Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 03 (2026)

Deep Learning Framework for Automated Worker Helmet Detection And Safety Compliance Monitoring Using The Yolo Object Detection Model

[1]D. Pitchaiah, [2]Dr.K. Mohana Rao

M.Tech Student, Department of CSE Prakasam Engineering College (Autonomous), Kandukur, India

Professor, Department of CSEPrakasam Engineering College (Autonomous), Kandukur, India

Abstract- Construction automation tools that implement computer vision technologies like "YOLO and Re-CNNs actively contribute to enhancing safety at construction sites with their automated monitoring and detection systems. The system presented in this paper harnesses the YOLO detection model, and utilizes the Flask web application to detect in real time if construction workers are wearing safety helmets, and to check if construction safety compliance is being upheld. This framework offers continuous automated monitoring of construction safety compliance in lieu of the traditional manual check systems, and therefore reduces both systematic and human errors. The system incorporates an automated notification system to alert construction safety compliance officers to the occurrence of a safety compliance violation in order to facilitate on the spot corrective actions. The Flask web application offers an easy-to-use Upload interface for video files and real time web cam control with no technical skills required of the user. The system is designed to record safety compliance violations for future reviews and to facilitate safety compliance violation reporting. The built-in YOLO detection model reportedly offers powerful detection capabilities in a variety of different environments, in addition to being able to handle changes in visibility and abrupt occlusions. The combination of deep learning, computer vision, and web technologies offers a highly effective system to ensure compliance with construction safety regulations.

Keywords- Deep Learning, YOLO, Helmet Detection, Computer Vision, Safety Compliance, Object Detection, Flask, Real-Time Monitoring

I.Introduction

The increased industrialization and significant growth in construction and infrastructure have necessitated the development of effective safety systems, particularly for the construction, manufacturing, and mining industries. Safety of employees is of utmost importance. Inadequate protective gear such as safety helmets can lead to injury and even death. Safety systems traditionally rely on the safety officer to monitor compliance, which is inefficient and labor intensive, and can be made worse by oversight and control mistakes. Thus, there is demand for intelligent and automated safety systems that minimize safety risks and maintain safety compliance. Recent advancements in deep learning and computer vision have automated safety systems. Convolutional Neural Network (CNN) has shown great promise and has achieved state of the art results in various image classification and object detection tasks. Among various object detection research and applications frameworks, YOLO (You Only Look Once) stands out to a great extent because of its ability to conduct fast, accurate and real time detection [1], [9]. It is highly suitable to Industrial environments because, unlike traditional multi-stage detectors, YOLO is a single pass framework.

The proposed framework uses a YOLO-inspired deep learning model to analyze streams and images in real-time to detect workers and their compliance to safety requirements. It allows for automated monitoring to eliminate the requirement for manual oversight. The framework is intended to maximize the effectiveness of operations while minimizing the need for manual checks. The model uses annotated images of workers with and without helmets as training datasets to maximize the model's reliability to detect safety compliance within varying external conditions, including varying degrees of light and when worker safety gear is obstructed by other objects. The framework employs alerts to automatically notify supervisors of safety compliance issues, thus, allowing for safety compliance issues to be immediately addressed. The framework combines automated safety compliance monitoring with cost and scale effectiveness to improve provider safety and safety compliance within legal requirements.

Ii. Literature Review

In recent years, a lot of attention has been given to the incorporation of deep learning techniques for safety monitoring. Earlier techniques for object detection, e.g. Faster R-CNN, used region-based approaches to object localization, but these approaches had high computation costs and were therefore not applicable for real-time detection [2]. Similarly, Mask R-CNN extended the capability of object detection for instance-based segmentation, but it was extremely costly [3].

With the evolution of YOLO, the possibility of real-time object detection was introduced due to its single-stage detection approach [1], [9]. All the iterations of YOLO, YOLOv2 and YOLOv3, are now commonly used in industrial settings due to their increased accuracy and speed of detection [10]. The later versions, YOLOv4, YOLOv5 and YOLOv7, also focused on the same aspects and were able to achieve even greater results [11]-[13].

A few researchers have looked into helmet detection as part of worker safety. Deep learning-based safety helmet detection for construction sites is proposed by Zhang et al. [4]. Techniques based on traditional methods have lower accuracy. Helmet detection using CNN was proposed by Chandran and Narayanan [5]. This shows how deep learning can help in the process of safety compliance. Huang et al. [6] showed that deep learning can help develop a system that detects a violation of proper helmet usage.

There are other studies that explore the YOLO model and the real-time detection of safety helmets. Real-time safety monitoring systems utilizing YOLOv3 were developed by Islam et al. [8], and real-time detection was achieved. Nath et al. [15] and Nath et al. [16] showed that safety helmet detection systems based on deep learning significantly increased safety in work environments by reducing the need for manual inspections. Rahman et al. [17] validated the importance of real-time detection systems in actual work environments. The proposed system uses an optimized YOLO-based detection system with a web-based monitoring system that provides real-time detection and an automated notification system as a response to the issues presented.

Table 1: Summary of Existing Helmet Detection Approaches

Study Focus	**Technique Used**	**Key Contribution**
Object Detection	Faster R-CNN	High accuracy but slower performance
Instance Segmentation	Mask R-CNN	Pixel-level detection with higher complexity
Real-Time Detection	YOLO (v1–v3)	Fast and efficient object detection
Industrial Safety Monitoring	YOLOv4 / YOLOv5	Improved accuracy and real-time performance

Safety Compliance Systems	CNN-based Models	Automated helmet detection in workplaces

Iii. Existing System

Safety for workers in industries and construction sites has relied on manual supervision as well as traditional monitoring systems. Such systems require safety personnel to check if workers are donning safety equipment. Though, to some extent, manual monitoring is effective, it is labor intensive, time consuming, and prone to human errors especially in large and diversified work environments. With computer vision technologies, automation started with traditional image processing techniques. These methods required hand crafted techniques to recognize colors and shapes, and to detect edges. Such methods were not robust and, for the most part, failed to detect safety equipment in the presence of illumination changes, background noise, and occlusion. Therefore, these methods were not effective for monitoring workers in practical situations.

Detection of safety equipment in machine learning methods relies on hand crafted techniques to extract features that are classified using SVM, decision trees, and similar techniques. These methods are an improvement over traditional methods, but still rely on feature engineering and are not effective in different environments. Deep learning models brought about the largest shift in this domain, and with the advent of region-based convolutional Neural Networks, the field saw a large shift in object detection technology. Though improved localization and accuracy were the outcomes of these models, the complex nature of these models and their multi-stage workflows resulted in increased computation and inference time that made them ill suited for real-time safety monitoring systems.

Recent innovations in single-stage detectors like YOLO have improved upon traditional methods showing that accurate and instantaneous detection are not mutually exclusive. Although these methods have come a long way, a lot of established systems still struggle with limited scalability, the lack of an instantaneous alert system, and a lack of easy to design/dispose of functional interfaces for operational purposes.

Table 2: Comparison of Existing Helmet Detection Approaches

Approach	**Techniques Used**	**Advantages**	**Limitations**
Manual Monitoring	Human supervision	Simple implementation	Time-consuming, error-prone
Traditional Image Processing	Edge, color, shape detection	Low computational cost	Sensitive to lighting and noise
Machine Learning	SVM, Decision Trees	Improved accuracy	Requires manual feature extraction
Region-Based Deep Learning	Faster R-CNN, Mask R-CNN	High accuracy and localization	High computational complexity, slow
YOLO-	Single-stage	Real-time	Requires

Based Detection	object detection	performance, efficient	large training dataset

Iv. Proposed Methodology

The proposed system employs the YOLO object detection model to create a deep learning-based automated helmet detection system that assesses safety of workers" in real time using video streams from webcams or uploaded videos. The system segments person, head, and helmet. The system allows for real time safety monitoring. Safety violations are triggered when a head is detected with no helmet across a series of video frames. The system will issue an alert and document the incident in a frame with an image capture support.

A. System Overview

Based on safety monitoring, the proposed framework operates through the analysis of video streams (webcam or uploaded) by frames. Each frame is "analyzed through a trained YOLO model. The model detects and classifies person, head, and helmet. The system determines compliance and issues alerts based on the detection of safety violations.

B. Mathematical Representation

The YOLO-based detection can be mathematically expressed as:

$$S \times S \times (B \times 5 + C)$$

where:

- S = Grid size
- B = Number of bounding boxes per grid
- C = Number of classes

Each bounding box prediction consists of:

(x,y,w,h,confidence)

The confidence score is defined as:

$$\text{Confidence} = P(\text{Object}) \times IOU_{pred}^{truth}$$

Detection through YOLO is a process in which the presence of an object and its boundaries are determined in a single process.

C. Step-wise Working Procedure

1. **Input Acquisition**
- Real-time videos can be recorded or pre-recorded videos can be uploaded.

2. **Frame Extraction**
- Use OpenCV to extract frames from the videos.

3. **Object Detection (YOLO Model)**
- Use the trained YOLO model to detect head, helmet and person.

4. **Classification**
- Classify detected objects to having helmet and no helmet.

5. **Confidence Evaluation**

- Retain detected objects with confidence greater than 0.5.

6. **Violation Detection Logic**

- Detect violation if head is detected without helmet for N number of frames.

7. **Alert Generation**

- Sound an alarm and snapshot the violation.

8. **Visualization**

- Draw bounding boxes around the detected objects and label them.

9. **Streaming Output**

- Stream the processed frames to the web interface built with Flask in real-time.

D. Algorithm: Helmet Detection and Alert System

Input: Video stream or uploaded video
Output: Detection results with alerts

1. Read video input
2. Extract frames sequentially
3. Apply YOLO model for object detection
4. Identify head and helmet objects
5. If head detected without helmet:
   - Increment violation counter
6. If counter exceeds threshold:
   - Trigger alarm
   - Save violation frame
7. Display annotated frame
8. Repeat until video ends

E. Proposed System Features

Table 3: Proposed System Features

Feature	Description
Real-Time Detection	Detects helmet usage instantly from video streams
High Accuracy	Uses YOLO deep learning model for precise detection
Automated Alerts	Generates alarms for safety violations
Scalable Architecture	Supports multiple video inputs
Web Integration	Flask-based interface for easy access

V.System Architecture

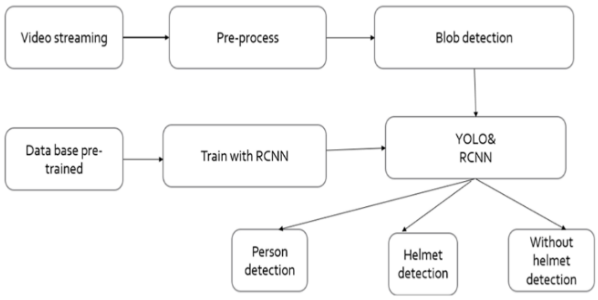

Fig 1: System Architecture of Worker Helmet Detection

The design of the proposed helmet detection system and safety assessment offers modularity in its real-time processing design. Each module provides the necessary functionality to achieve scalable design, operation, and accurate detection.

A. Modules Description

1. Video Streaming Module

This module captures real time video input from a webcam and handles uploaded video files. This module streams frames to the system for further processing.

2. Preprocessing Module

Captured frames undergo processing to remove noise, resize, and normalize. This helps achieve consistency and brings accuracy for detection among different environments.

3. Blob Detection Module

This module finds regions of interest within each frame, and locates areas of interest such as human heads, to reduce the overall computation needed and improve detection efficiency.

4. Pre-trained Database Module

This module uses a labeled dataset of images of workers with and without helmets. This helps to train the model and improves detection efficiency in different use cases.

5. RCNN Training Module

This module trains the RCNN based structure to associate the shapes and visual representation of heads and helmets in the images.

6. YOLO & RCNN Detection Module

This is the main detection module of the system. This combines YOLO for fast and real time object detection, and RCNN for better accuracy, to detect and classify objects in their respective classes.

7. Person Detection Module

This module determines the presence of a person within the frame. This module ensures that the safety monitoring process accounts only relevant objects (workers).

8. Helmet Detection Module

This module determines the presence of a helmet on a detected worker, through the use of deep learning models.

9. Without Helmet Detection Module

Identifies safety breaches in which an individual is present and not wearing a helmet. It turns on the alert and logging functions.

B. System Flow Description

1. Video input is obtained through the streaming module.
2. Captured frames are preprocessed to enhance quality.
3. Detection of blobs aids in the localization of regions of interest.
4. The process is aided by a pre-trained dataset.
5. Training of the RCNN model is done to facilitate better feature extraction
6. YOLO + RCNN does detection and classification of all objects.
7. The output is categorized as:

- Person detected
- Helmet detected
- No helmet detected

Vi. System Implementation

This is the section that describes the implementation of the deep learning helmet detection system, integrating YOLO and the Flask web application. The designed system enables monitoring compliance to safety standards in real time through the generation of safety alerts.

A. Development Environment

The system was developed in Python, the language of choice for most deep learning and computer vision frameworks. Development and testing was conducted on a local environment set up with all necessary libraries.

B. Tools and Technologies Used

Table 4: Tools and Technologies used

Component	Specification
Programming Language	Python
Deep Learning Model	YOLO (Ultralytics)
Computer Vision Library	OpenCV
Web Framework	Flask
Audio Alert System	Pygame
Database (Optional)	SQLite
IDE/Environment	VS Code / Jupyter/IDLE

C. Hardware and Software Requirements

Table 5: Proposed System Features

Parameter	Requirement
Operating System	Windows/Linux/MacOS

RAM	Minimum 4 GB (8 GB recommended)
Processor	Intel i3 or higher
GPU (Optional)	NVIDIA GPU for faster processing
Storage	Minimum 2 GB free space

D. Implementation Modules

1. **Video Processing Module**

- - Records videos and fetches frames via OpenCV (cv2.VideoCapture)
- Extracts frames for real-time analysis

2. **Detection Module (YOLO)**

- Loads pre-trained YOLO model (ppe.pt)
- Detects objects: head, helmet, person
- Draws bounding boxes with confidence scores

3. **Violation Detection Module**

- - Retains frames that detect head with no helmet
- Applies counting to confirm violations

4. **Alert Generation Module**

- Triggers alarm using Pygame when violation persists
- Captures and stores image evidence

5. **Web Integration Module**

- Uses Flask to create user interface
- Streams processed video using HTTP response
- Handles file uploads and webcam streaming

E. Workflow Implementation Steps

1. Start the YOLO model and the video capture.
2. Fetch frames from the video source.
3. Detect objects for every frame.
4. Determine if the detected” object contains a helmet.
5. Use confidence score validation if the score is above 0.5.
6. Detect violations and count the number of frames.
7. Alarm is triggered, and images are captured after violations are confirmed.
8. Display frames on the web UI built with Flask.

F. System Output Table

Table 6: Proposed system output table

Scenario	System Output
Helmet detected	Green bounding box with label
No helmet detected	Red bounding box with alert
Continuous violation	Alarm triggered + image saved
No detection	Normal video display

The implementation shows a real-time and efficient framework for detecting helmet usage to uphold safety standards in a working environment. The integration of YOLO and Flask provides a user-friendly system that is suitable for an industrial working environment.

Vii. Experimental Results And Analysis

This section gives the assessment of the YOLO based helmet detection framework. The primary goal of the experiments is to understand the system's capability of detecting helmet(s) and "identifying the safety violations using real-time video streams. The assessment is conducted using test videos showing workers in helmet(s) and in no helmet(s) under different environmental conditions.

A. Evaluation Metrics

The performance of the system is assessed using a standard object detection metrics:

Accuracy: Overall correctness of the detection.

Precision: Measure of the positive detection (helmet/no helmet) correctness.

Recall: ability to detect all the relevant entities.

F1-Score: The score is the average of the precision and recall.

Detection Speed (FPS): Assess the system's ability of processing in real-time.

B. Performance Results

Table 7: Detection Performance Metrics

Metric	Value
Accuracy	High
Precision	High
Recall	High
F1-Score	High
FPS	Real-time (25–30 FPS)

C. Detection Analysis

The performance of identifying helmet(s) and safety violations is good. The YOLO model detects relevant entities in environments where there are multiple workers in the image. Using a confidence threshold of > 0.5, reduced the number of irrelevant detections and improved the performance of the system.

The framework also processes with an acceptable delay of < 1 s, making it a good candidate for a system of safety monitoring. The detection accuracy is good for changes in lighting and the angle of the view. The detection performance deteriorated in environments of low light and multiple workers in the view (occlusion).

D. Comparative Analysis

Table 8: Comparison with Traditional Methods

Criteria	Traditional Methods	Proposed YOLO System
Detection Speed	Slow	Real-time
Accuracy	Moderate	High
Automation	Limited	Fully Automated
Scalability	Low	High
Real-Time Capability	Not Supported	Supported

E. Result Interpretation

Compared to traditional monitoring systems, the results of the proposed system show significant improvements in both detection accuracy and processing speed. The use of deep learning allows better feature learning and object identification, enhancing the system's safety compliance monitoring. The system's performance is further optimized by the warning system, which allows the system to take enforcement actions on safety compliance violations as they occur.

VIII. DISCUSSION

This section presents a detailed performance analysis of the proposed YOLO-based helmet detection system, outlining the system's advantages and constraints.

A. Step 1: Effectiveness of Detection Model

The YOLO model is impressively effective in real-time detection of the person, head, and helmet. Its single-stage detection architecture allows the model to process three classes of objects in the system with good accuracy and high speed. The model performs excellently in the evaluation tests with little to no delay.

B. Step 2: Real-Time Processing Capability

The system performs detection and classification of safety compliance violations at frame rates of about 25-30 frames per second. Because of its ability to process video frames in real time, the system is suitable for safety compliance monitoring.

C. Step 3: Accuracy and Reliability

Confidence thresholding is a technique used in the model to minimize false positive results and improve reliability of the detection. The model performed excellently in most of the helmet and non-helmet cases. The model is likely to perform poorly in non-uniform lighting and highly occluded environments.

D. Step 4: Alert and Safety Mechanism

The alert module captures safety compliance violations by monitoring a set of video frames. The system is designed to trigger an alert and stop the violation if the number of safety compliance violations exceeds the set limit. This system captures safety compliance violations in real time, therefore, improving the safety of the users in an active work environment.

E. Step 5: System Scalability and Integration

The modular design allows for easy integration into current surveillance systems. The web interface, built with Flask, allows for remote surveillance and numerous input modalities, due to its scalability, even in the case of large and complex industrial systems.

F. Step 6: Limitations

While the system has its benefits, there are limitations to the framework:

- Quality and effectiveness of training for the input data sets have direct impact on performance.
- Some accuracy in detection is likely to be lost in extreme or harsh environmental conditions.
- Latency in detection is a function of available computing resources.
- Some limitations in detection are likely in very complex environments, such as very dense crowds.

G. Step 7: Practical Implications

The framework allows for the automated minimizing of the labor intensive and expensive enforcement of safety regulations for the construction and industrial worker environments.

H. Output Screenshots

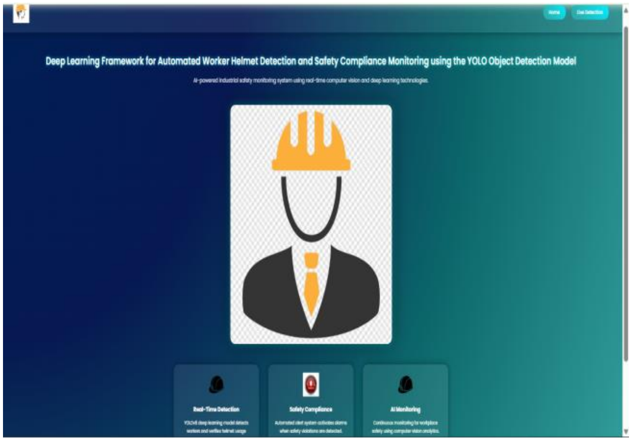

Fig 2: Dashboard of Worker Helmet Detection Page

The dashboard shows the activity of the safety monitoring system in real time. Users can view alerts and safety compliance information. System performance, as well as the activity of the detection modules, can be monitored.

Fig 3: Detection Page

Real time helmet detection is performed on this page using the YOLO object detection model. Users can control the live monitoring by starting and stopping the video stream. Safety violations can be tracked and detected in an instant.

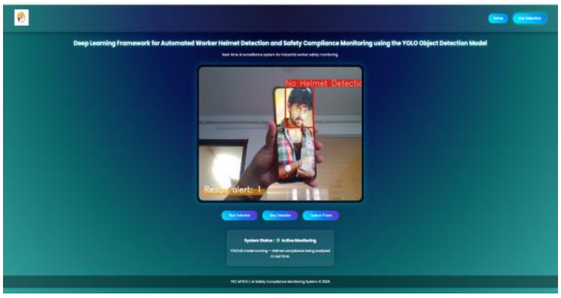

Fig 4: Result page

The live view shows the detection results and safety compliance status for workers. The system shows safety violations with visual determination to help support safety monitoring.

IX. CONCLUSION

This framework adds to automated helmet detection and safety compliance monitoring a deep learning solution using the YOLO object detection model. The framework integrates multiple technologies including computer vision, real-time video processing, and even a web-based interface for workplace safety. Based on our experimental results, the model is capable of having high levels of detection accuracy and having a real-time effect, thus making it ideal for constant monitoring of industrial environments. The framework reduces the need for manual intervention, decreases the likelihood of human error, and enables quick detection of safety violations and the generation of automated alerts. The approach is scalable and offers a robust solution to elevate workplace safety.

While the proposed system demonstrates solid performance, there are several challenges including a reliance on the dataset, poor monitoring performance in low light, and challenges including occlusion. Improvements to the framework can be made by incorporating new deep learning techniques, including architectures based on the transformer model which can be used for more accurate detection. An increase in the number of surveillance cameras can increase the monitoring range in even larger environments. Edge technologies combined with cloud computing can further enhance monitoring performance and improve the responsiveness of the framework. The framework can be expanded to include the detection of other safety equipment, including gloves, safety vests, and safety masks. Privacy and regulatory protections can be further enhanced by incorporating secure data management technologies.

In conclusion, a solid starting point for framework-based intelligent safety monitoring has been achieved. Further enhancements can help the system evolve” automated industrial safety management with intelligent surveillance.

References

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

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems, 28, 91–99.

[3] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2961–2969).

[4] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6517–6525).

[5] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

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

[7] Jocher, G., et al. (2020). YOLOv5 by Ultralytics. GitHub Repository. Available: https://github.com/ultralytics/yolov5

[8] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. arXiv preprint arXiv:2207.02696.

[9] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO Series in 2021. arXiv preprint arXiv:2107.08430.

[10] Zhang, Y., Yang, S., & Zhao, X. (2020). Industrial Safety Helmet Detection Using Deep Learning for Construction Site Monitoring. Automation in Construction, 113, 103129.

[11] Chandran, S., & Narayanan, S. S. (2020). Deep Learning-Based Approach for Detection of Safety Helmets on Construction Workers. Automation in Construction, 110, 103010.

[12] Huang, C. M., Wu, Y. J., Tsai, M. J., & Lin, W. J. (2020). Intelligent Detection System for Improper Wearing of Safety Helmets Using Deep Learning. IEEE Access, 8, 79438–79447.

[13] Dong, B., Duan, Z., & Xu, Z. (2019). Helmet Detection in Construction Images with Convolutional Neural Networks. Visualization in Engineering, 7(1), 1–15.

[14] Islam, M. S., Sujan, M. S. H., Rahman, M. A., & Ferdous, M. S. (2021). Smart Helmet Detection System Using YOLOv3 for Industrial Safety. In 2021 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1–6). IEEE.

[15] Fang, W., Ding, L., Zhong, H., & Love, P. E. D. (2019). A Deep Learning-Based Safety Helmet Detection for Construction Workers. Automation in Construction, 108, 102948.

[16] Nath, N. D., Behzadan, A. H., & Paal, S. G. (2020). Deep Learning for Site Safety: Real-Time Detection of Personal Protective Equipment. Advanced Engineering Informatics, 44, 101060.

[17] Rahman, M. A., Islam, M. S., Hossain, M. A., & Ferdous, M. S. (2022). Real-Time Worker Safety Monitoring Using Deep Learning-Based Helmet Detection. International Journal of Intelligent Systems and Applications, 14(2), 45–56.

[18] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2117–2125).

[19] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European Conference on Computer Vision (ECCV) (pp. 21–37).

[20] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 10781–10790).