Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3810~3820
e-ISSN: 2548-6861

Performance Analysis of YOLO, Faster R-CNN, and DETR for Automated Personal Protective Equipment Detection

Rihan Naufaldihanif [1], Dedy Kurniawan[2]*, Ken Ditha Tania [3]
[1 2 3] Sistem Informasi, Universitas Sriwijaya
rihannaufal22@email.com [1], dedykurniawan@unsri.ac.id [2], kenya.tania@gmail.com [3]

Article Info

Article history:

Received 2025-10-28
Revised 2025-12-04
Accepted 2025-12-10

Keyword:

Comparative Study,
Object Detection,
Personal Protective Equipment (PPE),
YOLO,
Faster R-CNN,
DETR.

ABSTRACT

Automated monitoring of Personal Protective Equipment (PPE) is crucial for enhancing safety in high-risk environments like construction sites, yet selecting the optimal detection model requires careful evaluation of accuracy versus efficiency trade-offs. This study presents a comparative performance analysis across distinct object detection paradigms represented by YOLO (YOLOv8, YOLOv11n), Faster R-CNN, and DETR to benchmark their suitability for real-time PPE detection. However, this study moves beyond a simple technical benchmark by also proposing a logical process to transform raw model detections (e.g., 'person', 'hardhat') into actionable compliance verification information (e.g., 'Compliant'/'Non-Compliant'). Using a curated construction site safety dataset, models were evaluated based on standard accuracy metrics (including mAP@.5:.95) and efficiency measures (inference latency). Results indicate that DETR and YOLOv11n achieved the highest overall accuracy with an identical mAP@.5:.95 of 0.770, closely followed by YOLOv8 (0.763), while the YOLO family demonstrated significantly superior real-time efficiency (6-7 ms latency). Faster R-CNN recorded a lower mAP (0.703) and the highest latency. Conclusively, YOLOv11n offers the most compelling balance for the detection phase, and the proposed logical process provides a practical method for integrating this technical output into automated safety monitoring systems.

I. INTRODUCTION

The construction sector is consistently ranked as one of the industries with the highest risk of occupational accidents, both globally and in Indonesia. This high level of risk is not merely a perception but a reality reflected in the data. According to the International Labour Organization (ILO), an estimated 2.78 million workers die annually from work-related incidents and diseases, with the construction sector being one of the primary contributors [1]. At the national level, this urgency is acutely felt; data from BPJS reports that approximately 30% of 100,000 work accident cases in Indonesia occur in this sector [1]. These figures underscore a crucial fact: behind every development project lies a significant risk to human life, which demands the strict implementation of Occupational Safety and Health (OSH) standards.

As the frontline of risk mitigation, the use of Personal Protective Equipment (PPE), such as helmets and safety vests, has become a mandatory standard that serves as a direct barrier against various field hazards [1]. The proper use of PPE has been proven to play a significant role in reducing work-related accidents [2]. Nevertheless, various studies reveal a paradox: the greatest challenge is not the availability of PPE, but rather the low level of compliance among workers in the field, which is still considered low [1]. This phenomenon is reinforced by the finding that approximately 80-85% of work accidents are caused by human factors, including the failure to use PPE [1]. This low compliance often stems from factors such as discomfort, habit, or the perception that PPE hinders productivity [1]. This condition establishes supervision as a vital element, where the lack of consistent and effective oversight directly correlates with an increase in negligence and accident risk [1, 2].

Facing the fundamental limitations of a manual supervision system, which is subjective and cannot continuously cover all areas, the utilization of modern technology becomes an inevitable evolutionary step. Computer Vision (CV) technology emerges as a promising solution, offering the capability for automated monitoring that is fast, precise, and can operate in real-time [3]. Within the domain of Computer Vision, Deep Learning (DL) based approaches have become a major breakthrough, enabling machines to learn complex patterns directly from visual data and minimizing the biases inherent in human observation [3]. The superiority of DL-based methods over conventional approaches has been proven to be significant, particularly in terms of automatic feature learning and achieving higher detection accuracy [4].

The task of localizing and classifying objects within an image is known as object detection. Modern object detection architectures can be broadly classified into several fundamental paradigms [4]. On one hand, there are two-stage detectors, such as Faster R-CNN, which prioritize high accuracy by first generating region proposals before performing classification [4]. On the other hand, one-stage detectors revolutionized the field by framing object detection as a single regression problem, enabling real-time speeds [4], [5]. The You Only Look Once (YOLO) family of models is the most popular representation of this approach, with its evolutions like YOLOv8 continuing to offer an outstanding balance of accuracy, inference speed, and a compact model size [6]. More recently, a third paradigm has emerged in the form of transformer-based architectures, such as DETR, which treats object detection as an end-to-end set prediction problem [4]

Based on a review of previous research, while many studies have validated the advantages of each paradigm separately, there is limited research that directly conducts a cross-paradigm comparative performance analysis spanning one-stage, two-stage, and transformer-based models for the specific domain of PPE compliance monitoring. This research aims to fill this gap. However, this study moves beyond a simple technical benchmark to address its application within an information system context. This approach aligns with recent advancements in the field, such as the work by Shahin et al. [19], which demonstrated the efficacy of integrating object detection algorithms to enhance safety implementation and compliance verification (5S+1). Building on this context, the primary contribution of this study is twofold: first, to comprehensively benchmark a representative model from each architectural paradigm of the YOLO family, Faster R-CNN, and DETR to identify the optimal model in terms of accuracy and efficiency. Second, this study demonstrates how the output of this optimal detector can be integrated into a logical process for automated compliance verification. This process serves to transform raw detection data (e.g., 'person', 'hardhat', 'vest') into actionable compliance information (e.g., 'Compliant' or 'Non-Compliant'), aligning the technical computer vision task with the practical needs of a safety management information system. Ultimately, this research seeks to provide a data-driven recommendation on the most effective architecture for this end-to-end monitoring and verification application.

II. Related Research

The application of Computer Vision to enhance Occupational Safety and Health (OSH) has become an active area of research, particularly in automating the monitoring of Personal Protective Equipment (PPE) usage. Various deep learning architectures have been explored for this task. For instance, research by Azizi et al. evaluated two-stage architectures like Faster R-CNN and compared them against Few-Shot Object Detection (FsDet) approaches, focusing on addressing challenges in complex construction environments such as occlusions and lighting variations [7]. In addition to two-stage architectures, one-stage detectors have been extensively studied for their potential speed. A broad comparative study by Isailovic et al. evaluated three different architectures—Faster R-CNN, MobileNetV2-SSD, and YOLOv5 for detecting 12 types of head-mounted PPE. Their results indicated that YOLOv5 delivered slightly superior performance compared to the alternatives, highlighting the effectiveness of the YOLO family of architectures for PPE detection tasks [8].

As a primary representative of the two-stage detector paradigm, Faster R-CNN is widely recognized for its ability to achieve high detection accuracy [4]. The effectiveness of this architecture has been directly validated in the relevant domain of PPE detection. A study by Azizi et al. [7] demonstrated that Faster R-CNN with a ResNet-50 backbone is capable of achieving an mAP of 73.8% and shows strong recall and precision performance for each PPE class, underscoring its role as a solid benchmark [7]. This robust performance has also been proven in other complex environments, where research by Kong et al. in fruit detection in orchards noted that a standard Faster R-CNN reached an mAP of 0.614 and an AP@0.5 of 0.917 [16]. However, the same study also critically highlights that the architecture's reliance on a CNN backbone can limit its ability to capture global context and long-range dependencies [16]. Despite this limitation, its high and proven performance across various studies makes it an essential benchmark for the two-stage approach in this comparative study.

As the field of object detection rapidly advances, comparative evaluation has become crucial for identifying the most optimal models. Research trends extend beyond merely applying existing models to actively improving them. For example, Wang et al. not only used but also modified the YOLOX architecture, comparing it with other modern models like YOLOv7 to address specific challenges such as small object detection in low-light conditions [9]. More recent comparative studies continue to validate the superiority of the latest YOLO versions. Research by Rastogi [10] and Alahdal et al. [11], in their analyses involving YOLOv8, reaffirmed that modern YOLO architectures consistently offer the best trade-off between accuracy and speed for real-time

applications. Furthermore, the latest trend has shifted towards enhancing state-of-the-art architectures, as demonstrated by Ma et al., who significantly improved the performance of YOLOv8, increasing its mAP from 0.757 to 0.814 by incorporating an attention mechanism [12].

Alongside the dominance of convolution-based architectures, recent research has explored a third, transformer-based paradigm. Although pioneering models like DETR demonstrated potential in capturing global spatial relationships, their primary drawbacks were slow training convergence and high computational overhead, limiting their viability for real-time applications [4]. To address this, architectures such as Real-Time DETR (RT-DETR) were developed to improve efficiency. A comparative study by He et al. demonstrated that in the challenging task of medical image detection, RT-DETR was able to outperform other state-of-the-art models, achieving an mAP@.5:.95 of 0.76, compared to 0.72 for YOLOv8 and 0.68 for the original DETR [14]. This advantage highlights its potential for effectively handling small and dense targets [14]. The relevance of RT-DETR as a strong baseline has also been validated in other domains, where a study by Zhao et al. for tomato detection in agricultural environments noted that the base RT-DETR model achieved a performance of 44.8% mAP@.5:.95 and 85.4% mAP@.5, which then served as a starting point for further development [15].

The existing body of research clearly establishes a performance trade-off between the accuracy-focused two-stage models and the speed-oriented one-stage models [4]. However, the recent introduction of transformer-based architectures like DETR presents a third, distinct approach whose comparative performance in the specific application domain of PPE monitoring is not yet well-documented. Consequently, a comprehensive benchmark that evaluates representatives from all three modern paradigms YOLO (one-stage), Faster R-CNN (two-stage), and DETR (transformer-based) within a unified experimental framework remains a notable gap in the literature. This research, therefore, aims to fill this gap by presenting a direct performance analysis to determine which architectural paradigm offers the optimal performance-to-efficiency ratio for automated safety monitoring applications.

III. Research and Methodology

A. Dataset

The dataset utilized in this study was curated from the public "Construction Site Safety" dataset, available on the Roboflow Universe platform. To align with the scope of this research, a manual curation process was performed. This involved data cleaning and remapping the original nine classes into five final classes relevant to this study: hardhat, vest, no-helmet, no-vest, and person. This process aimed to enhance the quality and consistency of the dataset's labels.

To accommodate the different frameworks used in this study, the curated dataset was exported in two standard annotation formats. The YOLO format, which consists of a separate text file for each image, was used for the Ultralytics-based models (YOLO and DETR). The COCO JSON format, which centralizes all annotations into a single file per data split, was used for the PyTorch-based Faster R-CNN pipeline.

In total, the final dataset consists of 5,199 images, which were randomly split into three sets: 3,656 images (70.3%) for the training set, 1,033 images (19.9%) for the validation set, and 510 images (9.8%) for the test set. The distribution of annotated instances for each class across the entire dataset is as follows: 8,276 hardhat instances, 6,129 vest instances, 524 no-helmet instances, 1,107 no-vest instances, and 3,820 person instances. These statistics reveal a significant class imbalance, where the number of instances for violation classes (no-helmet and no-vest) is substantially lower than for compliance classes (hardhat and vest). This condition presents a realistic challenge that the models must overcome.

B. Image Preprocessing and Data Augmentation

The data preparation stage included preprocessing and augmentation. All images were resized to 640x640 pixels to match the model's input dimensions. To enhance model generalization and prevent overfitting, a series of data augmentation techniques were applied on-the-fly, exclusively to the training set. These techniques included geometric transformations such as rotation, translation, scaling, and horizontal flip, alongside photometric adjustments to hue, saturation, and brightness. An identical augmentation pipeline was consistently applied to all models to ensure a fair comparison.

C. Model Architecture Selection

To conduct a cross-paradigm comparative analysis, this study selects a representative model from each of the three primary architectural approaches in modern object detection: one-stage detectors, two-stage detectors, and transformer-based detectors. Each model was chosen based on its relevance, state-of-the-art performance, and unique architectural characteristics to provide a comprehensive benchmark.

1) *One-Stage Detectors (The YOLO Family)*: The You Only Look Once (YOLO) approach, first introduced by Redmon et al., revolutionized object detection by framing it as a single regression problem, enabling real-time image processing in a single network evaluation [5]. A general YOLO architecture consists of three main components: a Backbone as a feature extractor network, a Neck to aggregate features from various scales, and a Head to perform the final predictions [14]. In this study, we evaluate the YOLOv8 and YOLOv11 variants. The YOLOv8 model

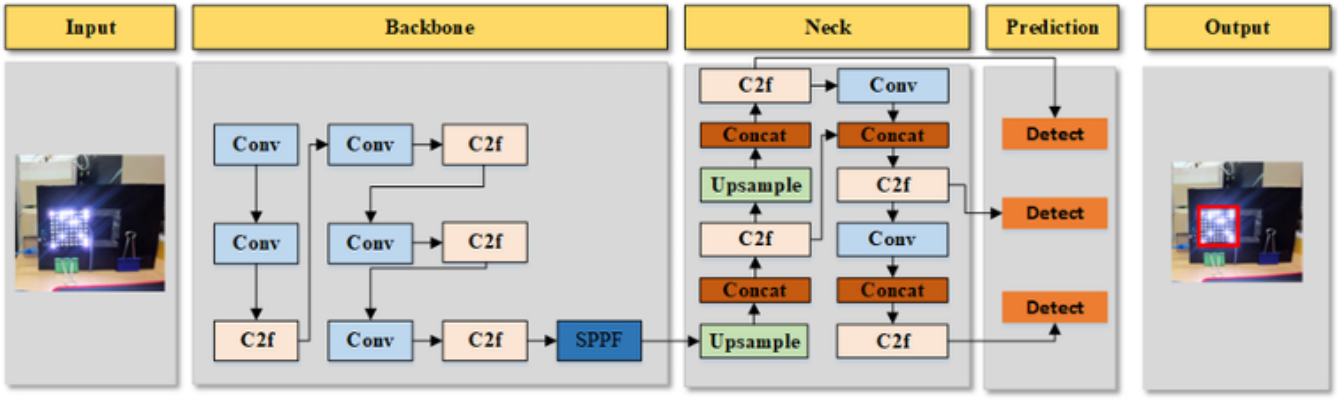

Figure 1.The architecture of Yolo.

was selected for its advanced architecture, which introduces innovations such as an anchor-free approach and the efficient C2f (Cross Stage Partial Faster) block for capturing rich gradient information [15]. Meanwhile, YOLOv11 was chosen as its counterpart, introducing new components like the more computationally efficient C3k2 block and the C2PSA attention module to enhance the model's focus on relevant image regions [14]. Both models also leverage the SPPF (Spatial Pyramid Pooling - Fast) module to aid in the detection of objects at various scales [14], [15]. The "nano" (n) variants of both models were specifically chosen to evaluate the accuracy-to-efficiency ratio, which is highly relevant for real-world implementation scenarios.

2) *Two-Stage Detector (Faster R-CNN):* As a representative of the two-stage paradigm, Faster R-CNN was chosen for its status as a fundamental architecture that prioritizes accuracy [4]. This architecture, as introduced by Ren et al., divides the detection task into two main stages executed by the following key components [17]. Backbone Network is A pre-trained convolutional neural network (CNN), such as the ResNet-50 used in this research, serves to extract feature maps from the input image [7]. Region Proposal Network (RPN) is the core innovation of Faster R-CNN. The RPN is a small convolutional network that slides over the feature maps to generate a set of object region proposals called Regions of Interest (RoI), complete with an "objectness score" [17]. RoI Head (Detector Head) For each RoI proposed by the RPN, the RoI Head extracts a fixed-size feature (using RoI Pooling or RoI Align) and then passes it through several fully-connected layers to perform two final tasks in parallel: classifying the object into a specific category and refining the bounding box coordinates for greater precision [2].

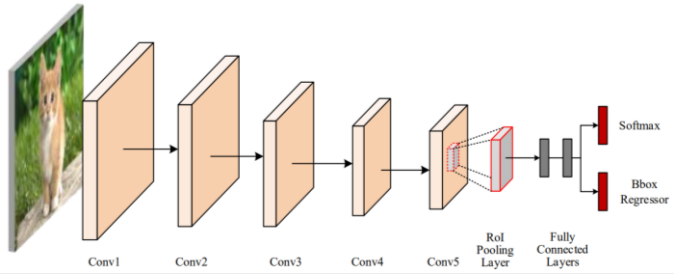

Figure 2.The architecture of Faster R-CNN.

3) *Transformer-Based Detector (RT-DETR):* Representing the latest paradigm, RT-DETR (Real-Time Detection TRansformer) was selected for evaluation. This architecture fundamentally reframes object detection as an end-to-end set prediction problem, eliminating the need for many hand-designed components like NMS [4], [18]. Its main architectural components include: CNN Backbone similar to other architectures, a CNN backbone (e.g., ResNet) is used to extract a 2D feature representation from the input image [18]. Transformer Encoder-Decoder is the core of DETR. The encoder uses a self-attention mechanism to reason about the global relationships between all parts of the image. The decoder takes a small number of object queries (learned vectors) and, by attending to the encoder's output, transforms them into representations for each object in the image. The RT-DETR variant specifically redesigns the encoder into an efficient hybrid encoder to reduce computational bottlenecks and enable high-speed inference [18]. Feed-Forward Networks (FFNs) is appended to the decoder's output, two separate FFNs predict the class and bounding box coordinates in parallel for each object representation, yielding a unique set of detections [4].

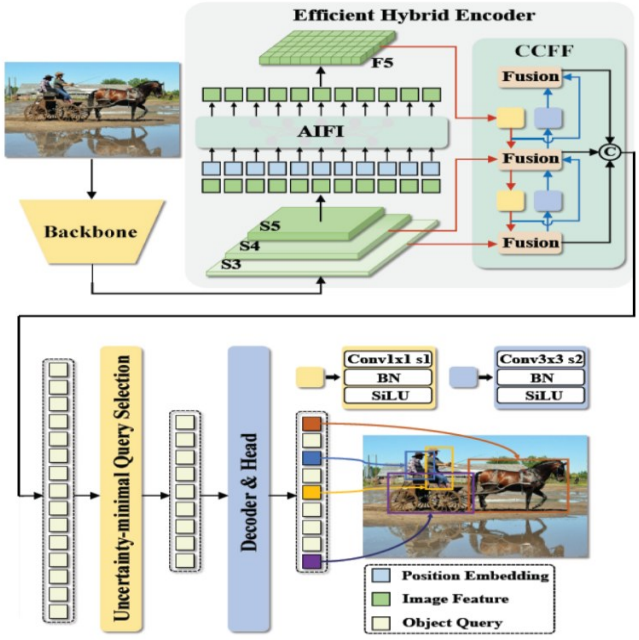

Figure 3.The architecture of RT-DETR.

D. Experimental Setup and Training

All training and evaluation experiments were utilizing T4 GPU hardware acceleration to ensure computational efficiency. The foundational framework used was PyTorch. Given the architectural and ecosystem differences, two distinct training pipelines were implemented: one utilizing the Ultralytics framework for the YOLO and DETR models, and a custom pipeline based on Torchvision for the Faster R-CNN model. To ensure fair comparison and accelerate convergence, all models utilized a Transfer Learning strategy, initialized with pre-trained weights on the COCO dataset.

The models from the YOLO family and RT-DETR were trained using the Ultralytics framework, leveraging official pre-trained models as the starting point for fine-tuning. The input image size was uniformly set to 640x640 pixels. Training was conducted for 50 epochs with a batch size of 16. The AdamW optimizer was employed with an initial learning rate of 0.01 (1e-2). To improve model generalization, a comprehensive set of data augmentations was applied, including mosaic, mixup, as well as geometric and color transformations.

Conversely, the Faster R-CNN model with a ResNet-50 backbone was trained using a custom Torchvision-based pipeline, utilizing a ResNet-50 backbone pre-trained on COCO_V1 weights. Training for this model was run for 25 epochs, taking into account the higher computational cost of the two-stage architecture and the accelerated convergence provided by the pre-trained backbone. A smaller batch size of 4 was used due to greater memory requirements. The SGD optimizer was employed with an initial learning rate of 0.005, a momentum of 0.9, and a weight decay of 0.0005. A MultiStepLR learning rate scheduler was also applied to decrease the learning rate by a factor of 10 at the 16th and 22nd epochs. Given the heavy computational load, data augmentation for this model was limited to only random horizontal flips.

TABLE I
COMPARISON OF PRIMARY TRAINING HYPERPARAMETERS

Hyperparameter	YOLO / DETR (Ultralytics)	Faster R-CNN (Custom)
Framework	PyTorch (Ultralytics)	PyTorch (Torchvision)
Epochs	50	25
Optimizer	AdamW	SGD
Initial LR	0.01	0.005
LR Scheduler	Cosine Learning Rate Scheduler	MultiStepLR
Batch Size	16	4
Input Size	640x640	640x640
Augmentation	Comprehensive (Geometric and Color)	Comprehensive (Geometric and Color)

E. Evaluation Metrics

To quantitatively evaluate and compare the performance of the models, a series of standard metrics for object detection tasks were used. These metrics are:

1) *Precision:* Measures the ratio of correct positive predictions to the total number of positive predictions made by the model. High precision indicates a low false positive rate.

$$P = \frac{TP}{TP + FP}$$

2) *Recall:* Measures the ratio of correct positive predictions to the total number of actual positive instances in the data. High recall indicates a low false negative rate.

$$R = \frac{TP}{TP + FN}$$

3) *mAP@0.5:* Mean Average Precision (mAP@0.5) calculated at an Intersection over Union (IoU) threshold of 0.5. This metric assesses the model's general object localization capability.

$$P = \frac{1}{N}\sum_{i=1}^{N} AP_1$$

4) *mAP@0.5:0.95:* Mean Average Precision (mAP@0.5:0.95) calculated by averaging the mAP across ten different IoU thresholds, from 0.5 to 0.95 with an interval of 0.05. This metric provides a stricter and more comprehensive assessment of the bounding box localization precision.

$$P = \frac{1}{10}\sum_{i=0}^{9} \left(\frac{1}{N}\sum_{i=1}^{N} AP_1\right)$$

5) *Inference Speed:* Measured in Frames Per Second (FPS) on the test set. FPS indicates how many images the

model can process per second, serving as a direct indicator of its real-time capabilities.

6) Number of Parameters: The total trainable parameters in the model, typically expressed in millions (M).

7) GFLOPs (Giga Floating-Point Operations): Measures the computational load required for the model to process a single image. A lower value indicates a more computationally lightweight model.

The variables are defined as follows, where

P is the precision.
R is the recall. N i=1 APi
TP denotes the true positives (correctly predicted positive instances).
FP denotes the false positives (incorrectly predicted positive instances).
FN denotes the false negatives (incorrectly predicted negative instances).
N is the number of classes.
AP1(mAP@50) represents the average precision for class i at IoU = 0.5.
AP1(mmAP@95) represents the average precision for class i at IoU = 0.5 + 0.05k; k is the index representing each threshold from 0.5 to 0.95 (0.5, 0.55, 0.6, ..., 0.95).

F. Process for Automated Compliance Verification

To bridge the gap between the technical benchmark of object detection models and their practical application within an Information Systems context, this research proposes a logical post-processing process. The objective of this process , as illustrated in Figure 4, is to transform the raw data output from the detector (i.e., a list of bounding boxes and class labels) into actionable compliance information. This approach, which separates object detection from compliance verification, allows the system to not only "see" objects but also to automatically "understand" a worker's compliance status.

The process applies a set of spatial-based logical association rules to determine the compliance status of each detected person. The process begins by identifying and iterating through all person objects within a frame. For each person, the system then scans for any hardhat and vest detections (PPE) or no-helmet and no-vest detections (violations) that are spatially overlapping with the person's bounding box, based on a predefined Intersection over Union (IoU) threshold. Based on these associations, compliance rules are applied to generate status information. A 'NON-COMPLIANT' status is assigned if the person is associated with a no-helmet or no-vest detection, or if the person fails to be associated with either a hardhat or a vest. The 'COMPLIANT' status is only assigned if the person is associated with both a hardhat AND a vest, and no violation detections are associated.

The output of this process is no longer just bounding box data, but rather discrete compliance information (e.g., 'WORKER-1: COMPLIANT', 'WORKER-2: NON-COMPLIANT'). This information is ready to be consumed by a higher-level Safety Management Information System for reporting, analytical dashboard creation, or triggering real-time alerts.

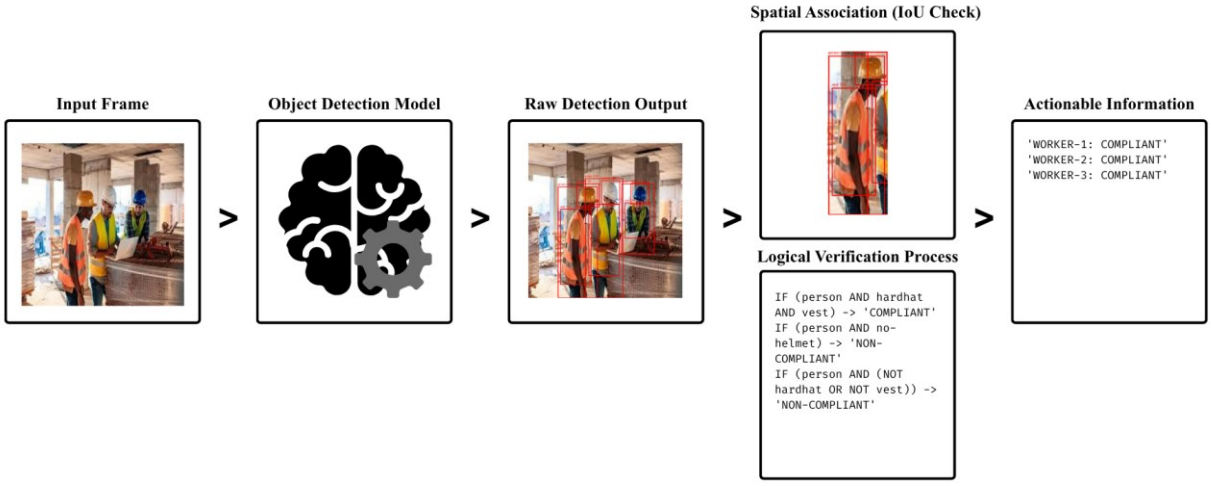

Figure 4.Flow diagram of the automated compliance verification process.

IV. RESULT

A. Overall Performance Benchmark

To obtain a general overview of the performance across different architectural paradigms, a comprehensive benchmark was conducted on the test set. Table 1 presents a thorough comparison of each evaluated model, covering key accuracy and computational efficiency metrics. This analysis aims to identify the fundamental trade-offs between detection accuracy and feasibility for real-time implementation.

In Table II, the YOLOv11n and DETR models demonstrated identical top performance in mAP@.5:.95 (0.770), indicating the highest overall detection capability. YOLOv8 followed closely (0.763). The performance of Faster R-CNN showed a significant improvement compared to previous evaluations, reaching an mAP@.5:.95 of 0.703. Interestingly, on the mAP@.5 metric, Faster R-CNN is now tied with YOLOv11n for the best score (0.939), suggesting its general localization ability is excellent at looser IoU thresholds. However, on the Recall (MAR) metric, the DETR architecture remains significantly superior (0.932), while Faster R-CNN (0.770) ranks lowest, indicating this model may miss more objects compared to the others.

The efficiency analysis continues to show very significant differences between paradigms. The YOLO family of models (YOLOv8 and YOLOv11n) remain the most efficient. YOLOv11n is the most lightweight model (2.58 M parameters, 6.3 GFLOPs) and is extremely fast (6.8 ms latency). DETR is considerably heavier (32.0 M parameters, 103.5 GFLOPs) and roughly 6 times slower than YOLO(41.7 ms latency). Faster R-CNN remains the most computationally demanding architecture (41.32 M parameters, 536.0 GFLOPs) with the highest inference latency (96.6 ms), approximately 14 times slower than YOLOv11n.

TABLE II
OVERALL ACCURACY AND EFFICIENCY BENCHMARK

Architecture	mAP@.5:.95	mAP@.5	Recall	Parameters (M)	GFLOPs	Latency (ms/img)
YOLOv8	0.763	0.938	0.857	3.01	8.1	6.2
YOLOv11n	0.770	0.939	0.902	2.58	6.3	6.8
DETR	0.770	0.934	0.932	32.0	103.5	41.7
Faster R-CNN	0.703	0.939	0.770	41.32	536.0	96.9

Based on this updated overall benchmark, the trade-off between accuracy and efficiency becomes clearer. Although Faster R-CNN shows improved accuracy, making it more competitive, especially at mAP@.5, its computational cost and latency remain major obstacles for real-time applications. DETR, despite its high accuracy and best recall, also exhibits relatively high latency. Thus, the previous conclusion still holds: modern one-stage architectures, particularly YOLOv11n, continue to offer the most compelling balance between very high detection accuracy (tied for the highest mAP with DETR) and superior computational efficiency (lowest parameters, GFLOPs, and latency), making it the most practical choice for field implementation.

B. Detailed Per-Class Performance Analysis

To understand the specific strengths and weaknesses of each architecture, a performance analysis of mAP@.5:.95 was conducted for each target class. Table 2 presents this comparison, which is crucial for evaluating the models' ability to handle challenges such as occlusion and class imbalance.

The quantitative analysis per class in Table III reveals more nuanced performance trends. For common classes with numerous instances like hardhat and vest, the transformer-based architecture, DETR, consistently achieved the highest mAP. This is likely due to its global self-attention mechanism, enabling it to better reason about contextual relationships between objects in dense or occluded scenes. For the person class, YOLOv11n showed a slight advantage.

Performance on the violation classes also provided key insights. For the no-helmet class, all models demonstrated highly competitive and strong performance (though YOLOv11n leads slightly), indicating reliable detection for this category across architectures. However, the analysis for the minority class no-vest yields particularly interesting results. YOLOv11n now clearly emerges as the top performer for this class (0.890 mAP), followed closely by YOLOv8 and DETR. Contrary to initial expectations for two-stage architectures, Faster R-CNN exhibited the weakest performance on this less frequent class (0.845 mAP). This finding suggests that while the two-stage approach can be meticulous, it does not inherently guarantee superior performance on minority classes in this specific application, and highly optimized one-stage models like YOLOv11n demonstrate remarkable effectiveness even for challenging, less common objects.age models like YOLOv11n demonstrate remarkable effectiveness even for challenging, less common objects.

TABLE III
PER-CLASS PERFORMANCE (MAP@.5:.95)

Class	YOLOv8	YOLOv11	DETR	Faster R-CNN
hardhat	0.724	0.731	0.744	0.703
vest	0.706	0.709	0.730	0.646
no-helmet	0.828	0.829	0.821	0.788
no-vest	0.883	0.890	0.885	0.845
person	0.674	0.691	0.672	0.607

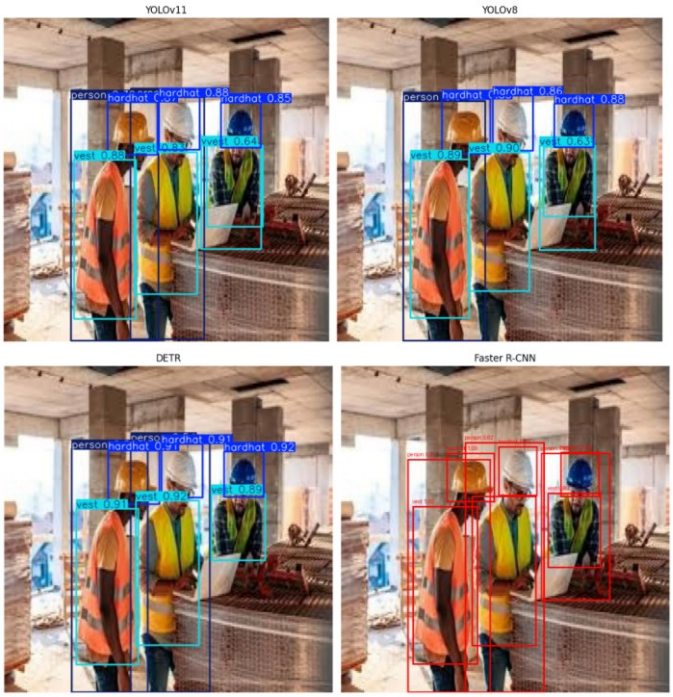

Figure 5.Visual comparison of detection results on a test image featuring workers in close proximity with partial occlusion.

To complement the quantitative analysis and provide deeper insight into model behavior in real-world scenarios, a visual comparison of detection results was performed on representative test cases (Figures 5-7). Figure 5 depicts a scene with three workers in close proximity. While all models detected the primary PPE (hardhat and vest) well, Faster R-CNN uniquely identified all three person instances, whereas YOLOv11n/DETR detected two, and YOLOv8 detected only one. Interestingly, while this visual inspection might suggest strong recall for Faster R-CNN in specific scenes, the overall quantitative Mean Average Recall (MAR) score in Table 1 places it as the lowest performer (0.770). This discrepancy might indicate that while capable in certain scenarios, Faster R-CNN's recall performance may be less consistent across the entire test set or potentially more sensitive to higher IoU thresholds compared to the other architectures.

Figure 6.Comparative detection results on a challenging test image with numerous workers viewed from a distance, resulting in smaller object scales.

Figure 6 presents a more challenging scenario with numerous workers viewed from a distance. Here, YOLOv11, YOLOv8, DETR, and Faster R-CNN exhibited remarkably similar and robust performance on the smaller objects, producing relatively clean outputs.

Figure 7.Detection results comparison on an indoor test image depicting an individual without required PPE (no-helmet, no-vest), with PPE items present in the background.

Finally, Figure 7 shows an indoor scene where all models correctly detected person and the no-vest status. Notably, YOLOv8 also identified a partially visible vest located in the background on a table, showcasing a higher sensitivity to background objects compared to the others in this instance. Collectively, this qualitative analysis visually corroborates many of the quantitative findings, highlighting the specific strengths (e.g., DETR's consistency in dense scenes, YOLO's balanced efficiency) and potential weaknesses or inconsistencies (e.g., Faster R-CNN's lower average recall

despite visual successes) of each architectural paradigm when faced with practical detection challenges.

C. *Accuracy vs. Efficiency Trade-off*

To visually synthesize the relationship between detection accuracy and computational efficiency, the performance of each evaluated architecture is plotted in Figure 5. This scatter plot positions each model based on its primary accuracy metric (mAP@.5:.95) against its inference latency (ms/image), providing a clear illustration of the inherent trade-offs involved in selecting an object detection paradigm for real-time applications. The vertical axis represents accuracy, where higher values indicate better performance, while the horizontal axis represents latency, where lower values (further to the left) indicate greater speed and efficiency. Consequently, the upper-left quadrant of the plot represents the most desirable performance zone, signifying high accuracy coupled with low latency.

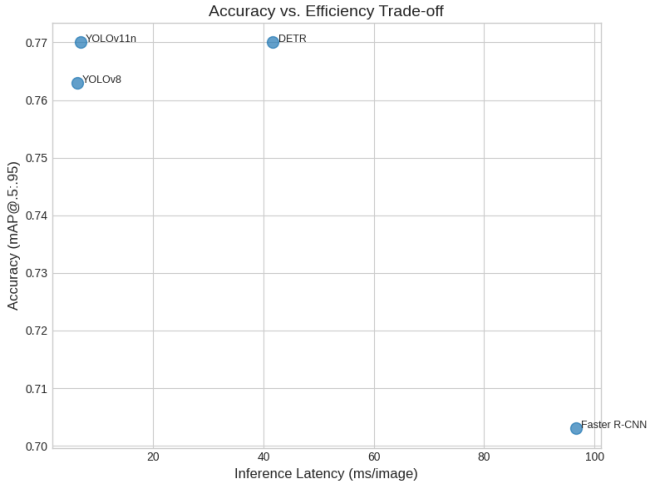

Figure 8.Accuracy vs. Efficiency trade-off comparison for the evaluated object detection models on the test set.

Figure 8 clearly highlights the distinct performance profiles of the evaluated architectures. The YOLO family models (YOLOv8 and YOLOv11n) occupy the highly desirable upper-left quadrant, demonstrating both state-of-the-art accuracy and exceptional speed (latency under 7 ms). This firmly positions them as the leading candidates for real-time deployment scenarios where both high performance and rapid inference are critical. In contrast, the transformer-based DETR model, while achieving top-tier accuracy comparable to YOLOv11n, resides significantly further to the right on the plot due to its substantially higher latency (41.7 ms). Similarly, Faster R-CNN, representing the two-stage paradigm, is located in the lower-right quadrant, indicating both lower accuracy (in terms of mAP@.5:.95) and the highest inference time (96.6 ms).

The visualization underscores the computational return on investment offered by each approach. The highly optimized one-stage detectors from the YOLO family provide the best balance, delivering near-maximal accuracy at a minimal computational cost. While DETR showcases the potential of transformer architectures to achieve high accuracy without hand-crafted components, its current real-time efficiency lags considerably behind CNN-based counterparts. Faster R-CNN, despite its foundational importance and improved performance compared to earlier evaluations, demonstrates lower efficiency relative to the more modern architectures for this specific task and hardware configuration. Therefore, based explicitly on the accuracy versus efficiency trade-off depicted, the YOLO models, particularly YOLOv11n, present the most practical and effective solution evaluated in this study for real-time PPE detection.

D. *Discussion of Findings*

Synthesizing the experimental results from the preceding sections provides a comprehensive assessment of the evaluated object detection architectures. The primary research question focused on analyzing the performance trade-off between accuracy and efficiency across different paradigms for automated PPE detection. The overall benchmark (Section 4.1) indicated that YOLOv11n and DETR achieved the highest overall accuracy (mAP@.5:.95), while the YOLO family demonstrated vastly superior computational efficiency in terms of parameters, GFLOPs, and inference latency. DETR exhibited the highest recall (MAR), whereas Faster R-CNN, despite competitive accuracy in some metrics, was the least efficient. Based explicitly on the accuracy-efficiency trade-off visualized in Figure 8, YOLOv11n emerges as offering the most compelling balance and is thus recommended as the primary architecture for real-time applications among those tested.

A deeper analysis of per-class performance (Section 4.2) and qualitative results (Figures 5-7) revealed further nuances. DETR's strength in achieving the highest mAP for common classes like hardhat and vest, coupled with its top overall recall, aligns with the theoretical advantages of its global self-attention mechanism, likely enabling better handling of occlusion in dense scenes (visually suggested in Figure 2). However, this capability comes at the cost of significantly higher latency. Analysis of the minority class no-vest yielded particularly interesting results following the data update. YOLOv11n clearly emerged as the top performer for this class based on quantitative mAP scores. Faster R-CNN, conversely, displayed inconsistent behavior; while capable of correct detection in certain visual instances (Figure 3), its average mAP score (0.845) ranked the lowest for this class, suggesting potential instability or sensitivity to specific conditions. This contrasts with the expectation that two-stage detectors might excel at minority classes. Furthermore, the observation that YOLOv8 uniquely detected a background vest (Figure 7) might indicate higher sensitivity, potentially beneficial but also hinting at a risk of false positives requiring careful threshold calibration. The discrepancy noted between Faster R-CNN's visual recall in some scenes (Figure 5) and its low overall MAR score (0.770) further underscores that its performance may be less consistent across the entire test set compared to other architectures.

These findings carry direct practical implications for implementing automated PPE monitoring systems. For real-time applications deployed on hardware similar to the test environment (NVIDIA T4 GPU), the YOLO family (specifically YOLOv11n or YOLOv8) represents the most pragmatic choice due to its balance of speed and accuracy. DETR could be considered if maximizing recall is the absolute priority and its higher latency (~40 ms) is tolerable. While Faster R-CNN demonstrated improved accuracy with added augmentations, its significant computational requirements and latency render it less suitable for real-time deployment unless specific legacy constraints or unique niche performance characteristics necessitate a two-stage approach.

Beyond the model-specific findings, the introduction of the post-processing process (Section 3.F) directly addresses the practical application gap between technical detection and functional compliance verification. This two-stage approach, which separates object detection from compliance logic, is critical for building a complete safety monitoring system. This methodology aligns conceptually with validated approaches in other research, such as the work by Shahin et al. In their study, they also employed a distinct two-stage process: (1) object detection using YOLOv7, followed by (2) compliance verification using a secondary machine learning or deep learning classifier (like VGG-16 or ResNet-50).[19].

The advantage of the spatial-based logical process proposed in our study, however, lies in its high computational efficiency. By not requiring a secondary, trainable classifier (unlike Approach I or Approach III in the Shahin et al. study), our entire pipeline (e.g., YOLOv11n combined with the logical process) remains exceptionally lightweight. This characteristic makes it an ideal candidate for real-time field implementation, particularly for the resource-constrained edge computing (IoT) scenarios that were an initial focus. This process effectively transforms the technical detector from a simple model into a functional safety management tool, capable of delivering the "actionable compliance information" promised in the introduction.

Finally, it is essential to acknowledge the limitations of this study. Firstly, while efforts were made to align basic data augmentation strategies, the Faster R-CNN pipeline did not include advanced techniques like Mosaic or MixUp used by the Ultralytics models; this residual difference could remain a confounding factor. Secondly, all performance evaluations, particularly latency measurements, were conducted on specific hardware (NVIDIA T4 GPU); results may vary considerably on different hardware platforms. Thirdly, the dataset, although curated, possesses inherent limitations regarding the diversity of construction environments and potential annotation biases. Lastly, this study focused on specific variants of each model family (e.g., YOLO 'n' versions, Faster R-CNN with ResNet-50); performance could differ with larger model variants or different backbones. Future work could address these limitations by exploring hardware-specific optimizations, expanding dataset diversity, and evaluating a wider range of model scales.

V. CONCLUSION

This research presented a comprehensive comparative performance analysis of three distinct object detection paradigms (one-stage, two-stage, and transformer-based) for automated Personal Protective Equipment (PPE) detection. Furthermore, it introduced a practical post-processing process to translate these technical detection results into actionable compliance information. The primary objective was to establish a clear benchmark for model selection while also demonstrating a practical pathway for integrating these models into real-world safety monitoring applications.

The experimental findings revealed significant insights into the capabilities of each architecture. While YOLOv11n and DETR achieved the highest overall accuracy in terms of mAP@.5:.95, the YOLO family consistently demonstrated superior efficiency, exhibiting the lowest parameter counts, GFLOPs, and inference latencies, making them ideal for real-time processing. DETR, leveraging its transformer architecture, attained the highest overall recall (MAR), showcasing excellent capability in identifying existing objects, but this came at the cost of substantially higher latency compared to YOLO models. Faster R-CNN, representing the two-stage approach, delivered competitive accuracy, particularly at the mAP@.5 threshold, but ultimately ranked lowest in mAP@.5:.95 and recall, while incurring the highest computational cost and slowest inference speed. Furthermore, its performance, especially on the minority class no-vest, showed inconsistencies between quantitative metrics and specific visual examples.

Based on the thorough evaluation, this study concludes that modern one-stage detectors, specifically YOLOv11n, offer the most compelling balance for the detection phase of an automated, real-time PPE monitoring system. More importantly, when combined with the proposed logical verification process (Section 3.F), this approach provides a complete, efficient, and practical end-to-end solution. This combined pipeline successfully pairs state-of-the-art detection accuracy with exceptional computational efficiency. The findings therefore provide not only a valuable data-driven benchmark for model selection, but also a practical implementation pathway for deploying robust and timely safety monitoring systems in construction environments and similar domains.

However, it is important to acknowledge the limitations of this work. The performance evaluation, particularly latency, was conducted on specific hardware (NVIDIA T4 GPU), and results may differ on other platforms, especially resource-constrained edge devices. Minor residual differences in data augmentation strategies between the training pipelines could also have influenced the direct comparison. Furthermore, the dataset, while curated, may not encompass the full spectrum of environmental variations found in all construction sites. Future research should focus on validating these findings on diverse hardware, including edge computing platforms (IoT devices). Exploring the performance of larger model variants, investigating hardware-specific optimizations for non-YOLO

architectures, and further expanding the dataset's diversity are also recommended avenues to enhance the robustness and generalizability of automated PPE detection systems. Additionally, future work should investigate the integration of object tracking algorithms (example DeepSORT) to ensure temporal consistency in continuous video streams and further optimize the rule-based compliance logic to handle complex, dynamic field scenarios.

REFERENCES

[1] A. C. P. Nusantara, Andriyani, and T. Srisantyorini, "Kepatuhan Penggunaan Alat Pelindung Diri (APD) pada Pekerja Kontruksi: Kajian Literatur tentang Pengaruh Faktor Individu dan Pendekatan Keselamatan Kerja," Jurnal Riset Ilmu Kesehatan Umum, vol. 3, no. 2, pp. 135-146, Apr. 2025.

[2] Fitriadi, Muzakir, A. Saputra, S. A. Lestari, K. Hadi, H. Noviar, and Sudarman, "Peningkatan Keselamatan Kerja Di Industri Galangan Kapal Tradisional Melalui Edukasi Dan Implementasi Standar K3," Jurnal Pengabdian Kolaborasi dan Inovasi IPTEKS, vol. 3, no. 1, pp. 26-39, Feb. 2025.

[3] A. Upadhyay et al., "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," Artificial Intelligence Review, vol. 58, p. 92, Jan. 2025, doi: 10.1007/s10462-024-11100-x.

[4] E. Edozie, A. N. Shuaibu, U. K. John, and B. O. Sadiq, "Comprehensive review of recent developments in visual object detection based on deep learning," Artificial Intelligence Review, vol. 58, p. 277, Jun. 2025, doi: 10.1007/s10462-025-11284-w.

[5] 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.

[6] W. A. Shobaki and M. Milanova, "A Comparative Study of YOLO, SSD, Faster R-CNN, and More for Optimized Eye-Gaze Writing," Sci, vol. 7, no. 2, p. 47, Apr. 2025, doi: 10.3390/sci7020047.

[7] R. Azizi, M. Koskinopoulou, and Y. Petillot, "Comparison of Machine Learning Approaches for Robust and Timely Detection of PPE in Construction Sites," Robotics, vol. 13, no. 2, Art. no. 31, Feb. 2024.

[8] V. Isailovic, A. Peulic, M. Djapan, M. Savkovic, and A. M. Vukicevic, "The compliance of head-mounted industrial PPE by using deep learning object detectors," Scientific Reports, vol. 12, no. 1, Art. no. 16347, Sep. 2022.

[9] Z. Wang, Z. Cai, and Y. Wu, "An improved YOLOX approach for low-light and small object detection: PPE on tunnel construction sites," Journal of Computational Design and Engineering, vol. 10, no. 3, pp. 1158–1175, May 2023.

[10] S. Rastogi, "Traffic Congestion Reduction through Real-time Object Detection: Analyzing the Effectiveness of different CNN models such as Mask RCNN, SSDNet and Yolo," M.Sc. Research Project, National College of Ireland, 2024.

[11] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif, "Real-time Object Detection in Autonomous Vehicles with YOLO," Procedia Computer Science, vol. 246, pp. 2792–2801, 2024.

[12] B. Ma et al., "Distracted Driving Behavior and Driver's Emotion Detection Based on Improved YOLOv8 With Attention Mechanism," IEEE Access, vol. 12, pp. 37983–37994, 2024.

[13] P. Hidayatullah, N. Syakran, M. R. Sholahuddin, T. Gelar, and R. Tubagus, "YOLOV8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review," Preprint, Jan. 2025.

[14] W. He, Y. Zhang, T. Xu, T. An, Y. Liang, and B. Zhang, "Object Detection for Medical Image Analysis: Insights from the RT-DETR Model," in Proc. 2025 Int. Conf. Artif. Intell. Comput. Intell. (AICI), Kuala Lumpur, Malaysia, 2025, pp. 415-420.

[15] Z. Zhao et al., "RT-DETR-Tomato: Tomato Target Detection Algorithm Based on Improved RT-DETR for Agricultural Safety Production," Appl. Sci., vol. 14, no. 14, Art. no. 6287, Jul. 2024.

[16] X. Kong, X. Li, X. Zhu, Z. Guo, and L. Zeng, "Detection model based on improved faster-RCNN in apple orchard environment," Intell. Syst. Appl., vol. 21, Art. no. 200325, Jan. 2024.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, Jun. 2017.

[18] Y. Zhao et al., "DETRs Beat YOLOs on Real-time Object Detection," arXiv preprint arXiv:2304.08069, 2023.

[19] M. Shahin, F. F. Chen, A. Hosseinzadeh, H. Khodadadi Koodiani, H. Bouzary, and A. Shahin, "Enhanced safety implementation in 5S + 1 via object detection algorithms," *International Journal of Advanced Manufacturing Technology*, vol. 125, no. 7–8, pp. 3701–3721, Apr. 2023, doi: 10.1007/s00170-023-10970-9.