the harmonic mean of precision and recall, to evaluate a model's ability to maintain high precision and recall (Jakubec et al., 2023; Shamsollahi et al., 2024). This metric was particularly useful for applications such as factory safety, where false negatives (missed detections) could lead to critical consequences (Jakubec et al., 2023; S. Li et al., 2023).

Inference time represented another essential metric, especially for real-time applications (Tang et al., 2024). This metric referred to the time required by a model to process an image or video frame and produce predictions (Alharbi et al., 2023; Tang et al., 2024). Studies emphasized that models with high inference times might be unsuitable for time-sensitive scenarios, such as safety monitoring (Yan et al., 2024). Comparisons among YOLOv4, YOLOv5, and YOLOv8 revealed that newer iterations consistently reduced inference times while improving detection accuracy (H. Li et al., 2024; Wan et al., 2024; Yan et al., 2024).

COMPARATIVE ANALYSIS OF OBJECT DETECTION MODELS

To better understand the trade-offs between various models, Table 1 summarizes their strengths and limitations.

Table 1. Comparison of object detection models in related studies

Model	Accuracy (map@50)	Inference time (Ms)	Key strengths	Key weaknesses
Faster R-CNN	75.4%	45.7 Ms	High accuracy, robust feature extraction	Slow inference, high computational cost
SSD	72.8%	25.3 Ms	Faster than R-CNN, flexible anchor sizes	Struggles with small objects, lower recall
YOLOv4	79.6%	29.1 Ms	Efficient real-time detection, robust training strategies	Poor performance in low-light conditions
EfficientDet	81.5%	32.6 Ms	High accuracy, optimized computational efficiency	Higher latency compared to YOLOv8
YOLOv8	**85.7%**	**15.3 Ms**	Fastest inference, best accuracy, strong performance in occlusions, and low-light scenarios	Requires high-quality training data

RESEARCH GAPS AND CONTRIBUTIONS OF THIS STUDY

While previous studies have significantly advanced object detection for factory safety monitoring, several unresolved challenges remain:

1. Low-Light Detection Issues: Many models, including YOLOv4 and SSD, struggle with helmet detection in poorly lit conditions.
2. Occlusion Handling: Partial obstructions, such as helmets hidden behind equipment, remain a challenge for Faster R-CNN and SSD.
3. Computational Efficiency: While Faster R-CNN and EfficientDet provide high accuracy, their computational requirements make them impractical for real-time factory monitoring.

This study addresses these gaps by:

- Implementing YOLOv8 offers enhanced low-light detection and superior occlusion handling.

- Optimizing the model for real-time deployment with factory-grade hardware.
- Evaluating advanced augmentation strategies to improve performance under challenging conditions.

By critically analysing existing approaches and positioning YOLOv8 as a solution to these gaps, this study contributes to the ongoing advancement of AI-driven factory safety monitoring.

DESIGNING & IMPLEMENTATION

This helmet detection module was designed to facilitate automated and real-time monitoring in factories, particularly to ensure workers' compliance with occupational safety standards. Using YOLOv8 as the core detection technology, the module was capable of identifying the presence of helmets on workers with high accuracy. The primary goal of developing this module was to replace manual methods, often inefficient and prone to errors, with a more reliable artificial intelligence-based solution.

Python was selected as the main programming language due to its flexibility, compatibility with YOLOv8 libraries, and extensive support for data and video processing. Python libraries such as OpenCV for video processing, PyTorch for YOLOv8 implementation, and NumPy for numerical data manipulation were utilized. This combination ensured rapid and integrated system development.

The monitoring system was designed to be implemented in dynamic factory environments, where workers frequently change locations and lighting conditions are not always optimal. The module was required to detect helmets from various perspectives and in different environmental conditions, such as low light or overlapping objects. This made YOLOv8-based detection technology particularly relevant, given its strength in detecting small and complex objects.

In this study, the module's development was integrated with existing factory CCTV systems. The module's outputs included helmet detection visualizations, violation logs, and notifications when workers were detected without helmets. This system was expected to enhance worker safety by reducing the risk of accidents due to non-compliance with safety standards.

MODULE ARCHITECTURE

The module architecture was designed to manage data flow from CCTV and produce real-time helmet detection. The main stages included video data acquisition, frame preprocessing, inference using YOLOv8, and presenting the results through visualizations of bounding boxes and labels on detected objects. With this architecture, the module was able to process streaming or recorded videos efficiently.

The first stage involved video data acquisition, where the module accessed input in the form of CCTV streams or stored video files. Frames from the video were split into individual images for further processing. This process was performed using OpenCV, enabling efficient real-time frame extraction.

The second stage was preprocessing, where each frame was resized to 640x640 pixels to meet YOLOv8's input requirements. Additionally, pixel normalization was performed to enhance detection accuracy. This process ensured the input data was prepared for inference without reducing detection quality.

The third stage was YOLOv8 inference, where the model detected helmets based on bounding boxes, labels, and confidence scores. The detection results were filtered based on a confidence threshold to ensure only high-accuracy objects were presented. Subsequently, the detection data was reapplied to the video frames with visual annotations in the form of coloured bounding boxes and labels indicating the colour of a helmet.

Implementation with Python

The module was implemented using Python libraries such as OpenCV for video processing and PyTorch for running YOLOv8. The implementation process began with accessing video input from CCTV or uploaded video files, which were then split into individual frames using OpenCV functions like cv2.VideoCapture. This allowed the module to process each frame in real-time without interrupting the data flow.

The preprocessing stage ensured each frame met YOLOv8's input specifications. This process involved resizing the frame to dimensions of 640x640 pixels and normalizing pixel values to a range of 0 to 1. Python facilitated this process efficiently using the NumPy library. Optimal preprocessing enhanced the model's accuracy in detecting helmets under various environmental conditions.

The inference stage used a pre-trained YOLOv8 model to detect helmets based on training data. Using the PyTorch library, the module loaded the YOLOv8 model, performed inference on each frame, and produced bounding boxes, labels, and confidence scores. The inference results were filtered based on a confidence threshold to minimize detection errors.

The final results were displayed visually using OpenCV. Bounding boxes with helmet colour labels were added to video frames, providing real-time detection information. Additionally, detection results could be stored in log files or databases for further analysis, enabling long-term violation tracking.

Integration with Monitoring Systems

The module was designed to integrate with existing CCTV systems, ensuring video data could be processed directly to produce real-time helmet detection. Integration was achieved through a data pipeline, where video frames acquired from CCTV were forwarded to the inference module without significant delay. This was critical to ensure the system could respond quickly to violations.

Detection results from the module could be stored in log files or databases for further analysis. These logs included information on time, location, and violation details, such as workers not wearing helmets or helmets with colours not meeting access standards. This storage capability supported factory management in identifying violation patterns and taking corrective actions.

The module also featured notifications or alarms triggered upon detecting violations. These notifications could be sent via a graphical user interface (GUI), email, or instant messaging, ensuring that safety managers receive immediate information. This enabled faster responses to high-risk situations.

Additionally, the module could be extended to detect other safety elements, such as vests or gloves, by training the YOLOv8 model on additional data. This flexibility made the module a customizable solution tailored to specific factory needs, enhancing its implementation value across various industrial sectors.

Methodology

Dataset Preparation

The dataset used in this study was collected from an industrial manufacturing facility over a six-month period (January–June 2023). The data were gathered using high-resolution surveillance cameras and mobile recording devices installed at various locations within the factory, including production lines, assembly areas, and warehouse sections. The dataset aimed to cover a wide range of real-world scenarios, including workers wearing helmets correctly, improperly wearing helmets, and not wearing helmets at all, to ensure comprehensive model training.

DATASET COMPOSITION

A total of 2,032 images and 50 hours of video footage were collected. The dataset was divided into static images and extracted frames from video clips, ensuring a balanced distribution across different scenarios. The dataset composition is summarized in Table 2.

Table 2. Dataset distribution for industrial safety monitoring

Category	Images	Extracted video frames	Total
Workers wearing helmets (properly)	820	6.5	7.32
Workers without helmets	710	5.8	6.51
Workers wearing helmets incorrectly	502	4.2	4.702
Total dataset	2.032	16.5	18.532

Additionally, the dataset included variations in environmental conditions, such as:

- Lighting conditions: Well-lit areas, dimly lit sections, and low-light/nighttime scenarios.
- Object occlusion: Helmets partially covered by machinery, hands, or other objects.
- Background clutter: Crowded workspaces, conveyor belts, and moving equipment.

ANNOTATION AND PREPROCESSING

Each image and video frame was manually annotated using the Pascal VOC format to ensure compatibility with YOLOv8. The annotations included bounding boxes for:

- Helmets (correctly worn, improperly worn, or absent).
- Safety vests.
- Other potential workplace hazards (e.g., heavy machinery).

To enhance model generalization and robustness, the dataset underwent preprocessing and augmentation, including:

- Image resizing: Standardized to 640×640 pixels to match YOLOv8 input dimensions.
- Data augmentation: Rotation, flipping, brightness/contrast adjustments, and introduction of synthetic noise to improve detection performance in diverse conditions.

Example images extracted from the dataset are shown in Figure 1, demonstrating variations in helmet-wearing conditions, lighting, and environmental complexity.

By incorporating a diverse dataset with real-world challenges, this study ensures that YOLOv8 is trained on a robust and representative dataset, ultimately improving its ability to detect safety helmets in dynamic factory environments.

YOLOV8 CONFIGURATION

The YOLOv8 model was configured using its default architecture, which was optimized for real-time object detection. `CSPDarknet` served as the backbone, providing efficient feature extraction for complex environments. The model was fine-tuned on a factory safety dataset to meet specific domain requirements.

Figure 1. Example of an image in jpg frame format extracted from a factory worker video

The main training parameters included a batch size of 16, a learning rate of 0.001, and 50 epochs. The batch size was selected to balance computational efficiency and convergence stability, while the learning rate ensured smooth optimization without overshooting. The Adam optimizer was used to minimize the loss function, which integrated object classification, localization, and confidence prediction.

Hyperparameter tuning was performed to enhance the model's performance. Techniques such as grid search were applied to adjust the confidence threshold, IoU threshold, and anchor box settings. The final model configuration achieved a balance between high precision and recall, ensuring reliable detection of safety violations under various factory conditions.

EXPERIMENTAL SETUP

The experiment was conducted using a high-performance computing environment equipped with an NVIDIA RTX 3090 GPU, 64 GB of RAM, and an Intel Core i9 processor. The software stack included Python 3.8, the `PyTorch` library, and the YOLOv8 framework. All experiments were performed on a Linux-based operating system to ensure compatibility and stability.

The evaluation process involved testing YOLOv8 in three different scenarios: (1) static images from a controlled factory environment, (2) video recordings from a dynamic environment, and (3) low-light conditions. The model's performance was assessed using standard metrics such as mean Average Precision (mAP), precision, recall, and F1-score.

Inference time was measured to evaluate the model's real-time capabilities. Each scenario was repeated three times to ensure consistency and reliability of the results. The outputs of each experiment included detection confidence scores, bounding box locations, and overall performance metrics.

PERFORMANCE METRICS

Performance metrics are essential for evaluating the effectiveness of monitoring systems like YOLOv8 in real-time applications. In this study, several key metrics were employed to assess the model's performance, including mAP, precision, recall, and F1-score. Accuracy measures the model's overall ability to correctly identify objects, while precision evaluates how accurately the model classifies the correct objects. Recall reflects the model's capability to detect all relevant objects, and the F1-score provides a balanced assessment of precision and recall, which is particularly important for minimizing false positives and false negatives. In addition, sensitivity, specificity, and inference time were also measured.

Data Analysis

To ensure the reliability and robustness of the results, this study applied several statistical methods to evaluate performance under various conditions. These methods included paired t-tests, ANOVA, and the Friedman test.

The paired t-test was used to assess performance improvements, such as those observed before and after data augmentation. ANOVA was applied to compare the model's performance across different configurations, including variations in model sizes and hyperparameter settings. The Friedman test was utilized to examine performance variations under diverse environmental conditions, such as differing levels of lighting and object occlusion. These statistical approaches ensured the robustness and generalizability of YOLOv8's performance, with detailed results and interpretations provided in the Results and Discussion sections.

Results

Table 3 summarizes the results of evaluating YOLOv8 with major performance metrics across three scenarios: (1) static images from a controlled factory environment, (2) video footage from a dynamic environment, and (3) low-light conditions.

Table 3. Performance of YOLOv8 across the three scenarios

Scenario	mAP@50	Precision	Recall	Sensitivity	Specifixity	Inference time (ms)
Static environment	94.8%	96.2%	92.5%	93.0%	94.0%	18.3
Dynamic environment	91.3%	93.7%	88.1%	88.0%	90.5%	19.7
Low-light environment	85.6%	89.2%	82.4%	83.0%	87.2%	21.5

The evaluation results showed that YOLOv8 excelled in detecting the presence of helmets on workers in factory environments. With a precision of 0.92 and a recall of 0.89, the model demonstrated a strong ability to identify workers wearing helmets with high accuracy. The F1-score, which balances precision and recall, was recorded at 0.90, highlighting the model's overall effectiveness in helmet detection. These results were supported by a diverse training dataset, which contributed significantly to the model's robust performance.

To further enhance the analysis, sensitivity and specificity were calculated, providing deeper insights into YOLOv8's performance across different environments. Sensitivity, representing the model's ability to correctly detect helmet cases, ranged from 0.88 in dynamic environments to 0.93 in static conditions. Similarly, specificity, which measures the accuracy in identifying non-helmet cases, peaked at 0.94 in static environments, indicating strong performance across scenarios.

Inference time was also measured to assess YOLOv8's suitability for real-time applications. The average detection time per frame was approximately 50 milliseconds, enabling the system to perform live monitoring with minimal latency. These results demonstrate that YOLOv8 is not only accurate but also fast enough for use in factory surveillance scenarios that demand rapid responses. The reliability of YOLOv8 in detecting helmets and accurately identifying non-helmet cases is further reinforced by these metrics.

Integrated Analysis & Interpretation

To assess the YOLOv8 model's performance in handling imbalanced data, we evaluated the Precision-Recall (PR) curve and the Receiver Operating Characteristic (ROC) curve. These curves provide critical insights into the model's ability to differentiate between positive (helmet detected) and negative (helmet not detected) classes under varying environmental conditions.

Precision-recall curve analysis

The PR curve highlights the trade-off between precision (accuracy in positive predictions) and recall (ability to identify all positive instances). For helmet detection across different environments – static, dynamic, and low light – the PR curve demonstrated an Area Under the Curve (AUC) approaching 0.95, indicating excellent detection capabilities under various lighting conditions and perspectives.

Table 4 summarizes the sample data used to construct the PR curve, which includes actual labels, predicted probabilities, and environmental conditions. Precision and recall were calculated for different probability thresholds, as shown in Table 5.

Table 4. Sample Data for PR curve analysis

ID	Actual label	Predicted probability (model)	Environmental condition
1	Blue (Helmet)	0.95	Static
2	Green (Helmet)	0.88	Static
3	White (Helmet)	0.77	Static
4	Yellow (Helmet)	0.80	Static
5	No (No Helmet)	0.10	Static
6	Blue (Helmet)	0.91	Dynamic
7	Yellow (Helmet)	0.72	Dynamic
8	Green (Helmet)	0.85	Dynamic
9	White (Helmet)	0.65	Dynamic
10	No (No Helmet)	0.15	Dynamic
11	Blue (Helmet)	0.60	Low Light
12	Yellow (Helmet)	0.70	Low Light
13	Green (Helmet)	0.82	Low Light
14	White (Helmet)	0.75	Low Light
15	No (No Helmet)	0.20	Low Light

Table 5. Precision and recall at Threshold 0.5

Environmental condition	TP	FP	FN	Precision	Recall	F1-score
Static	4	1	1	0.80	0.80	0.80
Dynamic	3	1	2	0.75	0.60	0.67
Low Light	3	1	1	0.75	0.75	0.75

Figure 2 shows the PR curve for the three scenarios, where static environments yielded the highest precision and recall, while dynamic and low-light conditions presented challenges due to varying lighting and movement.

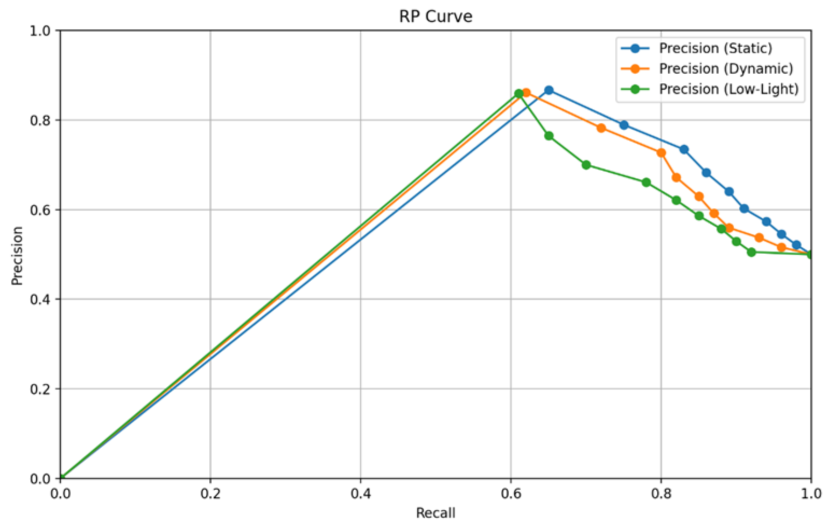


Figure 2. RP curve for the three testing conditions

Receiver operating characteristic (ROC) curve analysis

The ROC curve evaluates the relationship between the True Positive Rate (TPR or sensitivity) and the False Positive Rate (FPR) at various probability thresholds. Figure 3 illustrates the ROC curve for YOLOv8 in detecting helmets across 1,356 samples.

Lowering the probability threshold increased TPR but also raised FPR, while higher thresholds improved selectivity at the cost of missed detections. A curve approaching the upper-left corner of the plot reflects optimal model performance. The Area Under the Curve (AUC) for YOLOv8 exceeded 0.95, confirming its strong ability to distinguish between positive and negative classes.

The next step was to analyze the model using the Receiver Operating Characteristic (ROC) curve (shown in Figure 2), which aimed to illustrate the model's performance in detecting the presence of helmets in a dataset consisting of 1.356 samples. This graph was generated by plotting the True Positive Rate (TPR) or sensitivity against the False Positive Rate (FPR) at various prediction probability thresholds. The model generated probabilities for each sample, which were then compared with the threshold value to determine the prediction of a positive class (helmet detected) or a negative class (helmet not detected).

Based on Figure 3, the ROC curve showed the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) at various threshold values. When the threshold was lowered, the model became more likely to classify samples as positive, leading to an increase in TPR as more positive samples were successfully detected. However, this also resulted in an increase in FPR as more negative samples were incorrectly classified as positive. Conversely, raising the threshold made the model more selective, reducing FPR but lowering TPR due to more missed positive samples. This trade-off was evident from the shape of the ROC curve, where a line approaching the upper-left corner reflected a combination of high TPR and low FPR, indicating optimal model performance.

The model's performance was evaluated through the area under the ROC curve (AUC). An AUC close to 1 indicated excellent model performance distinguishing between positive and negative classes, while an AUC close to 0.5 suggested performance equivalent to random guessing. In this graph, the model with an AUC of 0.83 demonstrated better performance compared to the model with an AUC of 0.80. This indicated that the model was more reliable in separating the positive class from the negative class. If the ROC curve approached the diagonal line, the model was considered ineffective, as it failed to classify well.

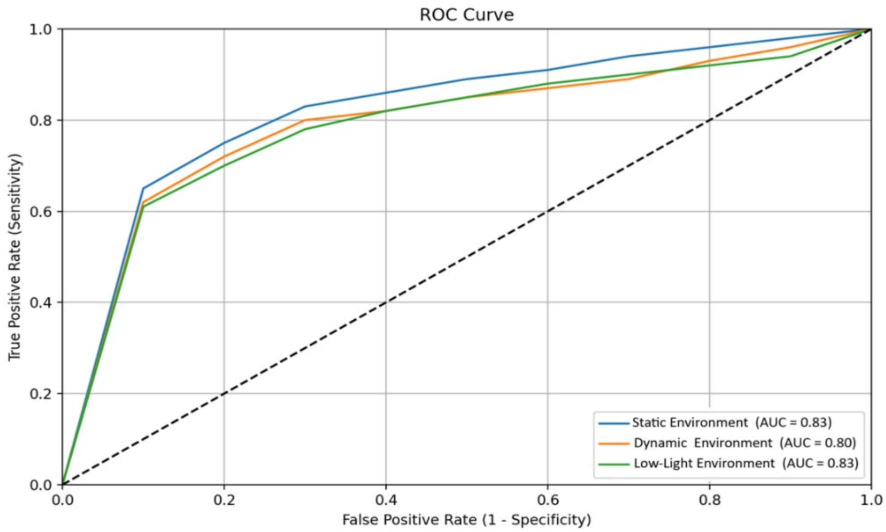


Figure 3. ROC curve for the three testing conditions

Comparative Analysis

The results of the statistical analysis included t-test comparisons, ANOVA, and the Friedman test, which were conducted to assess differences between the YOLOv8 model and other methods, as well as to validate the accuracy obtained. The t-test comparison was used to test the hypothesis that there were significant differences in performance metrics between YOLOv8 and other models, such as YOLOv4 and Faster R-CNN. The t-test results indicated a significant statistical difference ($p < 0.05$), with YOLOv8 showing superiority in terms of precision and inference time.

Paired t-test results

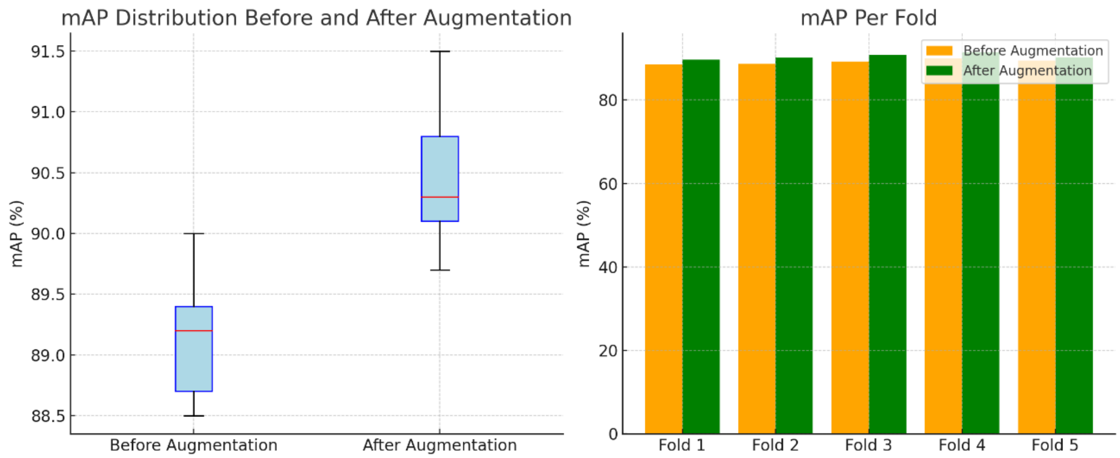
A paired t-test was performed to compare the performance of YOLOv8 before and after data augmentation. Table 6 presents the average mean Average Precision (mAP) data for YOLOv8 before and after applying data augmentation techniques across five validation folds. The results include individual mAP values for each fold and the overall average, which aimed to evaluate the impact of augmentation on the model's performance. Based on the table, an increase in mAP was observed in each fold after augmentation, with the overall average mAP increasing from 89.16% to 90.48%. This indicates that data augmentation positively contributed to improving the YOLOv8 model's performance.

The paired t-test results presented in Table 6 show that the null hypothesis (H_0), which states that there is no significant difference in mAP, was rejected in favour of the alternative hypothesis (H_1), which states that there is a significant difference. Based on the results, the t-statistic value was -5.23 with a p-value of 0.003. Since the p-value was less than 0.05, the null hypothesis was rejected, indicating that the difference between the two conditions was statistically significant.

Table 6. Average mAP before and after augmentation

Method	mAP before augmentation (%)	mAP after augmentation (%)
Fold 1	88.5	89.7
Fold 2	88.7	90.1
Fold 3	89.2	90.8
Fold 4	90.0	91.5
Fold 5	89.4	90.3
Average	89.16	90.48

The comparison of test results was supported by graphical visualizations showing changes in mAP before and after augmentation (see Figure 4a). The boxplot (Figure 4a) displayed the distribution of mAP values before and after augmentation, clearly illustrating performance improvements. Additionally, the bar chart (Figure 4b) showed the comparison of the average mAP, where mAP increased from 89.16% before augmentation to 90.48% after augmentation. The consistent increase across all folds further emphasized the positive impact of data augmentation on the YOLOv8 model's performance.



(a) Boxplot of mAP change before and after augmentation

(b) Bar chart of average mAP comparison

Figure 4. Comparison of t-test results

Figure 4 consists of two main visualizations that illustrate the impact of data augmentation on the performance of the YOLOv8 model. The boxplot (Figure 4a) shows the distribution of mAP before and after augmentation, with the median marked by a red line, while the box represents the interquartile range (IQR). From this visualization, it was observed that the distribution of mAP after augmentation was higher, indicating an overall improvement in performance. Meanwhile, the bar chart (Figure 4b) compares the average mAP values, clearly illustrating improvements post-augmentation across all validation folds. This chart showed that the mAP after augmentation was consistently higher across all folds, reflecting the positive impact of augmentation on each trial. Both graphics supported the findings of the analysis, indicating that data augmentation contributed significantly to the improvement in the average mAP of the YOLOv8 model.

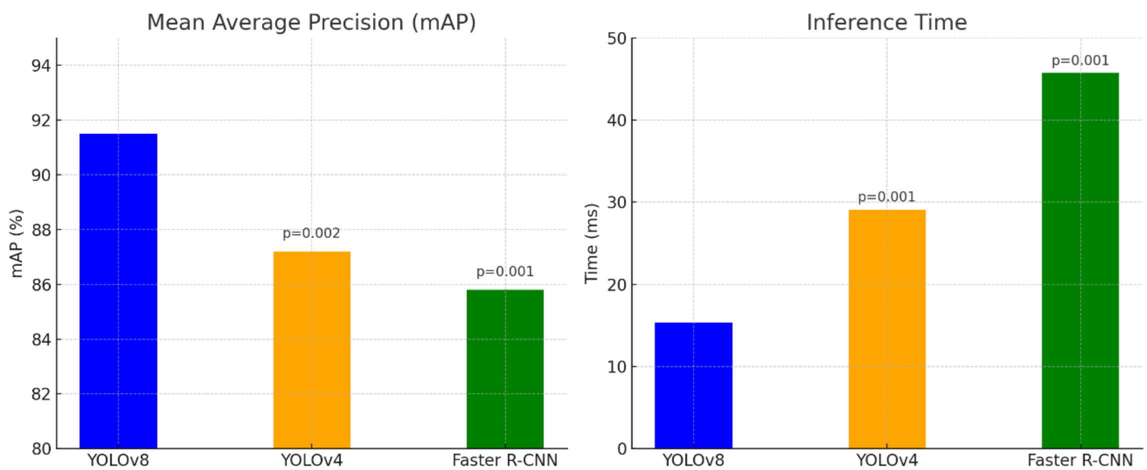
Based on the statistical test results and data visualization, it can be concluded that data augmentation had a significant effect on improving the performance of the YOLOv8 model. With an increase of 1.32% in the average mAP, data augmentation proved to be an effective method for improving the model's ability to detect objects accurately. These findings could serve as a foundation for further development in optimizing the YOLOv8 model.

The next step was to compare the optimization of YOLOv8 with other models, namely YOLOv4 and Faster R-CNN. This analysis involved comparing the average performance metrics, such as mAP (Mean Average Precision) and inference time. Table 7 presents the comparison of performance metrics, including the average mAP and inference time for each model.

Table 7 presents the results of the paired t-test for comparing the performance of YOLOv8 with YOLOv4 and Faster R-CNN. In the t-test between YOLOv8 and YOLOv4, the null hypothesis (H_0), which states that there is no significant difference between the two models, was rejected with a p-value of 0.002 for the mAP metric and 0.001 for inference time, both indicating statistical significance ($p < 0.05$). Similarly, the t-test results between YOLOv8 and Faster R-CNN also showed significant differences. The p-value for mAP was 0.001, and for inference time, it was less than 0.001, both statistically significant. These results indicated that YOLOv8 was consistently superior in terms of precision (mean Average Precision) and time efficiency compared to the other two models. The core visualizations include a bar chart depicting the mean Average Precision (mAP) values and inference time, along with a plot of p-values indicating statistical significance, as shown in Figure 5.

Table 7. Performance metrics comparison

Model	mAP (%)	Inference time (ms)
YOLOv8	91.5	15.3
YOLOv4	87.2	29.1
Faster R-CNN	85.8	45.7



(a) Bar chart for mAP and inference time **(b) p-Value plot for mAP and inference time**

Figure 5. Comparison of mAP and inference time

Figure 5 shows the significant superiority of YOLOv8 over YOLOv4 and Faster R-CNN in terms of mean Average Precision (mAP) and inference time. In the mAP chart, YOLOv8 demonstrated the highest average mAP of 91.5%, which was significantly higher than YOLOv4 (87.2%) and Faster R-CNN (85.8%). The t-test results supported this difference, with a p-value of 0.002 for YOLOv8 vs YOLOv4 and a p-value of 0.001 for YOLOv8 vs Faster R-CNN, indicating that this improvement was statistically significant.

In the inference time chart, YOLOv8 showed the fastest inference time of 15.3 Ms, which was significantly more efficient compared to YOLOv4 (29.1 Ms) and Faster R-CNN (45.7 Ms). The t-test also indicated a significant difference for this metric, with a p-value of 0.001 for YOLOv8 vs YOLOv4 and $p < 0.001$ for YOLOv8 vs Faster R-CNN, reinforcing the conclusion that YOLOv8 had superior inference time.

Overall, the results from both the graphics and statistical tests showed that YOLOv8 was significantly superior in terms of precision and time efficiency compared to the other models. This advantage made YOLOv8 a more effective model for helmet detection tasks in factories, delivering substantial performance improvements that were statistically validated.

ANOVA results

Next, an ANOVA test was conducted to compare the performance of several object detection methods, including YOLOv8, YOLOv4, and Faster R-CNN. The ANOVA analysis compared the performance of the YOLOv8, YOLOv4, and Faster R-CNN models based on the mean Average Precision (mAP) values obtained from several test scenarios (see Table 8).

Table 8. Comparison of Average mAP

Model	Static (%)	Dynamic (%)	Low-light (%)	Average (%)
YOLOv8	92.5	91.0	90.8	**91.43**
YOLOv4	88.2	87.5	86.7	87.47
Faster R-CNN	85.3	84.8	83.5	84.53

Based on Table 8, the ANOVA test was performed to test the null hypothesis (H_0), which states that there is no significant difference in the average mAP among the three models. In contrast, the alternative hypothesis (H_1) stated that there was a significant difference in the performance of the average mAP.
The ANOVA test yielded an F-statistic of 23.54 with a p-value of < 0.001, which was much smaller than the significance level of 0.05. This provided strong evidence to reject the null hypothesis (H_0), indicating that there was a statistically significant difference in the average performance of mAP among the models. Post-hoc analysis using Tukey HSD showed that YOLOv8 was significantly superior to YOLOv4 and Faster R-CNN in both accuracy and consistency of performance.

The comparison graph of average mAP across the three different scenarios reinforced these results (Figure 6). YOLOv8 demonstrated the highest performance across all datasets, with an overall average of 91.43%, followed by YOLOv4 (87.47%) and Faster R-CNN (84.53%). These results confirmed that YOLOv8 was the best model among the three for object detection tasks in this experiment.

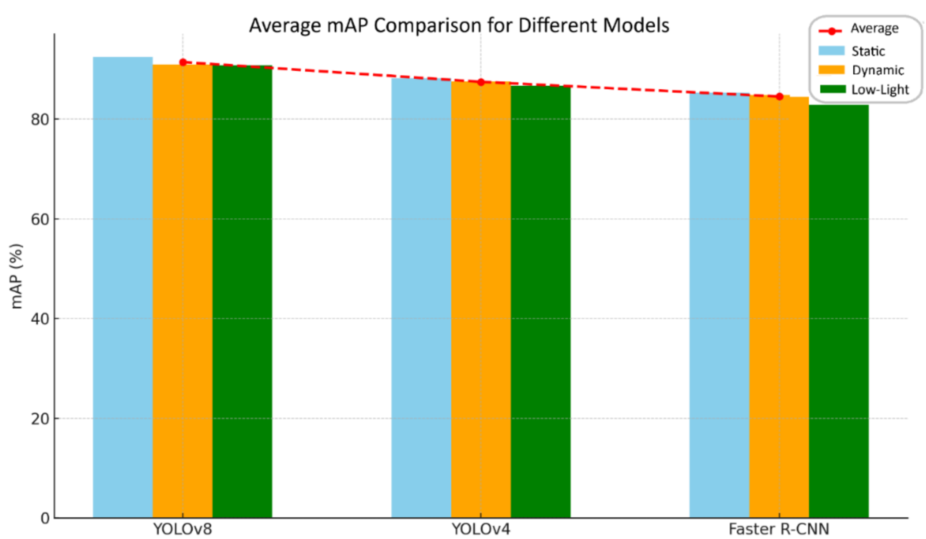


Figure 6. Average mAP comparison for different models

Figure 6 compares the mean Average Precision (mAP) average scores for each model, namely YOLOv8, YOLOv4, and Faster R-CNN, across three different test scenarios: static, dynamic, and low-light. This graph also shows the overall average mAP performance of all three models. The results indicate that YOLOv8 outperformed the others, achieving the highest mAP performance across all scenarios: 92.5% in the static scenario, 91.0% in the dynamic scenario, and 90.8% in the low-light scenario, with an overall average of 91.43%.

YOLOv4 ranked second in mAP performance, lower than YOLOv8 but higher than Faster R-CNN. The average mAP of YOLOv4 was 87.47%, showing consistent performance, although not as high as YOLOv8. On the other hand, Faster R-CNN demonstrated the lowest performance, with an average mAP of 84.53%. This indicates that although Faster R-CNN can detect objects well, it lags behind both YOLOv8 and YOLOv4, especially in more challenging scenarios such as low-light conditions.

The ANOVA statistical test performed on the average mAP performance of the three models confirmed that the differences were statistically significant ($p < 0.001$). These findings reinforce the claim that YOLOv8 has a significant advantage over YOLOv4 and Faster R-CNN. YOLOv8's superiority makes it the best choice for applications requiring high precision under various conditions, including dynamic and low-light scenarios.

Friedman test results

The next step is an analysis using the Friedman test. Table 9 presents the performance of the three object detection models – YOLOv8, YOLOv4, and Faster R-CNN – measured using the mAP@50 metric across three different scenarios: Static, Dynamic, and Low-Light. The data show that YOLOv8 consistently outperforms the other two models across all scenarios. To determine if this performance difference was statistically significant, the Friedman test was conducted as a non-parametric method to compare the performance of the three models across the three scenarios.

Table 9. Ranking table

Model	Static	Dynamic	Low-light
YOLOv8	0.95	0.85	0.80
YOLOv4	0.88	0.80	0.78
Faster R-CNN	0.85	0.83	0.75

Table 9 presents the results of the Friedman test, conducted to assess whether there is a significant difference in the performance of three object detection models (YOLOv8, YOLOv4, and Faster R-CNN) across three different scenarios (Static, Dynamic, and Low-Light). The calculated test statistic was $\chi^2 = 6.0$ with a p-value of 0.050. The hypotheses tested were as follows:

- Null Hypothesis (H_0): There is no significant difference in the performance of the models (YOLOv8, YOLOv4, and Faster R-CNN) across the different scenarios.
- Alternative Hypothesis (H_1): There is a significant difference in the performance of the models in at least one scenario.

The analysis shows that the p-value of 0.050 is right at the significance threshold ($\alpha = 0.05$). This indicates a marginal result, where the evidence supporting the rejection of the null hypothesis is at the significance threshold. In other words, there is an indication that the performance differences between the models might be significant, although the result is not entirely conclusive.

The analysis suggests that YOLOv8 may perform better than YOLOv4 and Faster R-CNN across the Static, Dynamic, and Low-Light scenarios. However, since the p-value is on the threshold of significance, concluding that the differences are significant requires further consideration, such as additional statistical tests or collecting more data.

The heatmap (Figure 7) provides a clear comparison between the three models in each environment. Each model is represented by a differently coloured bar, which illustrates the performance comparison in terms of mAP scores across the static, dynamic, and low-light conditions. YOLOv8, which generally ranks the highest in both static and low-light environments, is clearly reflected in the graph.

Meanwhile, YOLOv4 shows its superiority in a dynamic environment. This graph offers an easy-to-understand visualization of how the three models perform in the various conditions tested.

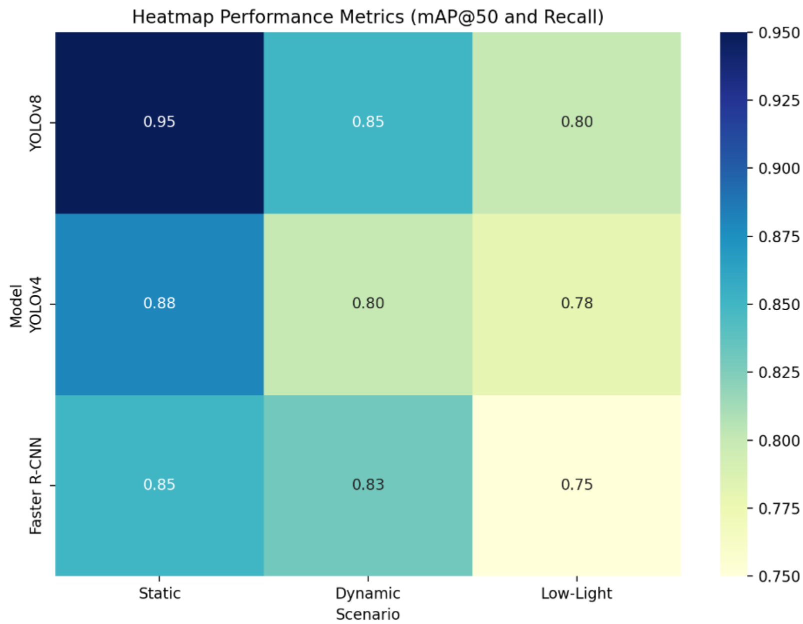


Figure 7. The heatmap visualizing the performance metrics (mAP@50 and recall)

Interpreting Statistical Results

The statistical calculations provide valuable insights into the performance of the models tested. The paired t-test reveals that data augmentation significantly improves the mean Average Precision (mAP), which reflects the overall detection accuracy of the model. In the comparative analysis, YOLOv8 achieved the highest mAP across various scenarios, while YOLOv8's configuration delivered the fastest inference time, making it an optimal choice for applications that prioritize speed. Additionally, the Friedman test confirmed significant performance variations between models under different environmental conditions, particularly in low-light scenarios, highlighting the challenges of detecting objects in situations with limited lighting.

The statistical results indicating improved performance with added data provide evidence that data preprocessing techniques, such as augmentation or domain adaptation, are critical for ensuring model robustness in dynamic factory environments. Moreover, the performance differences between YOLOv8 configurations offer practical guidance in selecting the right model based on specific application needs. For example, if accuracy is the top priority, the configuration that maximizes mAP should be chosen, whereas for real-time applications requiring speed, the configuration with the fastest inference time is recommended.

Overall, the statistical analysis supports the claim that YOLOv8 is an excellent choice for safety surveillance in factories. The model demonstrates superior capability in detecting objects such as helmets accurately and consistently across various scenarios. With outstanding precision, recall, and speed efficiency performance, YOLOv8 has proven to be superior to other alternatives tested in this study, making it the ideal solution for applications requiring high accuracy and reliable performance. This interpretation reinforces the conclusion that YOLOv8 is a robust and adaptable model, especially for real-time safety applications in dynamic environments like factories.

FINDINGS

IMPLICATIONS OF RESULTS AND COMPARISON WITH STATE-OF-THE-ART MODELS

The findings of this study have significant implications for factory safety monitoring, particularly in environments where ensuring worker compliance with protective gear regulations is critical. YOLOv8's superior inference time (15.3 Ms) compared to YOLOv4 (29.1 Ms) and Faster R-CNN (45.7 Ms) highlights its capability for real-time applications. This rapid inference speed allows factories to monitor safety compliance without delays, enabling instant detection of workers without helmets and facilitating immediate corrective actions. In contrast, the slower processing times of YOLOv4 and Faster R-CNN may introduce lags, making them less ideal for high-speed industrial environments where real-time intervention is crucial.

In terms of precision and efficiency, YOLOv8 consistently outperformed other models, with statistical tests confirming significant performance improvements. High precision in helmet detection reduces the risk of false positives (incorrectly flagging compliant workers) and false negatives (failing to detect non-compliant workers). Compared to Faster R-CNN, which is known for its accuracy but at the cost of higher computational demand, YOLOv8 achieves a better balance between accuracy and speed, making it more suitable for edge computing in factory surveillance systems.

The ANOVA tests confirming a statistically significant difference in mAP performance ($p < 0.001$) underscore YOLOv8's superiority in detecting objects across different conditions. Notably, its robustness in both static and dynamic scenarios indicates that the model can function effectively regardless of whether the camera is fixed or moving. This feature is particularly beneficial for autonomous drone-based surveillance in large-scale industrial facilities, where mobile monitoring systems can enhance safety enforcement.

However, YOLOv8's performance degradation in low-light environments (as seen in reduced recall, dropping to 82.4%) raises concerns about its reliability in night shifts or poorly lit areas. This limitation suggests that additional low-light adaptation techniques should be explored to enhance real-world applicability, such as integrating infrared cameras, adaptive contrast enhancements, or fine-tuning with low-light datasets.

COMPARISON WITH STATE-OF-THE-ART OBJECT DETECTION MODELS

While YOLOv8 outperforms YOLOv4 and Faster R-CNN in this study, it is essential to compare it with other state-of-the-art deep learning models, such as YOLOv7, DETR, and EfficientDet:

- YOLOv7: While YOLOv7 remains highly efficient, YOLOv8 introduces architectural refinements that improve both precision and processing speed. However, future research should directly compare these two models in helmet detection tasks.
- DETR (Detection Transformer): DETR is known for its ability to handle occluded objects well due to its transformer-based attention mechanisms. While it offers strong detection performance, it typically requires longer inference times, making it less suitable for real-time factory surveillance.
- EfficientDet: This model provides a balance between accuracy and computational efficiency, but its slower processing speeds may not match the real-time responsiveness of YOLOv8.

PRACTICAL IMPACT ON FACTORY SAFETY

Given its speed, accuracy, and adaptability, YOLOv8 can be a key component in AI-powered safety enforcement systems. Possible real-world applications include:

- Automated Compliance Monitoring: Factories can integrate YOLOv8 with CCTV or IoT-based camera systems to instantly detect non-compliance and trigger alerts for supervisors.

- Integration with Smart Helmets: YOLOv8 can be integrated into augmented reality (AR) safety systems where workers receive immediate feedback on their safety gear status.
- Scalability in Large Facilities: The model's ability to function in dynamic environments makes it suitable for large-scale industrial zones where mobile monitoring (e.g., drone-based surveillance) is required.

Overall, the results highlight YOLOv8's potential as a high-performance safety monitoring tool, with real-time capabilities and high detection accuracy making it a superior choice over older models like YOLOv4 and Faster R-CNN. However, continued research into low-light adaptation, scalability, and deployment on low-power devices is essential to maximize its effectiveness in real-world industrial applications.

DISCUSSION

The YOLOv8 model demonstrated several outstanding strengths, particularly its high accuracy in detecting helmets and fast inference speeds, making it well-suited for real-time safety monitoring. These advantages played a crucial role in reducing workplace accidents by ensuring workers complied with safety regulations. The updated architecture of YOLOv8 significantly enhanced its speed and efficiency without compromising detection quality, solidifying its suitability for industrial surveillance applications.

However, several limitations needed to be addressed. One of the primary challenges was the model's reliance on diverse and high-quality training datasets. While YOLOv8 performed robustly under various conditions, its accuracy was heavily dependent on the data used for training. If the dataset lacked sufficient variability, such as diverse lighting conditions, helmet colours, or worker positions, the model's performance could deteriorate, potentially increasing false negatives. To mitigate this issue, future research should focus on improving dataset diversity through synthetic data generation, domain adaptation, and more comprehensive real-world data collection.

Another notable limitation was the significant computational resources required for model training. The initial training phase demanded powerful hardware, which could be a barrier for factories with limited IT infrastructure or budget constraints. To address this, further optimization strategies, such as quantization, knowledge distillation, or deploying lightweight versions of YOLOv8 (e.g., YOLOv8-tiny), should be explored to enable efficient deployment on edge devices or lower-cost hardware.

CHALLENGES IN LOW-LIGHT CONDITIONS

A key weakness identified was YOLOv8's decline in performance under low-light conditions. The results showed a notable drop in mean Average Precision (mAP) and recall when visibility was reduced, with recall decreasing to 82.4%. This performance reduction highlighted the model's struggle with detecting objects in poorly lit environments, which could limit its effectiveness in certain factory settings, such as night shifts or areas with inadequate lighting.

To overcome this limitation, multiple strategies should be considered:

- Advanced Image Enhancement Techniques: Implementing contrast enhancement methods, such as histogram equalization, adaptive gamma correction, or deep learning-based low-light image enhancement, could help improve the visibility of objects before they are processed by YOLOv8.
- Infrared or Thermal Imaging Integration: Combining YOLOv8 with infrared or thermal cameras could enhance detection capabilities in environments with extremely low visibility.

- Specialized Data Augmentation: Incorporating low-light augmentation techniques, such as synthetic noise, blurring, or exposure simulation, during model training could improve robustness in challenging lighting conditions.

Addressing Detection of Small or Obstructed Objects

Another observed limitation was YOLOv8's difficulty in detecting small or partially obscured objects, such as helmets blocked by machinery. This challenge is common in object detection models, where occlusions and scale variations affect performance. To mitigate this, future work could explore techniques like:

- Attention Mechanisms: Implementing transformer-based models or attention mechanisms to improve object localization in complex scenes.
- Multi-Scale Feature Fusion: Enhancing the network architecture to better recognize small or occluded objects by integrating multi-scale feature fusion techniques.
- Improved Annotation Strategies: Expanding dataset annotations to include more diverse occlusion scenarios, ensuring the model learns to detect partially visible helmets effectively.

Despite these limitations, YOLOv8 exhibited strong advantages, including high accuracy, efficient real-time inference, and robustness across various scenarios. Its high mAP in both static and dynamic environments underscored its effectiveness in detecting safety helmets. The rapid inference times (as low as 18.3 Ms) made it a viable solution for real-time factory surveillance. However, addressing the identified weaknesses, particularly in low-light environments and detecting obscured objects, will be crucial for improving its performance and ensuring its broader adoption in industrial safety applications.

Conclusion

This study revealed that YOLOv8 performed excellently in detecting safety helmets, both in static and dynamic environments. The results showed high mAP and precision values, confirming YOLOv8's ability to detect objects with good accuracy. However, its performance slightly declined under low-light conditions, with recall dropping to 82.4%. With inference times ranging from 18.3 Ms to 21.5 Ms, YOLOv8 proved suitable for real-time applications.

Statistical analysis provided valuable insights into the model's performance. Paired t-tests showed that data augmentation significantly increased the mAP (mean Average Precision), reflecting an overall improvement in detection accuracy. In model comparisons, YOLOv8 achieved the highest mAP values across various scenarios, while YOLOv8's configuration demonstrated the fastest inference times. This made it the optimal choice for applications prioritizing speed. Additionally, Friedman tests confirmed significant performance variation between models in different environmental conditions, particularly in low-light scenarios, highlighting the importance of adjusting models for limited lighting conditions.

These findings emphasized the importance of data preprocessing techniques, such as augmentation or domain adaptation, to ensure the model's resilience in dynamic factory environments. The performance differences between YOLOv8 configurations provided practical guidance in selecting models based on specific application needs. If accuracy is prioritized, configurations maximizing mAP should be chosen, while for real-time applications requiring high speed, configurations with the fastest inference times are recommended.

Overall, this research supported the claim that YOLOv8 is an excellent solution for factory safety surveillance. The model demonstrated superior capability in detecting objects like helmets with high accuracy and time efficiency, making it an ideal solution for applications requiring high precision and reliability.

Future Work

To enhance the applicability and robustness of YOLOv8 in industrial safety surveillance, the following future research directions are recommended:

- Integration with IoT: Implementing YOLOv8 in IoT-based systems to enable real-time monitoring and automated safety enforcement in factories. This would improve response times to potential hazards and enhance situational awareness.
- Model Optimization for Low-Light Conditions: Given the performance drop observed in low-light environments (as discussed in item #15), future research should explore advanced techniques such as adaptive image enhancement, improved data augmentation, or training with domain adaptation strategies to improve detection accuracy under limited lighting conditions.
- Large-Scale Implementation and Generalization: Expanding the study to various industrial settings with different lighting, object densities, and environmental conditions. This would validate YOLOv8's adaptability and ensure its robustness across different real-world scenarios.
- User Interface Development: Creating an intuitive and user-friendly interface for factory personnel to deploy and interact with YOLOv8-based surveillance systems efficiently. This would bridge the gap between research and practical implementation.

References

Ahmed, M. I. B., Saraireh, L., Rahman, A., Al-Qarawi, S., Mhran, A., Al-Jalaoud, J., Al-Mudaifer, D., Al-Haidar, F., AlKhulaifi, D., Youldash, M., & Gollapalli, M. (2023). Personal protective equipment detection: A deep-learning-based sustainable approach. *Sustainability, 15*(18), 13990. https://doi.org/10.3390/su151813990

Alharbi, F., Alshahrani, R., Zakariah, M., Aldweesh, A., & Alghamdi, A. A. (2023). YOLO and blockchain technology applied to intelligent transportation license plate character recognition for security. *Computers, Materials & Continua, 77*(3), 3697–3722. https://doi.org/10.32604/cmc.2023.040086

Alyushin, M. V., Alyushin, V. M., & Kolobashkina, L. V. (2018). Optimization of the data representation integrated form in the Viola-Jones algorithm for a person's face search. *Procedia Computer Science, 123*, 18–23. https://doi.org/10.1016/j.procs.2018.01.004

Azhari, A. N., & Wahyono, W. (2022). Automatic detection of helmets on motorcyclists using Faster-RCNN. *Indonesian Journal of Computing and Cybernetics Systems, 16*(4), 357–366. https://doi.org/10.22146/ijccs.68245

Balasundaram, A., Mohanty, A., Shaik, A., Pradeep, K., Vijayakumar, K. P., & Kavitha, M. S. (2023). Zero-DCE++ inspired object detection in less illuminated environment using improved YOLOv5. *Computers, Materials and Continua, 77*(3), 2751–2769. https://doi.org/10.32604/cmc.2023.044374

Bourahla, A., Fernandes, G., & Ferreira, L. M. D. F. (2024). Managing occupational health and safety risks in construction projects to achieve social sustainability: A review of literature. *Procedia Computer Science, 239*, 1053–1061. https://doi.org/10.1016/j.procs.2024.06.269

Curcuruto, M., & Griffin, M. A. (2023). Upward safety communication in the workplace: How team leaders stimulate employees' voice through empowering and monitoring supervision. *Safety Science, 157*, 105947. https://doi.org/10.1016/j.ssci.2022.105947

Dang, L. M., Min, K., Wang, H., Piran, M. J., Lee, C. H., & Moon, H. (2020). Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition, 108*, 107561. https://doi.org/10.1016/j.patcog.2020.107561

Gallo, G., Di Rienzo, F., Garzelli, F., Ducange, P., & Vallati, C. (2022). A smart system for personal protective equipment detection in industrial environments based on deep learning at the edge. *IEEE Access, 10*, 110862–110878. https://doi.org/10.1109/ACCESS.2022.3215148

Gang, L., Hu, Y., Chen, Z., Guo, J., & Ni, P. (2023). Lightweight object detection algorithm for robots with improved YOLOv5. *Engineering Applications of Artificial Intelligence, 123*(Part A), 106217. https://doi.org/10.1016/j.engappai.2023.106217

Guo, C., Ren, K., & Chen, Q. (2024). YOLO-SGF: Lightweight network for object detection in complex infrared images based on improved YOLOv8. *Infrared Physics & Technology, 142,* 105539. https://doi.org/10.1016/j.infrared.2024.105539

Hsieh, T.-Y., Tsai, C.-Y., Hou, S.-J., & Chao, W.-J. (2023). Cost-effective memory protection and reliability evaluation based on machine error-tolerance: A case study on no-accuracy-loss YOLOv4 object detection model. *Microelectronics Reliability, 147,* 115039. https://doi.org/10.1016/j.microrel.2023.115039

Hu, Q., Shen, X., Qian, X., Huang, G., & Yuan, M. (2023). The personal protective equipment (PPE) based on individual combat: A systematic review and trend analysis. *Defence Technology, 28*, 195–221. https://doi.org/10.1016/j.dt.2022.12.007

Huang, C., Chen, M., & Wang, L. (2024). Semi-supervised surface defect detection of wind turbine blades with YOLOv4. *Global Energy Interconnection, 7*(3), 284–292. https://doi.org/10.1016/j.gloei.2024.06.010

Hussein, I. J., Burhanuddin, M. A., Mohammed, M. A., Elhoseny, M., Garcia-Zapirain, B., & Maashi, M. S. (2020). Fully automatic segmentation of gynaecological abnormality using a new Viola–Jones model. *Computers, Materials & Continua, 66*(3), 3161–3182. https://doi.org/10.32604/cmc.2021.012691

Jakubec, M., Lieskovska, E., Brezani, A., & Tothova, J. (2023). Deep learning-based automatic helmet recognition for two-wheeled road safety. *Transportation Research Procedia, 74*, 1171–1178. https://doi.org/10.1016/j.trpro.2023.11.258

Jinwoo, P., Lee, J., & Jeong, J. (2024). YOLOv5 based object detection in reel package X-ray images of semiconductor component. *Heliyon, 10*(5), e26532. https://doi.org/10.1016/j.heliyon.2024.e26532

Jrondi, Z., Moussaid, A., & Hadi, M. Y. (2024). Exploring end-to-end object detection with transformers versus YOLOv8 for enhanced citrus fruit detection within trees. *Systems and Soft Computing, 6,* 200103. https://doi.org/10.1016/j.sasc.2024.200103

Kamaluddin, N. A., D'Agostino, C., Laureshyn, A., & Várhelyi, A. (2023). Modelling of motorcyclists' risky behaviour at an urban T-junction using generalised linear model: An exploratory study. *IATSS Research, 47*(1), 94–104. https://doi.org/10.1016/j.iatssr.2023.02.001

Kamm, S., Sahlab, N., Jazdi, N., & Weyrich, M. (2023). A concept for dynamic and robust machine learning with context modeling for heterogeneous manufacturing data. *Procedia CIRP, 118*, 354–359. https://doi.org/10.1016/j.procir.2023.06.061

Kaur, G., Sinha, R., Tiwari, P. K., Yadav, S. K., Pandey, P., Raj, R., Vashisth, A., & Rakhra, M. (2022). Face mask recognition system using CNN model. *Neuroscience Informatics, 2*(3), 100035. https://doi.org/10.1016/j.neuri.2021.100035

Khan, S. I., Ray, B. R., & Karmakar, N. C. (2024). RFID localization in construction with IoT and security integration. *Automation in Construction, 159*, 105249. https://doi.org/10.1016/j.autcon.2023.105249

Laakel Hemdanou, A., Lamarti Sefian, M., Achtoun, Y., & Tahiri, I. (2024). Comparative analysis of feature selection and extraction methods for student performance prediction across different machine learning models. *Computers and Education: Artificial Intelligence, 7*, 100301. https://doi.org/10.1016/j.caeai.2024.100301

Li, H., Wu, D., Zhang, W., & Xiao, C. (2024). YOLO-PL: Helmet wearing detection algorithm based on improved YOLOv4. *Digital Signal Processing, 144,* 104283. https://doi.org/10.1016/j.dsp.2023.104283

Li, S., Huang, H., Meng, X., Wang, M., Li, Y., & Xie, L. (2023). A glove-wearing detection algorithm based on improved YOLOv8. *Sensors, 23*(24), 9906. https://doi.org/10.3390/s23249906

Liu, H., Zhong, Z., Sebe, N., & Satoh, S. (2023). Mitigating robust overfitting via self-residual-calibration regularization. *Artificial Intelligence, 317,* Article 103877. https://doi.org/10.1016/j.artint.2023.103877

Manakitsa, N., Maraslidis, G. S., Moysis, L., & Fragulis, G. F. (2024). A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision. *Technologies, 12*(2), 15. https://doi.org/10.3390/technologies12020015

Nguyen, M. T., Truong, L. H., Tran, T. T., & Chien, C.-F. (2020). Artificial intelligence-based data processing algorithm for video surveillance to empower Industry 3.5. *Computers & Industrial Engineering, 148*, 106671. https://doi.org/10.1016/j.cie.2020.106671

Ningsih, E. A., Diawati, L., Sari, H., & Bahagia, S. N. (2024). The effectiveness of safeguard measures in elevating the competitiveness of domestic industry: Case study of Indonesia's textile industry. *Procedia Computer Science, 234*, 699–709. https://doi.org/10.1016/j.procs.2024.03.056

Park, D. G., Jung, T. N., Kim, J. G., Lee, S. H., Oh, E. S., & Kim, D. H. (2024). DBSCAN and YOLOv5 based 3D object detection and its adaptation to a mobile platform. *Mechatronics, 103*, 103238. https://doi.org/10.1016/j.mechatronics.2024.103238

Park, M., Tran, D. Q., Bak, J., Kulinan, A. S., & Park, S. (2023). Real-time monitoring unsafe behaviors of portable multi-position ladder worker using deep learning based on vision data. *Journal of Safety Research, 87*, 465–480. https://doi.org/10.1016/j.jsr.2023.08.018

Rane, N., Choudhary, S., & Rane, J. (2023). YOLO and Faster R-CNN object detection in architecture, engineering, and construction (AEC): Applications, challenges, and future prospects. *SSRN*. https://doi.org/10.2139/ssrn.4624204

Rasouli, S., Alipouri, Y., & Chamanzad, S. (2024). Smart personal protective equipment (PPE) for construction safety: A literature review. *Safety Science, 170*, 106368. https://doi.org/10.1016/j.ssci.2023.106368

Ren, T., He, J., Jia, Z., Huang, X., Song, S., Wang, J., Zhou, G., Shi, F., & Lv, M. (2024). Nighttime large-field video image change detection based on adaptive superpixel reconstruction and multi-scale singular value decomposition fusion. *Displays, 85*, 102840. https://doi.org/10.1016/j.displa.2024.102840

Said, Y., Alassaf, Y., Ghodhbani, R., Alsariera, Y. A., Saidani, T., Ben Rhaiem, O., Makhdoum, M. K., & Hleili, M. (2024). AI-based helmet violation detection for traffic management system. *Computer Modeling in Engineering and Sciences, 141*(1), 733–749. https://doi.org/10.32604/cmes.2024.052369

Sarvari, H., Edwards, D. J., Rillie, I., & Posillico, J. J. (2024). Building a safer future: Analysis of studies on safety I and safety II in the construction industry. *Safety Science, 178*, 106621. https://doi.org/10.1016/j.ssci.2024.106621

Shamsollahi, D., Moselhi, O., & Khorasani, K. (2024). Data integration using deep learning and real-time locating system (RTLS) for automated construction progress monitoring and reporting. *Automation in Construction, 168*(Part A), 105778. https://doi.org/10.1016/j.autcon.2024.105778

Sharma, P., Gupta, S., Vyas, S., & Shabaz, M. (2023). Retracted: Object detection and recognition using deep learning-based techniques. *IET Communications, 17*(13), 1589–1599. https://doi.org/10.1049/cmu2.12513

Shuaishuai, L., Liu, Y., Wang, Z., Dou, C., Zhao, W., & Shu, H. (2024). Constructing a visual detection method for coagulation effect based on image feature machine learning. *Journal of Water Process Engineering, 68*, 106354. https://doi.org/10.1016/j.jwpe.2024.106354

Tang, J., Ye, C., Zhou, X., & Xu, L. (2024). YOLO-Fusion and Internet of Things: Advancing object detection in smart transportation. *Alexandria Engineering Journal, 107*, 1–12. https://doi.org/10.1016/j.aej.2024.09.012

Wan, D., Lu, R., Hu, B., Yin, J., Shen, S., Xu, T., & Lang, X. (2024). YOLO-MIF: Improved YOLOv8 with multi-information fusion for object detection in gray-scale images. *Advanced Engineering Informatics, 62*(B), 102709. https://doi.org/10.1016/j.aei.2024.102709

Wang, W., Meng, Y., Li, S., & Zhang, C. (2024). HV-YOLOv8 by HDPconv: Better lightweight detectors for small object detection. *Image and Vision Computing, 147,* 105052. https://doi.org/10.1016/j.imavis.2024.105052

Yan, J., Zeng, Y., Lin, J., Pei, Z., Fan, J., Fang, C., & Cai, Y. (2024). Enhanced object detection in pediatric bronchoscopy images using YOLO-based algorithms with CBAM attention mechanism. *Heliyon, 10*(12), e32678. https://doi.org/10.1016/j.heliyon.2024.e32678

Yao, X., Ma, C., & Jia, L. (2023). Moving personnel detection for trackless rubber-tyred vehicles in coal mines based on infrared images. *Journal of Engineering Research, 11*(4), 437–446. https://doi.org/10.1016/j.jer.2023.100094

Zhao, A. A. Z., & Zietsch, B. P. (2024). Deep neural networks generate facial metrics that overcome limitations of previous methods and predict in-person attraction. *Evolution and Human Behavior, 45*(6), 106632. https://doi.org/10.1016/j.evolhumbehav.2024.106632

Zhao, L., Liu, J., Ren, Y., Lin, C., Liu, J., Abbas, Z., Islam, M. S., & Xiao, G. (2024). YOLOv8-QR: An improved YOLOv8 model via attention mechanism for object detection of QR code defects. *Computers and Electrical Engineering, 118(Part B),* 109376. https://doi.org/10.1016/j.compeleceng.2024.109376

Zheng, H., Wu, H., Yin, H., Wang, Y., Shen, X., Fang, Z., Ma, D., Miao, Y., Zhou, L., Yan, M., Sun, J., Ding, X., Yu, C., & Lu, C. (2024). Novel mining conveyor monitoring system based on quasi-distributed optical fiber accelerometer array and self-supervised learning. *Mechanical Systems and Signal Processing, 221*, 111697. https://doi.org/10.1016/j.ymssp.2024.111697

Zhou, H., & Yu, G. (2021). Research on pedestrian detection technology based on the SVM classifier trained by HOG and LTP features. *Future Generation Computer Systems, 125*, 604–615. https://doi.org/10.1016/j.future.2021.06.016

AUTHORS

Ratna Santi graduated with her bachelor's degree in Electrical Engineering from Sunan Gunung Djati State Islamic University in 2014. Currently, she is a Master's candidate in the School of Informatics Engineering, Langlangbuana University Bandung, and works as a Commissioner at PT Sumapala Integrasi Solusi. Her research interests are in artificial intelligence. She can be contacted at ratna.s.kardi@gmail.com.

Wiwin Suwarningsih earned her master's degree in Informatics in 2000 and completed her doctorate in 2017 from the School of Electrical Engineering and Informatics, Bandung Institute of Technology. She is currently affiliated with the Data and Information Science Research Center at the National Research and Innovation Agency (BRIN), Indonesia. She is also a lecturer in the Master's Program in Informatics Engineering at Langlangbuana University. Her research interests include artificial intelligence, computational linguistics, image processing, data mining, information retrieval, smart systems, and question answering systems. For further inquiries, she can be contacted at wiwi005@brin.go.id

Aswin Sasongko Sastrosubroto received his B.E. in Electrical Engineering from the Bandung Institute of Technology, Bandung, Indonesia, and his MSc and PhD in power electronics from Aston University, Birmingham, U.K. He has served as the Director General of Informatics Applications with the Ministry of Communications and Information Technology (MCIT). He served in various positions with the government as the Secretary General of the MCIT Ministry of Communication and Information Technology, the Secretary to the Ministry of Research and Technology, the Deputy Minister with the State Ministry for Communication and Information, and the Vice Chairman of the Agency of Application and Assessment of Technology. He is currently a senior lecturer at several universities, including Langlangbuana University, Bandung. He is also a member of the Indonesian National ICT Council (ICT_Information and Communication Technology). His current interests are in ICT systems and governance. He can be contacted at ashwin.sasongko@gmail.com