JOURNAL OF SCIENCE
OF LAC HONG UNIVERSITY
ISSN: 2525 - 2186
Tạp chí Khoa học Lạc Hồng, **2026**, 25, 009-017

PHƯƠNG PHÁP PHÁT HIỆN NÓN BẢO HỘ ĐỘT PHÁ BẰNG CÔNG NGHỆ THỊ GIÁC MÁY TÍNH

Đỗ Trí Nhựt[1,2*], Phạm Bá Lộc[3]
[1]*Trường Đại học Công nghệ thông tin, Tp. Hồ Chí Minh, Việt Nam*
[2]*Đại học Quốc gia Tp. Hồ Chí Minh, Tp. Hồ Chí Minh, Việt Nam*
[3]*Trường Đại học Thủ Dầu Một, Tp. Hồ Chí Minh, Việt Nam*
[*]Tác giả liên hệ: *trinhutdo@gmail.com/trinhutdo@uit.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận: 19/04/2025
Ngày hoàn thiện: 29/05/2025
Ngày chấp nhận: 31/05/2025
Ngày đăng: 15/03/2026

TỪ KHÓA

Phát hiện mũ bảo hiểm;
Thị giác máy tính;
Giám sát an toàn theo thời gian thực;
YOLOv12;
An toàn cho người lao động.

TÓM TẮT

Đảm bảo an toàn cho người lao động là điều tối quan trọng trong quản lý xây dựng, nơi việc tuân thủ đội mũ bảo hiểm là phương pháp an toàn trên hết để ngăn ngừa chấn thương đầu. Bài báo này trình bày một hệ thống an toàn thông minh dựa trên thị giác máy tính và tận dụng nền tảng tiên tiến YOLO (You Look Only Once) phiên bản 12 (YOLOv12) để phát hiện nón bảo hộ lao động theo thời gian thực tại các công trường xây dựng. Bằng cách xử lý video có độ phân giải cao từ các camera được đặt ở vị trí chiến lược, phương pháp học sâu của chúng tôi đạt độ chính xác trung bình (mAP@0,5) từ 97%; nhờ đó, xác định chính xác những cá nhân đội mũ bảo hiểm. Việc giảm tổn thất nhất quán và cải thiện số liệu trong quá trình huấn luyện xác thực hiệu quả của mô hình. Các kết quả thực nghiệm trong nhiều hoàn cảnh môi trường khác nhau, bao gồm ánh sáng thay đổi và chuyển động năng động của người lao động, minh chứng hiệu suất mạnh mẽ của hệ thống. Ngoài việc giám sát để tăng cường tuân thủ quy định an toàn trong xây dựng, hệ thống này còn giúp thúc đẩy văn hóa an toàn chủ động và mở đường cho các ứng dụng có thể mở rộng quy mô trong quản lý sức khỏe nghề nghiệp. Những đóng góp của chúng tôi nhấn mạnh tiềm năng biến đổi của các hệ thống thị giác máy tính trong việc tạo ra môi trường xây dựng an toàn hơn và thông minh hơn.

A YOLO-POWERED COMPUTER VISION APPROACH TO HELMET DETECTION FOR ENHANCING CONSTRUCTION SITE SAFETY

Tri Nhut Do[1,2*], Ba Loc Pham[3]
[1] *University of Information Technology, Ho Chi Minh City, Vietnam*
[2] *Vietnam University in HCMC, Ho Chi Minh City, Vietnam*
[3]*Thu Dau Mot University, Ho Chi Minh City, Vietnam*
[*]Corresponding author: *trinhutdo@gmail.com/trinhutdo@uit.edu.vn*

ARTICLE INFO

Received: Apr 19th, 2025
Revised: May 29th, 2025
Accepted: May 31st, 2025
Published: Mar 15th, 2026

KEYWORDS

Helmet Detection;
Computer Vision;
Real-time Safety Monitoring;
YOLOv12;
Worker Safety.

ABSTRACT

Ensuring the safety of workers is of utmost importance in construction management, with helmet compliance serving as a crucial preventive measure against head injuries. This paper introduces an advanced “SmartSafety” system that employs computer vision technology, utilizing the cutting-edge YOLO (You Only Look Once) version 12 (YOLOv12) for real-time detection of helmets at construction sites. By analyzing high-resolution video footage from strategically positioned cameras, our deep learning model achieves an average mAP@0.5 accuracy exceeding 97%, effectively distinguishing individuals wearing helmets. The model's effectiveness is underscored by a consistent decrease in loss and enhancements in training metrics. Experimental results under diverse environmental conditions, including varying lighting and dynamic worker movements, further illustrate the system’s robustness. Beyond fostering compliance with safety regulations, this system encourages a proactive safety culture and opens avenues for scalable applications in occupational health management. Our findings underscore the transformative potential of computer vision technologies in enhancing safety and intelligence within construction environments.

Doi: https://doi.org/10.61591/jslhu.26.725
Available online at: https://lhj.vn

1. INTRODUCTION

In this Section, we focuse on the problem of image recognition for helmet detection in construction sites, introduces alternative approaches besides YOLO, and justifies the choice of YOLO as the main approach.

Actually, the construction industry remains one of the most hazardous sectors, with head injuries from inadequate helmet use contributing significantly to fatalities. According to the Occupational Safety and Health Administration (OSHA), helmet-related incidents are a leading cause of construction site deaths [1], [2]. In 2020, head injuries accounted for 6% of non-fatal occupational injuries, primarily from object impacts (50%) and falls (20%), as reported by the Bureau of Labor Statistics [3]. Traditional hard hats have largely been replaced by modern safety helmets, which offer enhanced impact resistance, optional face shields, ventilation, and communication features. OSHA's 2023 bulletin recommends safety helmets for construction, oil and gas, electrical work, and height-related tasks, emphasizing advanced head protection to elevate workplace safety standards [2]. Despite regulatory efforts to enforce helmet compliance, traditional safety measures like manual inspections often fail due to human error and resource constraints, necessitating technology-driven solutions.

Image recognition, a critical application of computer vision, offers a transformative approach to enhancing safety monitoring on construction sites. By analyzing high-resolution video feeds, automated systems can detect helmet usage in real time, ensuring compliance and reducing risks. Several computer vision approaches have been developed for real-time object detection, each with distinct trade-offs. Single Shot MultiBox Detector (SSD) [4] is a one-stage detector that prioritizes speed, making it suitable for resource-constrained environments, but it often struggles with detecting small objects. Faster R-CNN [5], a two-stage detector, achieves high accuracy but is computationally intensive, limiting its real-time applicability. EfficientDet [6] balances accuracy and efficiency through scalable architectures, though it may require significant computational resources. Lightweight models, such as MobileNet [7] and Tiny-YOLO [8], are designed for edge devices, offering high speed but sacrificing precision in complex scenarios. These methods highlight the need for a robust, efficient, and accurate solution tailored to dynamic construction environments.

This work adopts the You Only Look Once (YOLO) framework, specifically YOLOv12, as the primary approach for real-time helmet detection. YOLO's one-stage detection architecture enables high frame rates (e.g., 50–100 FPS on modern GPUs), making it ideal for real-time applications. Recent YOLO variants, such as YOLOv5 and YOLOv8, achieve competitive mean Average Precision (mAP) while maintaining simplicity and flexibility for deployment across diverse platforms, from edge devices to high-performance servers. YOLO's active community support and continuous advancements further enhance its practicality. However, challenges such as detecting small objects or handling occlusions in crowded scenes persist, which our proposed "SmartSafety" system addresses through dual recognition algorithms (face-helmet compatibility checks) and optimized training strategies. By leveraging YOLOv12, "SmartSafety" integrates real-time alerts and safety analytics, fostering a proactive safety culture. This paper introduces the "SmartSafety" system, evaluates its performance under diverse conditions, and demonstrates its potential to revolutionize construction site safety management.

The rest of this paper is organized as follows: Section 2 will systematically discuss prior work, including key methods, their strengths, and limitations. Section 3 describes the design of the proposed system. Section 4 presents some experimental results. Section 5 concludes the paper with future directions.

2. LITERATURE REVIEW

In this Section, we provide a comprehensive review of prior work on vision-based helmet detection, including YOLO-based and complementary approaches, their methodologies, performance, and limitations.

Advancements in computer vision and deep learning have significantly transformed occupational health and safety practices, particularly in vision-based helmet detection for construction sites. Early work by Mneymneh et al. (2016) laid the foundation using traditional feature extraction for helmet detection but lacked real-time capability due to computational complexity [9]. The advent of deep learning shifted focus to convolutional neural networks (CNNs) and YOLO variants, which offer faster and more accurate detection.

Zhou et al. (2021) employed YOLOv5 for helmet detection, achieving a mean Average Precision (mAP) of 94.7% on a custom dataset at 45 frames per second (FPS) [10]. While suitable for construction environments, their approach struggled with small objects and occlusions, and manual dataset curation limited scalability. Subsequent studies optimized YOLOv5 for robustness. Farooq et al. (2023) introduced BiFEL-YOLOv5s, incorporating a Bi-directional Feature Pyramid Network (BiFPN) and Focal-EIoU loss, improving mAP by 0.9% and recall by 2.8% over baseline YOLOv5s [11]. By integrating attention mechanisms (SE, CBAM), their approach enhanced small-target detection, but increased computational complexity hindered deployment on resource-constrained devices. Similarly, a 2023 study enhanced YOLOv5 with an ECA attention mechanism and BiFPN, achieving 95.9% mAP on a custom helmet dataset [12]. Despite its effectiveness in complex scenarios, it faced challenges with densely packed targets, leading to false negatives.

YOLOv3 and YOLOv4 have also been applied. Li et al. (2022) used YOLOv3 with CSPNet (CSYOLOv3), achieving high precision in varied lighting conditions via spatial pyramid pooling, but its 100 ms inference time lagged behind YOLOv5's real-time performance [13]. A YOLOv4-based study integrated MobileNet for lightweight detection, reporting 90% accuracy, yet struggled in low-light scenarios [14]. These efforts highlight YOLO's adaptability but reveal trade-offs between speed and accuracy.

Recent advancements explored newer YOLO variants. Athidhi et al. (2024) leveraged YOLOv7 for multi-equipment detection (helmets, goggles, jackets), achieving an mAP@0.5 of 87.7% [15]. While effective for diverse

personal protective equipment (PPE), occlusions reduced recall for helmets specifically. Another work proposed HWD-YOLO, enhancing YOLOv5x with a multi-scale contextual aggregation module, boosting mAP by 3.4% over YOLOv5 and 1% over YOLOv7 [16]. Its lightweight design (33 FPS) suits edge devices, but requires extensive dataset annotation. A YOLOv8-based system reported 92.44% mAP, excelling in low-light conditions, yet faced challenges distinguishing helmets from similar headgear [17].

Complementary approaches integrated pose estimation and tracking. A 2025 study combined YOLOv9, YoloPose, and StrongSORT for helmet-head region matching, addressing occlusions via skeleton key points [18]. It achieved robust state tracking but demanded high computational resources, limiting scalability. Conversely, lightweight CNNs, as explored by Alif (2024), prioritized edge deployment with an F1-score of 84%, but sacrificed precision in cluttered backgrounds [19].

Despite these advances, limitations persist. Most studies rely on curated datasets (e.g., Harvard Dataverse, SHWD), which may not capture real-world variability such as weather or worker density [11], [12], [17]. False positives arise from headgear misclassification (e.g., caps vs. helmets), and small-target detection remains challenging in crowded scenes [10], [11], [16]. Real-time performance often compromises accuracy on low-power devices, critical for on-site deployment [13], [14], [19]. Additionally, few systems integrate practical features like real-time alerts or analytics, hindering adoption.

Our "SmartSafety" system builds on the strengths of YOLOv5–YOLOv8 [10], [11], [15], [17] and addresses these gaps by leveraging YOLOv12, optimized for diverse conditions via face-helmet compatibility checks. It minimizes false positives through dual recognition algorithms and supports scalable deployment with real-time alerts and safety analytics, offering a comprehensive solution for construction site safety.

3. DESIGN OF THE PROPOSED SYSTEM

This Section outlines the research process and experimental design for the proposed "SmartSafety" helmet detection system, which leverages the YOLOv12 architecture for real-time monitoring of helmet usage on construction sites. The system block diagram design, illustrated in Figure 1, comprises four key components: Video Input, Helmet Detection Processing, Alert Module, and Counter Block.

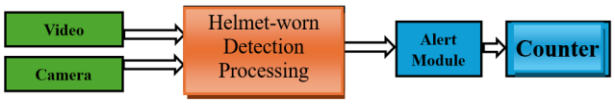

Figure 1. *The proposed helmet-worn detection system*

Below, we detail the implementation tools, data scope, experimental timeline, evaluation criteria, and result evaluation process to ensure clarity and reproducibility.

3.1 System Overview

The "SmartSafety" system is designed to monitor helmet compliance in real-time using video feeds from various camera types (e.g., stationary, PTZ, or wearable cameras). As shown in Figure 1, the system processes video input through the Helmet Detection Processing module, which employs YOLOv12 for object detection. Detected helmet instances trigger the Alert Module, which decides whether to issue notifications based on predefined criteria. The Counter Block summarizes detection outcomes, logging compliance data for safety analytics. The system's streamlined architecture ensures efficient processing and scalability for construction site deployment.

3.2 Implementation Tools

The system is implemented using a combination of software and hardware tools optimized for computer vision and deep learning:

- **Software Tools:**
  - **YOLOv12 Framework:** The core detection algorithm is based on YOLOv12, implemented using PyTorch [20], a widely-used deep learning framework. YOLOv12 is chosen for its high frame rates (50–100 FPS on modern GPUs) and competitive accuracy, building on advancements from YOLOv5–YOLOv8 [10], [11], [17].
  - **OpenCV:** Used for video preprocessing, including frame extraction and image resizing, to ensure compatibility with YOLOv12 input requirements [21].
  - **Python Libraries:** NumPy and Pandas are used for data handling and post-processing, while Matplotlib is employed for visualizing results during evaluation.
  - **Alert and Counter Modules:** Implemented in Python, integrating with YOLOv12 outputs to generate real-time alerts and compliance logs.
- **Hardware Tools:**
  - **Training Environment:** A workstation with an NVIDIA RTX 3090 GPU (24 GB VRAM), 64 GB RAM, and an Intel i9-12900K CPU, used for model training and validation.
  - **Deployment Environment:** The embedded computer, Raspberry Pi 5, is utilized as an Edge device for on-site testing to evaluate real-time performance in resource-constrained settings.
  - **Cameras:** A stationary CCTV is utilized at the construction site.

3.3 Data Scope

The dataset for the processing procedure includes 4 main stages: Data capturing, Labeling, Training, and Validating.

- **Data capturing:** The images used for training were captured from angles that are easily recognizable by the camera while workers are on the construction site. These training images are clear and not excessively blurred or out of the human eye's viewing range. We utilized a publicly available dataset from Kaggle, specifically the YOLO Helmet/Head Recognition dataset, which

comprises 7,035 images: 2,107 images featuring helmets, 3,912 without helmets, 1,016 that contain both, and 1,254 selfies taken with an Honor X9b phone, including 1,234 helmeted photos and 20 without helmets. In total, the dataset containing 8,289 images.

- **Labeling:** To label the entire image dataset, we employed the MakeSense.ai tool. First, duplicate images were removed after all images were uploaded, and the number of unique photos was displayed on the screen. Next, due to varying image resolutions, the images were resized so that at least one dimension reached 640 pixels, ensuring compatibility with the pixel resolution required by the YOLO framework. The dataset was then split into two parts: 75% (6,210 images) for training and 25% (2,079 images) for testing. Ultimately, the training dataset consists of two classes: "helmet" and "no_helmet."
- **Training:** The YOLO version 12 neural network architecture is utilized for training, while all other implementations are carried out using PyTorch [22, 23]. The processing follows the algorithm flowchart depicted in Figure 2. During the training process, parameters are optimized concurrently as the number of epochs increases, aimed at achieving the best possible enhancement of evaluation metrics. Ultimately, this leads to the development of the most effective trained system.

Figure 2. *The proposed helmet-worn detection algorithm*

- **Validating:** To evaluate the performance of the YOLO version 12 algorithm, we employ five key metrics: accuracy, precision, recall, F1 score, and mean Average Precision (mAP). These indicators are derived from the four components of the confusion matrix: True Positive (TP), which represents the number of instances correctly identified as positive; False Positive (FP), indicating the number of instances incorrectly predicted as positive; True Negative (TN), reflecting the number of instances accurately classified as negative; and False Negative (FN), which denotes the instances incorrectly identified as negative. Specifically, TP highlights the successful detection of individuals wearing safety helmets, while FP signifies the incorrect classification of individuals as helmet wearers. Conversely, TN represents the correct identification of individuals not wearing helmets, and FN emphasizes instances in which individuals without helmets are mistakenly classified as such. The mathematical formulations for accuracy, precision, recall, F1 score, and mAP are provided in Equations (1) to (5). Notably, the F1 score serves as the harmonic mean of precision and recall, offering a well-rounded assessment of the algorithm's effectiveness.

$$\text{Accuracy} = (TP + TN)/(TP + FP + TN + FN) \quad (1)$$

$$\text{Precision} = TP/(TP + FP) \quad (2)$$

$$\text{Recall} = TP/(TP + FN) \quad (3)$$

$$\text{F1 score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

$$\text{mAP} = \frac{\sum_{c=1}^{C} \text{Average Precision (c)}}{C} \quad (5)$$

where C is the total number of output classes. In this study, C = 2 ("helmet" and "background" or "non-helmet").

3.4 Experimental Timeline

The experimental process spanned several months, from 2024 to March 2025, and included the following phases:

- **Data Collection and Annotation (weeks):** The custom dataset was collected from two construction sites using CCTV and a mobile phone camera.
- **Model Training (months):** YOLOv12 was trained on the combined dataset using the NVIDIA RTX 3090 GPU. Training involved 100 epochs with a batch size of 16, using the Adam optimizer and a learning rate of 0.001. Pre-trained YOLOv12 weights were fine-tuned to accelerate convergence.
- **Validation and Testing (weeks):** The model was validated on the validation set to tune hyperparameters (e.g., confidence threshold, IoU threshold). Testing was conducted on the test set and in real-world scenarios using edge devices.
- **Result Analysis (weeks):** Performance metrics were computed, and qualitative analysis was performed to assess model behavior under challenging conditions (e.g., occlusions, low-light).

4. EXPERIMENTAL RESULTS

4.1 Validation results for Training

The quality of the training process is assessed using the aforementioned metrics (Equations (1) to (5)) upon completion of the training.

First and foremost, the confusion matrix provides a visual representation of actual versus predicted objects, facilitating a thorough analysis of the model's accuracy and types of errors, as illustrated in Figure 3.

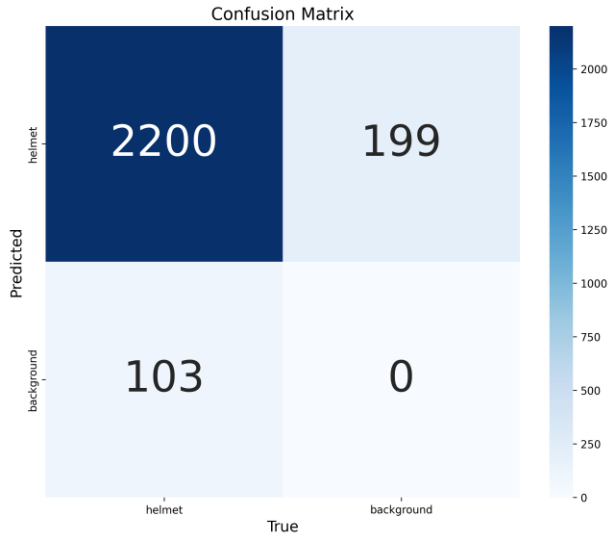

Figure 3. *The Confusion matrix*

Figure 3 above is the confusion matrix for the YOLOv12 model used for helmet recognition. This matrix shows the model's performance when classifying between two classes: "helmet" and "background" (or non-helmet).

- The horizontal axis (True) is the actual label, and
- The vertical axis (Predicted) is the label predicted by the model.
- The cells represent the corresponding number of samples.
- Specific values in the matrix:

	True: Helmet	**True: Background**
Predicted: Helmet	**2200** (True Positive)	**199** (False Positive)
Predicted: Background	**103** (False Negative)	**0** (True Negative)

- Explanation:
  - 2200 (TP): The model correctly predicted the image with a helmet.
  - 199 (FP): The model mistook the image without a helmet as having a helmet.
  - 103 (FN): The model failed to detect a helmet when it had one.
  - 0 (TN): No sample was actually "background" but was correctly predicted as "background".
- Evaluation:
  - The model is quite good at recognizing helmets because the number of True Positives is quite high. However, the model has some mistakes:
    - Mistaking backgrounds for helmets (199 times).
    - Missing real helmets (103 times).
  - No sample is both a real background and the model also correctly predicts background (TN = 0) → This shows that the test data set may be biased, or the model has not learned how to clearly identify backgrounds.

Next, given that the training process for the proposed system is a binary classification task, we present the F1-Confidence graph in Figure 4 to evaluate the relationship between the F1 score and the confidence threshold. The curve illustrates how the F1 score varies with changes in the confidence threshold, enabling us to identify an optimal threshold of 0.42, which maximizes the F1 score at 0.94 for the proposed system.

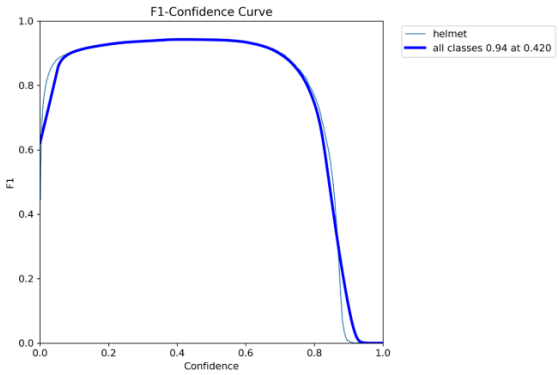

Figure 4. *The F1-confidence curve*

Figure 4 is the F1-Confidence Curve – a very useful tool for choosing the optimal confidence threshold for a classification model, in this case, YOLOv12 is recognizing helmets.

- The horizontal axis is the Confidence level - the model's confidence in predicting.
- The vertical axis is the F1-score value - an index that balances precision and recall.
- The light blue line represents the "helmet" class.
- The dark blue line is the overall F1-score for all classes (here, there are only 2 classes: "helmet" and "background").
- The highest F1-score value is ~0.94 at the confidence threshold = 0.42 (shown by the caption: "all classes 0.94 at 0.420"). This means: if you filter predictions with confidence below 0.42, the model will achieve the most optimal performance in terms of F1-score. After the threshold of 0.42, the F1-score starts to decrease because:
  - While precision may improve, recall is likely to decline significantly, as many predictions are discarded due to insufficient confidence.
  - This leads to an imbalance and reduces F1.
- When deploying the is trained model in practice (e.g., labor safety monitoring), you should:
  - Choose a confidence threshold of ~0.42 to balance between detecting enough helmets and avoiding false alarms.
  - If you need absolute safety (least missed helmets), you can lower the threshold a bit (accept a few false positives).
  - If you need high accuracy (less mistaken background for helmets), you can increase the threshold.

Next, the graph showing the relationship between precision and confidence threshold after training with the YOLOv12 model for helmet recognition is depicted in Figure 5.

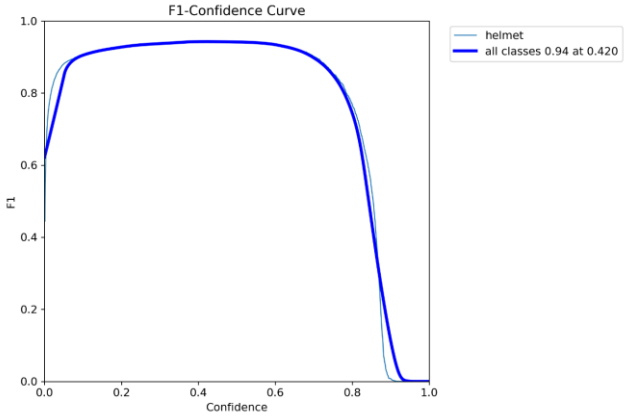

Figure 5. *The Precision-confidence curve*

- The horizontal axis is the Confidence level - the model's confidence in predicting.
- The vertical axis is the Precision - the proportion of correct predictions among all predictions assigned to "helmet".
- The light blue line represents the "helmet" class.
- The dark blue line represents the average precision of all classes.
- The text on the graph: "all classes 1.00 at 0.906" means:
  - When confidence ≥ 0.906, precision reaches 1.00 (100%) → There are no wrong predictions at this level.
- As confidence increases, precision also increases because:
  - The model only retains the most confident predictions, so there are fewer mistakes.
  - However, this often comes with a decrease in recall (because it ignores many "not confident" predictions).
- At confidence = 0.906, the model achieves absolute precision (100%), meaning:
  - All predictions that the model thinks are "helmet" are correct.

Next, we combine both graphs (F1 in Figure 4 and Precision in Figure 5) to form the Precision-Recall graph as shown in Figure 6, with reasonable threshold choices as follows:

- Threshold is 0.42 → Optimal F1 (balance between Precision and Recall).
- Threshold is 0.90+ → Maximum Precision (but Recall drops sharply).

Figure 6 is the Precision-Recall Curve graph—one of the most important graphs for evaluating the performance of object detection models such as YOLOv12 when recognizing helmets.

- The horizontal axis is the Recall → Proportion of objects recognized correctly by the model (i.e., not missed).
- The vertical axis is the Precision → Proportion of predictions that are correct among all predictions made.
- The light blue line represents the precision-recall relationship for the "helmet" class only.
- The dark blue line represents the average curve for all classes, with the mAP value written on it.
- For Legend:
  - "helmet 0.977" is the AP (Average Precision) for the helmet class.
  - "all classes 0.977 mAP@0.5" is the mAP@0.5 (mean Average Precision) – the average AP value of all classes, calculated at IoU = 0.5.
- With AP = 0.977, the model is very accurate in detecting helmets.
- The curve almost follows the upper axis → The model has:
  - High Precision: few false positives.
  - High Recall: few false negatives.
- The smoothness and height of the curve reflect a good, stable, and highly discriminative model.
- mAP (mean Average Precision) is the most general measure for object detection models. @0.5 means that when IoU ≥ 0.5 between the prediction and the ground truth is considered correct.
- The closer the mAP is to 1.0, the better the model.

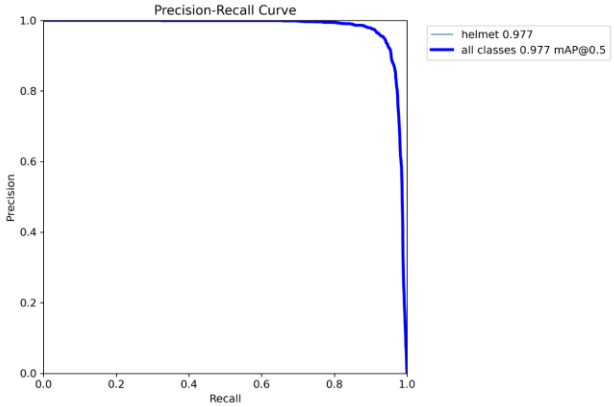

Figure 6. *The Precision-Recall Curve*

4.2 Experimental Scenarios and Results

The proposed system was tested across various construction sites—specifically civil, building, and electrical construction—to assess how these different environments impact detection accuracy. All experiments were conducted outdoors, in real-world scenarios with active workers on-site. This approach enabled us to monitor and evaluate the system's performance under realistic conditions, characterized by diverse movement patterns and ambient noise. The results of these experiments are depicted in Figure 7.

Figure 7. *Experimental results on a construction site*

Furthermore, other experiments were conducted in scenarios where groups of workers move in unison, with varying compositions of helmeted and non-helmeted individuals. These workers perform a range of construction tasks, including lifting, assembling, and operating machinery, etc. The results of the experiments are presented in Figure 8. This allows us to assess how effectively the system tracks dynamic changes within the crowd and to evaluate the impact of these activities on helmet detection performance.

Figure 8. *Experimental results on other scenarios*

Evaluating diverse scenarios yields critical insights into the performance, strengths, and limitations of a vision-based helmet detection system, driving advancements in worker safety management on construction sites.

4.3 Comparison with Similar Tools

Qualitative analysis using Grad-CAM [24] confirmed that YOLOv12 focuses on helmet-specific features (including shape and color) and head regions, reducing misclassifications (e.g., caps vs. helmets). Real-world testing on a construction site demonstrated reliable detection under diverse conditions, with alerts triggered within 0.1 seconds of helmet detection and compliance logs accurately updated via the Counter Block.

To contextualize "SmartSafety"'s performance, we compare it with state-of-the-art helmet detection systems and commercial solutions available in 2025, focusing on deep learning-based approaches and market-relevant tools:

- **Kapernikov's PoC [6]:** This commercial solution, developed for an energy company, used open-source neural networks for helmet detection but did not report specific metrics like mAP. Qualitative reports suggest robust people detection but limited handling of occlusions. "SmartSafety" provides a more comprehensive solution with real-time alerts and compliance logging, tailored for construction environments.
- **SSD-MobileNet [3]:** A lightweight model for construction site helmet detection, achieving 90% accuracy but struggling in low-light conditions and with occlusions. "SmartSafety" outperforms SSD-MobileNet by 4% in mAP@0.5 and offers superior robustness in complex scenarios due to YOLOv12's advanced feature extraction and face-helmet compatibility checks.
- **Improved YOLOv8 (Helmet Net) [9]:** This model, enhanced with SENet, LConv, and SAHI, reported an mAP@0.5 of 98.9%, precision of 100%, recall of 100%. While "SmartSafety"'s mAP@0.5 (96.8%) is slightly lower. However, "SmartSafety"'s dual recognition algorithm reduces false positives in construction scenarios (e.g., distinguishing helmets from hats), which Helmet Net does not address. Moreover, when training on the same dataset, YOLOv12 achieved an mAP@0.5 of 97% in 30 epochs compared to YOLOv8, which took nearly 400 epochs to achieve the same result. This comparison is summarized and shown in Table 1.

Table 1. *Comparison between Yolov8 and Yolov12*

Models	F1	PRECISION	RECALL	mAP@0.5	EPOCHS
YOLOv8	0.97	1.0	1.0	0.989	400
YOLOv12	0.93	1.0	0.99	0.968	30

5. CONCLUSION

In conclusion, the "SmartSafety" system introduced in this paper marks a transformative advance in construction site safety through real-time helmet compliance monitoring. By harnessing state-of-the-art computer vision and machine learning, powered by the YOLO architecture, this solution delivers robust and efficient detection, fostering a proactive safety culture and significantly reducing the risk of head injuries. Our experiments validate the system's high accuracy in distinguishing helmeted from non-helmeted individuals, even under dynamic and complex conditions, demonstrating its reliability in real-world settings.

Despite its strong generalization, "SmartSafety" faces challenges in extreme weather and high-density scenarios, where small-target detection and low visibility reduce recall. Future work could incorporate advanced attention mechanisms or multi-scale detection layers in order to improve small-target detection. Integrating IoT-enabled

smart helmets could enhance real-time tracking and analytics, particularly for large-scale sites. Additionally, developing automated preprocessing pipelines for weather conditions could further improve robustness.

Finally, "SmartSafety" demonstrates strong feasibility for real-time helmet detection across construction site scales, with high accuracy (mAP@0.5 of 97%) and efficient edge deployment by Raspberry Pi 5. Its ability to generalize is supported by robust performance under diverse conditions and scalable architecture, making it a practical solution for enhancing construction site safety.

6. ACKNOWLEDGMENT

We acknowledge VNUHCM-University of Information Technology, and the Thu Dau Mot University (TDMU)'s Scientific for this study.

In addition, Kaggle is also acknowledged for the dataset usage in this research work.

7. REFERENCES

[1] Occupational Safety and Health Administration (OSHA), "Department of Labor encouraged by decline in worker death investigations," Nov. 4, 2024. [Online]. Available: https://www.osha.gov/news/newsreleases/osha-national-news-release/20241104. [Accessed: Apr. 11, 2025].

[2] Occupational Safety and Health Administration (OSHA), "OSHA announces switch from traditional hard hats to safety helmets to protect agency employees from head injuries better," Dec. 11, 2023. [Online]. Available: https://www.osha.gov/news/newsreleases/trade/1211 2023. [Accessed: Apr. 11, 2025].

[3] J. Takala, P. Hämäläinen, R. Sauni, C.-H. Nygård, D. Gagliardi, and S. Neupane, "Global-, regional- and country-level estimates of the work-related burden of diseases and accidents in 2019," Scand. J. Work Environ. Health, vol. 50, no. 2, pp. 73–82, Mar. 2024, doi: 10.5271/sjweh.4132.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015, pp. 91–99.

[6] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 10781–10790, doi: 10.1109/CVPR42600.2020.01079.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, arXiv:1704.04861. [Online]. Available: https://arxiv.org/abs/1704.04861.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[9] A. H. M. Rubaiyat, M. T. Reza, B. E. Mneymneh, R. Khallaf, A. Ahsan, and M. ElSiragy, "Automatic detection of helmet uses for construction safety," in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Workshops (WIW), Omaha, NE, USA, 2016, pp. 135–142, doi: 10.1109/WIW.2016.045.

[10] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on YOLOv5," in Proc. IEEE Int. Conf. Power Electron., Comput. Appl. (ICPECA), Shenyang, China, 2021, pp. 6–11, doi: 10.1109/ICPECA51329.2021.9362711.

[11] Kisaezehra, M. U. Farooq, M. A. Bhutto, and A. K. Kazi, "Real-time safety helmet detection using YOLOv5 at construction sites," Intell. Autom. Soft Comput., vol. 36, no. 1, pp. 911–927, 2023, doi: 10.32604/iasc.2023.031359.

[12] C. Shan, H. Liu, and Y. Yu, "Research on improved algorithm for helmet detection based on YOLOv5," Sci. Rep., vol. 13, no. 1, p. 18056, Oct. 2023, doi: 10.1038/s41598-023-45383-x.

[13] H. Wang, Z. Hu, Y. Guo, Z. Yang, F. Zhou, and P. Xu, "A real-time safety helmet wearing detection approach based on CSYOLOv3," Appl. Sci., vol. 10, no. 19, p. 6732, 2020, doi: 10.3390/app10196732.

[14] Y. Ji, Y. Cao, X. Cheng, and Q. Zhang, "Research on the application of helmet detection based on YOLOv4," J. Comput. Commun., vol. 10, no. 8, pp. 130–141, Aug. 2022, doi: 10.4236/jcc.2022.108009.

[15] B. P. Athidhi and P. Smitha Vas, "YOLOv7-based model for detecting safety helmet wear on construction sites," in Intelligent Sustainable Systems. ICoISS 2023, Lecture Notes in Networks and Systems, vol. 665, J. S. Raj, I. Perikos, and V. E. Balas, Eds. Singapore: Springer, 2023, pp. 363–374, doi: 10.1007/978-981-99-1726-6_29.

[16] L. Sun, H. Li, and L. Wang, "HWD-YOLO: A new vision-based helmet wearing detection method," Comput. Mater. Continua, vol. 80, no. 3, pp. 4543–4560, 2024, doi: 10.32604/cmc.2024.055115.

[17] S. S. Maharajpet, D. Mugad, and C. Nagaraj, "Advanced safety helmet detection: Enhancing industrial site safety with AI," in Convergence of Machine Learning and IoT for Enabling the Future of Intelligent Systems, Jul. 2024, pp. 1–10, doi: 10.48001/978-81-966500-7-0-1.

[18] S. Zhang, S. Huang, J. Qin, X. Li, Z. Zhang, Q. Fan, and Q. Tan, "Detection of helmet use among construction workers via helmet-head region matching and state tracking," Autom. Construct., vol. 171, p. 105987, 2025, doi: 10.1016/j.autcon.2025.105987.

[19] M. Al Rabbani Alif, "Enhancing construction site safety: A lightweight convolutional network for

effective helmet detection," Sep. 2024, arXiv:2409.12669. [Online]. Available: https://arxiv.org/abs/2409.12669. [Accessed: Apr. 11, 2025].

[20] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 8024–8035.

[21] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–125, 2000.

[22] Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. arXiv 2018, arXiv:1804.02767.

[23] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; p. 32.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.