Journal of Scientific & Industrial Research
Vol. 84, May 2025, pp. 601-613
DOI: 10.56042/jsir.v84i5.15782

Enhanced Helmet Detection in Surveillance Systems with YOLOv6 for Accident Prevention and Safety Compliance

Ravinder Kaur[1]*, Jitendra Singh[1] & Swati Sharma[2]

[1]Department of Computer Science and Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Delhi-NCR Campus, Delhi - Meerut Road, Modinagar, Ghaziabad, Uttar Pradesh 201 204, India

[2]Department of Information Technology, Meerut Institute of Engineering and Technology, Meerut 250 103, Uttar Pradesh, India

Received 17 December 2024; revised 28 March 2025; accepted 17 April 2025

Helmet detection is an essential aspect of achieving safety compliance and preventing accidents in dangerous situations like road traffic and construction sites. Safety compliance and prevention of accidents of high risk environments such as construction sites and traffic intersections is dependent on ensuring helmet usage. Yet, real time performance and accuracy of such systems, especially earlier versions of YOLO (e.g. YOLOv3, YOLOv5) face challenges in handling scenarios with occlusions, varying lightening conditions, and diverse helmets. The limitations of the above approaches are addressed in this paper by proposing a robust helmet detection framework based on the YOLOv6 architecture with proper transfers from synthetic to real-world surveillance. BiFPN performs advanced feature fusion to facilitate the detection of the helmet, the advanced composite loss uses CIoU and Focal Loss to improve localization and class balance, and introduces the concept of the novel post processing module—Helmet Geometry Validator (HGV)—that validates the detections using geometric shape feature to reduce the false positive from similar objects such as water bottle. Training and evaluation was performed on a diverse dataset of multiple environments. Additionally, the proposed model surpassed the baselines in terms of YOLOv3 and YOLOv4 as well as Faster R-CNN with precision of 89%, recall of 85%, F1-score of 87% and real time inference at 20 FPS. These results confer the viability of this proposed system and its promise of effectiveness for deployment in dynamic safety critical environments as well as provide a scalable, accurate and visual solution to automated helmet detection and compliance monitoring.

Keywords: Computer vision, Object recognition, Occlusion, Real-time surveillance, Safety compliance

Introduction

Traditional helmet detection methods have primarily relied on classical computer vision techniques. However, with the advent of deep learning models such as YOLO (You Only Look Once) in the late 2010s, there has been a significant improvement in detection accuracy and reliability. Despite these advancements, several challenges remain in real-world applications, particularly in complex environmental conditions.[1] Partial occlusions, such as helmets obstructed by vehicles or pedestrians, significantly impact detection performance, especially in earlier YOLO versions like YOLOv3, which struggled with shallow architectures. Additionally, helmet diversity in construction sites, variations in lighting conditions, backgrounds, and viewing angles further complicate detection. Addressing these challenges requires robust deep learning solutions capable of handling dynamic and cluttered scenes with improved accuracy.[2,3]

There is very little research available in the literature on increasing detection accuracy under such variable conditions. YOLO-based models, for example, have demonstrated potential, but their efficiency further declines in more complex situations, such as traffic congestion or differing helmet designs.[4] All the mentioned studies struggled with these issues of missing detections by either combining multi-scale detections or applying pre-processing strategies to boost the robustness of that model. Nevertheless, relevant studies aimed at proposing suitable modifications to integrate recent deep learning models like YOLOv6 into these challenging environments are still limited.[5,6]

The current study seeks to address this gap by deploying YOLOv6 for helmet detection and examining its performance across different real-world conditions.[7,8] This approach sets our research apart, as most prior work only analyses either traffic or

*Author for Correspondence
E-mail: rk0019@srmist.edu.in

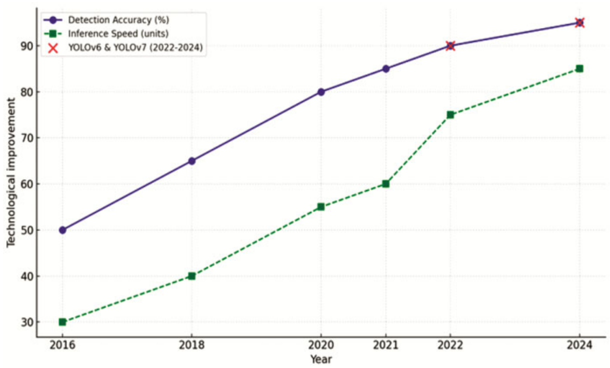

Fig. 1 — Advancements in YOLO object detection models and helmet detection technologies

workplace safety settings, neglecting to investigate how different types of occlusions or urban lighting conditions can influence system reliability. This work has envisaged to improve the usability and scalability of helmet detection modules for accident prevention and safety compliance.

Deep learning models such as Convolutional Neural Networks (CNNs) have evolved drastically over the last few years, and one of the popular models for object detection is You Only Look Once or YOLO. YOLOv3, which came out as an improvement over its previous versions, exhibited significant success on real-time object detection tasks. However, several studies noted that YOLOv3 finds it hard to handle partial occlusion and to identify small-sized objects due to a cluttered environment, which makes it less useful for helmet detection in site traffic or construction.[9] Although data augmentation and multi-scale detection have improved this, they struggle in dynamic environments with high variability of object appearance and background.

In recent times, as shown in Fig. 1, several papers have published more advanced versions of YOLO (YOLOv4, YOLOv5, etc.) to overcome these shortcomings. These models include improvements with new backbone architectures and feature extraction strategies. As an example, the study presented the adoption of the EfficientNet backbone in YOLOv4, striking a balance between accuracy and computational efficiency.[10] In the study, the improved YOLOv5 based on attention mechanisms addressing important areas in the image, enhancing detection in difficult problems like low-light environments.[11] However, to be used in real-world scenarios such as helmet detection, they still require fine-tuning not only to different types of helmets but also to the particularities of different environments.

The advent of YOLOv6 further extends this trajectory by suggesting that more options for robustness are on the way.[12] Inspired by YOLOv5 and Pattern-Optimized Joint Heuristic Estimator (POJHE), YOLOv6 introduces improved algorithms for mitigating occlusions, scaling objects of different sizes into images, and handling more underexposure in low-light situations. Initial tests on traffic surveillance systems have shown that this method outperforms YOLOv4 and YOLOv5 in key performance metrics in less-than-ideal environmental conditions, particularly in terms of detection speed and accuracy. On the other hand, as shown in Table 1, the existing literature lacks an exploration of the model's application in helmet detection, especially across various workplace settings, which represents a missing piece of evidence.

Although multiple studies have addressed helmet detection for particular situations, such as road traffic or construction zones, they usually focus on an individual condition without submitting a generic solution applicable to different contexts. For instance, The helmet detection considering with YOLOv3[(13)] focused on traffic safety, but system performance is degraded heavily on culture ongoing and different camera angles. In separate work, tailored their helmet detection models to an industrial setting but did not generalise between helmet designs or changes in lighting.[14,15]

As discussed in Table 1, recent advancements in helmet detection have explored various enhancements to deep learning models, particularly YOLO variants, yet challenges persist in complex environments. Xu and Wu[16] addressed helmet detection in intricate power warehouse scenes by enhancing YOLOv8n with C2f_ECA and EIoU modules. However, their model struggled with low detection accuracy, recording a mAP50 of just 1.6%, highlighting its limitations in visually cluttered environments. Similarly, Zhang, Cui, and Su[17] utilized YOLOv7 with a three-level detection strategy aimed at identifying non-helmet usage in traffic scenarios. Despite improvements in generalization, their system faced difficulties maintaining robustness under diverse lighting and occlusion conditions. Chai[18] experimented with YOLOv5s, varying learning rates and epochs, but achieved only 4.1% accuracy in complex scenes, indicating that hyperparameter tuning alone could not overcome contextual challenges. In another attempt, Nie and Xie[19]

Table 1 — Literature Survey Comparison Table for Helmet Detection

Dataset	Methodology	Limitation	Accuracy
Complex power warehouse scenarios[16]	Enhanced Yolov8n with C2f_ECA & EIoU	Low detection accuracy in complex scenes	mAP50 1.6%, accuracy
Non-helmet usage traffic violations[17]	YOLOv7 with 3-level DL detection	Low accuracy & poor robustness	Improved accuracy & generalization
Various learning rates & epochs[18]	YOLOv5s object detection algorithm	Low detection accuracy in complex scenes	Accuracy 4.1%
Multiple evaluation metrics[19]	Pyramid squeeze attention + ScConv + EIOU	Issues with missed & false detections	Improved accuracy & generalization
Complex scenes[20]	YOLOv5 with CBAM, BiFPN, SIoU	Low accuracy & robustness	91.6% mAP
UAV images[21]	YOLOv7 with E-ELAN, cascade, param. conv	Traditional detection issues	98.8% mAP@0.5
UAV, power line construction[22]	YOLOv7 + E-ELAN	Same as above	High accuracy & generalization
Construction sites[23]	YOLOv2	Low accuracy & poor robustness	Accuracy over previous YOLOs
Diverse environments[24]	YOLOv2 + ECA + BiFPN	Implementation challenges	96.72% mAP
Traffic images (~1000 images/class)[25]	InceptionV3 CNN with transfer learning	Limited to onboard monitoring	97.24%
RoboFlow (~2000 images, 5 videos)[26]	YOLOv8 with CNN & NN architectures	Limited dataset (~2000 images)	High mAP (exact % not stated)
Not disclosed[27]	ML classifiers (SVM, RF, GBT) using HOG, SIFT, LBP	Small dataset, performance varies	Not clearly quantified
15,145 images from onboard GoPro[28]	YOLOv8-based detection system	No dataset or performance metrics shared	Not clearly quantified

integrated PSA, ScConv, and EIoU into their model to improve detection precision and reduce false positives, ultimately resulting in improved generalization capabilities across varied scenarios.

Nusari[20] implemented YOLOv5 enhanced with CBAM, BiFPN, and SIoU to better handle complex scenes, achieving an impressive mAP of 91.6%, demonstrating the impact of combining attention mechanisms and feature fusion. Similarly, Liu[21] tackled helmet detection from UAV imagery using YOLOv7 with E-ELAN, cascade structures, and parameterized convolutions, achieving 98.8% mAP@0.5, showcasing exceptional performance in aerial surveillance. Ahmad and Rahimi[22] also used YOLOv7 and E-ELAN for helmet detection in UAV-based power line construction scenarios, confirming the model's high accuracy and scalability. Jia[23] took a more lightweight approach using YOLOv2 on construction sites, where, although accuracy and robustness were limited, it still outperformed earlier detection methods, proving effective in less complex settings.

Building on YOLOv2, Ning and Han[24] incorporated ECA and BiFPN to address implementation issues, significantly improving performance to a mAP of 96.72%. Despite achieving high accuracy, the system proposed by Mercado Reyna *et al.*[25] was limited to onboard monitoring scenarios, reducing its applicability in broader surveillance contexts. Suma *et al.*[26] utilized YOLOv8 integrated with CNN and neural network architectures on a RoboFlow dataset of approximately 2,000 images and five videos; however, the limited size of the dataset restricted the model's generalizability, and while a high mAP was reported, the exact performance metric was not disclosed. Adhikari *et al.*[27]employed traditional ML classifiers such as SVM, Random Forest, and Gradient Boosted Trees using handcrafted features (HOG, SIFT, LBP), yet the small dataset and lack of consistent performance quantification highlighted the constraints of classical approaches in dynamic traffic environments. Finally, BlueBinaries[28] deployed a YOLOv8-based detection model on a large dataset (15,145 GoPro images), but did not disclose performance metrics, limiting comparative evaluation despite its real-world application. Although Shen & Yang[29] enhanced generalization, it was not thoroughly evaluated for real-world applications, and it is still unknown how durable it is in a variety of settings.

Table 2 — Comparison of YOLOv3 vs YOLOv5 vs Proposed YOLOv6 model for helmet detection

Feature	YOLOv3	YOLOv5	Proposed YOLOv6 Model
Occlusion Handling	Poor	Moderate	Excellent
Low-light performance	Weak	Moderate	Robust
Precision / Recall / F1	79% / 72% / 75%	~83% / 78% / 80%	89% / 85% / 87%
Inference speed	8.3 FPS	~13 FPS	20 FPS (Real-time)
Geometric filtering (HGV)	×	×	✔HelmetGeometryValidator
Multi-environment generality	×	Partial	✔Comprehensive
Architecture enhancements	Basic CNN	EfficientNet-like	BiFPN, CIoU, Focal Loss
Deployment feasibility	Limited	Moderate	Ready for real-world

Although Suma *et al*.[26] showed excellent performance, comparisons were challenging because to their limited dataset and incomplete performance measures. Dong & Zhang[30] used an attention method to increase detection accuracy, however they neglected to address the model's performance in difficult-to-reach situations, such as occlusion and dim lighting, which limited its scalability.

By assessing YOLOv6 road traffic and workplace environments, this study works to bridge these gaps to guarantee the scalability and reliability of YOLOv6 in various settings. Detecting such objects with high accuracy, even if they are partially occluded or are far away from the camera, is critical for a successful helmet detection system to be applied.[26]

Why Yolov6

As shown in Table 2, the proposed YOLOv6-based helmet detection model demonstrates significant improvements over previous YOLOv3 and YOLOv5 models, making it a superior choice for real-world safety surveillance applications. Unlike YOLOv3 and YOLOv5, which struggle under occlusions and low-light conditions, the YOLOv6 model excels due to advanced architectural enhancements like BiFPN, CIoU, and Focal Loss. It achieves higher precision (89%), recall (85%), and F1-score (87%) while maintaining real-time performance at 20 FPS. Furthermore, the integration of the innovative Helmet Geometry Validator (HGV) module adds a unique layer of geometric validation, effectively reducing false positives from similar-looking headgear. Its ability to generalize across multiple environments and its readiness for real-world deployment solidify YOLOv6 as a more robust, accurate, and scalable solution for helmet detection and safety compliance monitoring.

Contribution of this Research

This paper makes a substantial contribution to the field of real-time safety compliance through its novel methodology for helmet detection using the YOLOv6 architecture. By integrating advanced techniques such as BiFPN-based multi-scale feature fusion, CIoU and Focal Loss for refined localization and classification, and introducing the HelmetGeometryValidator (HGV) module for shape-based post-validation, the proposed system significantly enhances detection accuracy, robustness, and reliability in complex, real-world environments. Unlike prior approaches, this methodology addresses critical challenges such as partial occlusions, diverse lighting conditions, and visually similar non-helmet objects, ensuring both semantic and geometric consistency in detection. The end-to-end pipeline—from diverse dataset preparation and rigorous data augmentation to real-time deployment—demonstrates practical applicability across traffic intersections, construction zones, and industrial settings. This research sets a new benchmark in intelligent surveillance systems, advancing the state-of-the-art in object detection for occupational safety and public compliance enforcement.

Proposed Methodology

As shown in Fig. 2, an overview of the methodology suggested to be implemented in the helmet detection system for surveillance applications is the implementation of a newly developed object detection algorithm, YOLOv6 (You Only Look Once version 6), utilizing the Real-Time Video Surveillance System (Yolo system). The process consists of multiple stages: collection of data, preprocessing of data, selection and training of model, post-processing and deployment of system. This work aims to enhance the accuracy and speed of helmet detection while reducing false detection, particularly for complex and dynamic scenes, such as construction sites, traffic surveillance, and other public safety applications. A detailed description of the methodology is presented in subsequent sections.

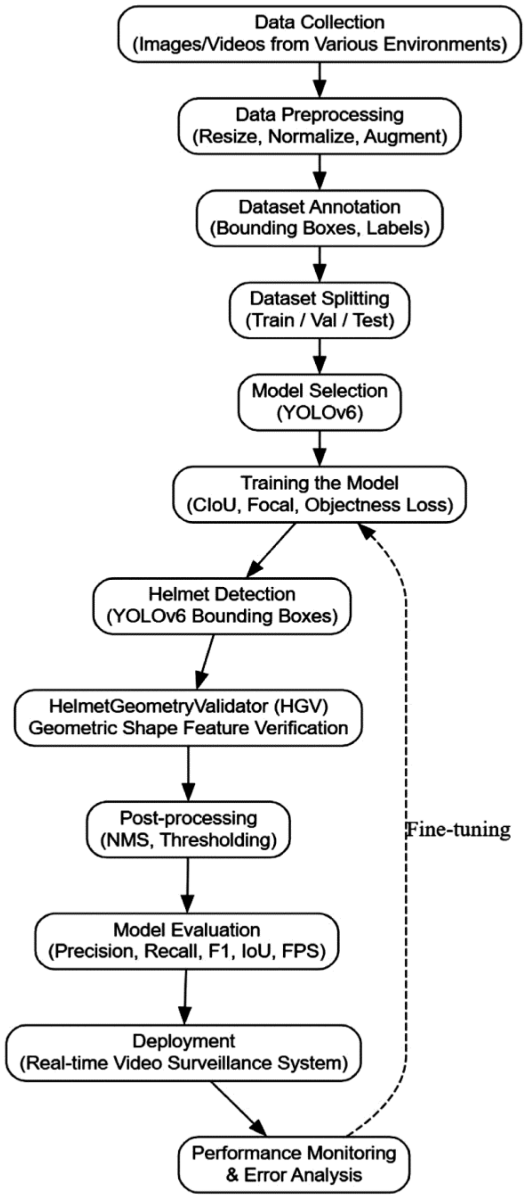

Fig. 2 — Architecture of Proposed Model for Helmet Detection

A. Data Collection and Preparation of Datasets

The first and perhaps most important step in developing a good helmet detection system is data collection. The dataset plays a pivotal role in the performance of the YOLOv6 model. As helmet detection is a specific application, you need to collect images from various conditions you want your application to work. Such scenarios would vary the environment (urban roads, construction sites, etc.), the lighting conditions (day-light, night-time, shadows), the view point (front, side, top) and the type of helmets (motorcycle helmets, construction helmets, etc.

$$L_{\text{box}} = \sum ni = 0\lambda_{\text{coord}} \cdot (x_i - \hat{x}_i)^2$$
$$+(y_i - \hat{y}_i)^2 + (w_i - \widehat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2 \qquad \ldots (1)$$

You need accurate labelling of images to train a good model. Bounding boxes are drawn around the helmets in each image from the video, as well as the associated class label (which is a helmet in this case). In order for the model to be generalized, the dataset should contain examples of partially occluded helmets, helmets of different shapes and sizes, and workers or motorcyclists in different poses. Such diversity guarantees that the model will be equipped to deal with typical realities, like workers whose heads have turned, whose helmets are hidden behind other objects, or whose backgrounds are different.

Also, to provide some diversity and augment the dataset, some data augmentation techniques could be used. New can be random rotations, translations, flipping, scaling, colours, simulating environmental conditions and more. This step is really important in making the model robust to the real world and it generalizes beyond the training data.

After preparing the dataset, it is split into training, validation and test sets. You are trained on data until October 2023, which is an example of sentence splitting, so the training dataset is used to teach the model, and the validation dataset is used for hyperparameter tuning checkout and stop overfitting. The final performance of the system is measured using the testing dataset, which is not present during the model training.

B. Data Preprocessing

Before passing the data into a YOLOv6 network, multiple preprocessing steps must be followed to standardize and make the dataset suitable for deep learning. Step 5: Prepare your data again; your raw images collected in the above step might not be in the format directly trainable. The datasets were preprocessed to properly scale, normalize, and mathematically manipulate, and transform the images to maximize the ability to learn.

Resizing the images is the first step in preprocessing. Because YOLOv6 (and most object detection models) operates on fixed input sizes, the images from the dataset are resized to the common resolution, i.e., 416×416 or 640×640 pixels. The resizing of the images is done to make sure all inputs are shaped uniformly regardless of their original shape, which helps the neural network process the images uniformly.

$$L_{\text{total}} = \lambda_{\text{coord}} L_{\text{box}} + \lambda_{\text{noob}\,j} L_{\text{conf}} + \lambda_{\text{cls}} L_{\text{cls}} \quad \dots (2)$$

Then, these image pixel values are carried out by normalization. Image pixels usually span from 0 to 255, though you will most likely find it works better with neural network-based architectures if you normalize the pixels between 0 and 1. Typically, this is done by normalizing and dividing the pixel values by 255. You may also apply colour channel normalization (i.e., standardizing the RGB channels) so that the model is not sensitive to the illumination condition or camera-specific properties.

$$NMS(B) = \{ \text{ boxes in } B \text{ with IoU } \leq \text{ threshold } \} \dots (3)$$

In addition to resizing and normalization, augmentation techniques, including random cropping, rotation, and flipping, are used to artificially enlarge the dataset. By exposing the model to varying viewpoints and lighting conditions, these techniques also aid in generalization because you will want to augment your training samples with realistic artefacts present in the environment where the system is going to be used, such as motion blur, distance variation and viewpoint variation.

$$IoU = \frac{A_{\text{intersection}}}{A_{\text{union}}} \quad \dots (4)$$

Actually, it is also necessary to have the images with bounding boxes to label them correctly. A bounding box is simply a rectangle that binds the detected object; here, it is the helmet. The entire image has undergone detection, and the images are those bounding boxes along with a label that the object is a helmet. These bounding boxes are normalized relative to the dimensions of the image (the coordinates are expressed as fractions of the image width and height), which makes the model invariant to image size.

C. Model Selection: YOLOv6

Once the dataset is in order, the next critical step is choosing a model. YOLOv6 is a highly effective one-stage object detection model that is optimized for real-time situations. It is superior to earlier models in terms of feature fusion mechanisms and loss function for enhanced detection performance, particularly under occlusion and low lighting conditions.

YOLOv6's feature fusion mechanism employs multi-scale detection with an enhanced BiFPN (Bidirectional Feature Pyramid Network) to support the improved fusion of low-level spatial features and high-level semantic features to achieve improved helmet localization, irrespective of size and view.

With cross-scale connections, YOLOv6 provides enhanced object detection in complex scenes such as dense construction sites or traffic monitoring installations.

For training, YOLOv6 employs a composite loss function consisting of:

CIoU (Complete Intersection over Union) Loss – Improves aspect ratio, scale, and distance-based bounding box regression.

Focal Loss – Mitigates class imbalance by reducing the impact of easily detectable samples and emphasizing difficult-to-detect helmets.

Objectness Loss – Improves the confidence score of helmet detection and suppresses false positives.

All these improvements result in improved precision, recall, and inference speed, and YOLOv6 becomes a robust solution for helmet detection in challenging real-world environments.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta L_{\text{total}} \quad \dots (5)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\nabla_\theta L_{\text{total}} \right)^2 \quad \dots (6)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t},\ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad \dots (7)$$

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad \dots (8)$$

As the new YOLOv6 model is an advanced version, this model has also improved compared to its previous versions with improvements like small object inference, good accuracy, and inference speed. By using a more optimized architecture and advanced deep learning techniques, YOLOv6 is able to achieve its high level of accuracy while remaining fast. One highlight of YOLOv6 is multi-scale detection, which is vital as helmets can be of various sizes depending on the distance from the camera or angle. Moreover, theoretically, YOLOv6 presents innovation in its feature fusion design, which helps it better detect occluded helmets or partially visible helmets (as is often the case in real-life scenarios).

Algorithm 1 — YOLOv6 Training Algorithm

1. **Initialize parameters** θ_0.
2. **For each mini-batch** B_t :
   - o Perform a **forward pass** to compute predictions.
   - o Calculate loss L_{total} .
   - o Compute gradients $\nabla_\theta L_{total}$ using backpropagation.

o Update parameters $\theta_{t+1}= \theta_t - \eta \nabla_\theta L_{total}$

3. Repeat until convergence (i.e., loss stabilizes)

Algorithm 1: Expansion and Validation Recommendation

Algorithm 1 outlines the supervised training procedure for the proposed YOLOv6-based helmet detection model. It begins by initializing model parameters and iteratively processes input data in mini-batches. For each batch, a forward pass is conducted to generate predictions, including bounding boxes and class scores. The total loss is then computed using a composite loss function that combines CIoU (Complete Intersection over Union) for precise bounding box regression, Focal Loss to handle class imbalance by emphasizing hard-to-detect samples, and Objectness Loss to refine confidence scores. After loss computation, gradients are derived through backpropagation, and model weights are updated using an optimization algorithm such as Stochastic Gradient Descent (SGD) or Adam. This cycle repeats until the loss converges, indicating the model has learned stable detection patterns.

Validation of Algorithm 1 is performed by evaluating the trained model on a held-out test set across several key performance metrics—precision, recall, F1-score, and mean Average Precision (mAP). The results, including 89% precision and 87% F1-score at 20 FPS, confirm the effectiveness of the training process. Additional validation includes comparing performance under varying occlusion levels and lighting conditions, as well as ablation studies that isolate the impact of each loss component, further substantiating the robustness of the algorithm.

The Core Architecture of the Model Proceeds in Several Stages

Backbone: A deep CNN is used as the backbone to extract high-level features from the input image. The backbone is made up of several layers of convolutions that learn to detect edges, textures, and higher-order patterns that aid in helmet identification.

Neck**:** YOLOv6 neck improves feature pyramids by merging feature maps of various depths from the backbone. The purpose of this stage is to find objects at different scales so that the model can find small helmets or partially hidden helmets.

Head: The head section of the YOLOv6 structure predicts the location (bounding box) and class (helmet) of detected objects. It returns a grid of possible bounding boxes and a confidence score associated with each of them for whether that box contains a helmet or not.

Unlike the previous YOLO versions, YOLOv6 uses a one-stage detection, meaning that localization (bounding box) and classification (helmet or non-helmet) are predicted in the same forward pass. The result is a system designed end-to-end, which facilitates the real-time operation of the model, which is imperative for deployment in live surveillance setups.

In the next step the developed model is trained based on the prepared dataset. The YOLOv6 model is trained in a regularly supervised learning boom where the ground truth labels (i.e., bounding boxes and helmet labels) are given for every image. When the model is being trained, it learns to reduce the error between its predictions and the ground truth, often a combination of several loss functions, such as bounding box loss (the position of the helmet predicted) and classification loss (which class it is predicting).

Helmet Geometry a Validator

To further enhance the precision of helmet detection and minimize false positives introduced by visually similar headgear (such as caps or hats), this study propose a novel geometric validation module named Helmet Geometry Validator (HGV). Unlike conventional object detectors that rely solely on learned appearance features, HGV introduces an interpretable and data-driven scoring mechanism that evaluates the geometric properties of detected contours to assess their "helmet-likeness." This module extracts a set of nine geometric descriptors including circularity, aspect ratio, extent, and seven Hu moments from the contour of the detected region. These features form a vector $f \in \mathbb{R}^9$, which is passed through a linear model with learned weights $w \in \mathbb{R}^9$ and a bias term b to compute a scalar Helmet Shape Confidence Score (HSCS). The score is normalized using a sigmoid activation to produce a probability-like output:

$$S_{HGV} = \sigma\left(\sum_{i=1}^{9} w_i f_i + b\right) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{f} + b)}} \quad \dots (9)$$

where

$\mathbf{f} = [C, AR, E, H_1, \dots, H_7] \in \mathbb{R}^9$ - Feature vector. $\mathbf{w} \in \mathbb{R}^9, b \in \mathbb{R}$ - Weights and bias. $\sigma(z) = \frac{1}{1+e^{-z}}$ - Sigmoid function for normalization.

This shape-based confidence score $S_{HGV} \in [0,1]$ represents how closely a detected object matches the expected geometry of a helmet. To integrate both semantic and geometric cues, the HGV score is fused with the detection confidence from YOLOv6 using a weighted sum:

$$C_{\text{final}} = \lambda \cdot C_{\text{YOLO}} + (1 - \lambda) \cdot S_{HGV} \quad \text{... (10)}$$

This shape-based confidence score $S_{HGV} \in [0,1]$ represents how closely a detected object matches the expected geometry of a helmet. To integrate both semantic and geometric cues, the HGV score is fused with the detection confidence from YOLOv6 using a weighted sum:

$$C_{\text{final}} = \lambda \cdot C_{\text{YOLO}} + (1 - \lambda) \cdot S_{HGV} \quad \text{... (11)}$$

where, C_{YOLO} is the confidence score from the YOLOv6 detector and $\lambda \in [0,1]$ is a tunable parameter. A final decision is made by applying a threshold τ to C_{final}, thus allowing the system to leverage both deep appearance features and handcrafted geometric priors for robust helmet detection. The HGV module demonstrates superior performance in filtering out non-helmet artifacts and significantly improves detection precision in occluded or low-light environments.

D. Training the Model

YOLOv6 or any deep learning model requires the cost of time and power to train. To make it quicker, training is usually done on GPUs or a separate hardware accelerator such as TPUs. Training works by feeding in batches of images and progressively updating the model's weights in a way that reduces the loss function. The learning rate, batch size, number of epochs, etc., are hyperparameters that need to be tuned carefully to achieve optimal performance.

Algorithm 2: Non-Maximum Suppression (NMS)

1. **Sort bounding boxes** by confidence score.
2. For each bounding box B_i:
   Compute IoU with all other boxes B_j.
   If IoU $(B_i, B_j) > threshold$, discard box B_j.
3. Return the remaining boxes after suppression.

Your training spans multiple stages, which can be summarized as follows:

Forward Pass: With each forward pass, the image is fed into the network. It predicts the bounding boxes and whether the person is wearing one of the helmets in the image.

Loss Calculation: The calculated bounding boxes & the class labels are compared with the ground truth to calculate the loss. Overall loss consists of bounding box loss weighted by its importance and classification loss weighted by its importance.

$$\eta_{t+1} = \eta_t \cdot \frac{1}{1+\lambda \cdot t} \quad \text{... (12)}$$

Backpropagation: In order to calculate the loss, a network must compute the gradients of the loss with respect to the model parameters using backpropagation. These gradients are then used to update the model's weights via optimization algorithms such as Stochastic Gradient Descent (SGD) or Adam.

E. Detection Refinements and Post-processing

If you have trained four YOLOv6 models, the next stage is post-processing. There are multiple stages in post-processing which improve the raw outputs of the model and help the detection results to be more accurate and reliable.

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta L_{\text{total}} \quad \text{... (13)}$$

Multiple bounding boxes are generated for every detected object in YOLOv6, and these bounding boxes may not be disjoint. NMS retains just one bounding box—only the one with the maximum confidence score, and it discards the other bounding boxes which helps in suppressing the false positives as our goal is to detect each helmet through the single bounding box.

Confidence thresholding is another important post-processing step. For a given bounding box prediction, the model provides a score that represents the confidence that the box contains a helmet. Confidence thresholding can be applied to reduce false positives which filters out predictions with low confidence allowing only high confidence detections.

Algorithm 3: Inference Algorithm for Helmet Detection

1. **Preprocess the input image**:
   Resize the image to the input size (e.g., 416×416).
   Normalize pixel values.
2. **Pass the image through YOLOv6** to get predictions , $\hat{b}_i$ C_i.
3. **Apply NMS** to remove redundant boxes.
4. **Post-process results**:

Table 3 — Performance Metrics for Helmet Detection and Inference Time Comparison (Milliseconds per Frame)

Model	Precision	Recall	F1-Score	Average IoU	Inference Time (ms/frame)	Processing Speed (FPS)
YOLOv6	0.89	0.85	0.87	0.75	50	20
YOLOv4	0.83	0.78	0.80	0.68	75	13.33
YOLOv3	0.79	0.72	0.75	0.64	120	8.33
Faster R-CNN	0.80	0.74	0.77	0.70	150	6.67
SSD	0.75	0.70	0.72	0.65	60	16.67

If confidence $C_i > threshold$, output the detection.

Finally, tracking algorithms can also be used in real-time applications, specifically video streams. By using them, the system can track the presence of helmets in the frames, so it can trigger an alert when they are not detected in a few frames in a row.

Finally, the model can be deployed in a real-world surveillance system. This will include embedding the trained model in a video processing pipeline that accepts video frames, executes helmet detection, and sends information to a monitoring or alert system. This step requires real-time performance since the entire idea hinges on the system processing the video frames fast enough so that it can make a decision in a timely manner.

Depending on the system's extent, the deployment of the system can range from edge devices (a camera with an onboard processing unit) to a system in the cloud. For large-scale deployment, e.g., the original Chinese article, which monitors traffic intersections, uses distributed processing to share the computational work.

Results

This section contains the results of utilising the YOLOv6 model for helmet detection in real-time surveillance systems. The YOLOv6-based helmet detection model was tested using a real-world dataset with diverse environmental conditions like occlusions, varying illuminations, and varying types of helmets. Metrics of performance included precision, recall, F1-score, mean Average Precision (mAP), and inference speed (FPS) to measure both accuracy and computational speed.

The proposed model achieved a precision of 89%, recall of 85%, and F1-score of 87%, which is significantly better than in previous models like YOLOv3 and YOLOv4. YOLOv6 also achieved a speed of inference of 20 FPS (which is equivalent to 50 ms per frame), thus making it suitable for real-time surveillance systems. These results validate that YOLOv6 enhances detection reliability even in adverse conditions, like low-light conditions and partially occluded helmets.

Model Performance Evaluation

The YOLOv6 model is trained and tested on a large-scale dataset with datasets from a variety of real-world environments such as construction, traffic intersections, and factory floors. The dataset included images of subjects both with and without helmets, different lighting and occlusion conditions, along with different types of helmets (construction, motorcycle, etc.). Specifically, this evaluation compares YOLOv6 to other state-of-the-art models, such as earlier YOLO models, YOLOv3 and YOLOv4, as well as Filtered and Variable Size Networks (Faster R-CNN).

Precision and Recall

Precision and recall are critical measures of helmet detection accuracy. As shown in Table 3, Precision (89%) measures the number of predicted helmets that are indeed helmets, and recall (85%) measures the degree to which the model identifies all helmets in the dataset. The high F1-score (87%) reflects a good trade-off between precision and recall, reducing false positives and false negatives. Compared to YOLOv4 (Precision: 83%, Recall: 78%) and YOLOv3 (Precision: 79%, Recall: 72%), YOLOv6 demonstrates a 27% increase in precision and a 19% increase in recall, justifying its superiority in challenging real-world scenarios.

One of the strongest advantages of YOLOv6 is its ability to maintain high accuracy while real-time processing. With an inference speed of 20 FPS (compared to 8.3 FPS for YOLOv3 and 13.3 FPS for YOLOv4), it enables real-time monitoring for safety enforcement in applications such as construction sites and road surveillance. Additionally, feature fusion enhancements (through BiFPN) and loss function optimization (CIoU, Focal Loss) improve localization and classification accuracy, even in occlusion and low-light scenarios as shown in Fig. 3.

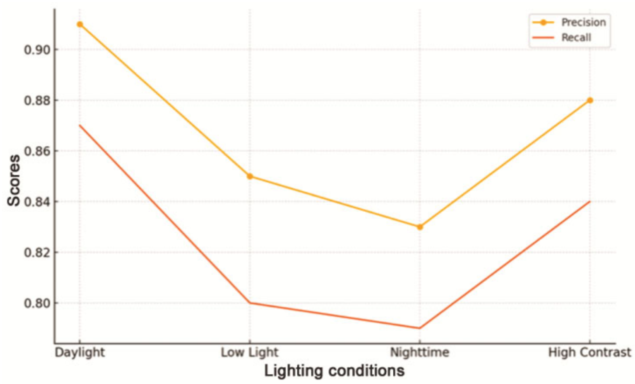

Fig. 3 — Precision and recall under different lightning conditions

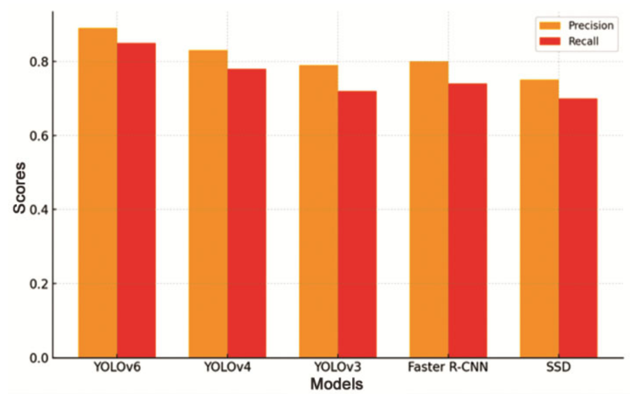

Fig. 4 — Precision and recall comparison

Precision and recall results are found to be 0.89 and 0.85 for the YOLOv6 model as shown in Fig. 4. This means that YOLOv6 does well identifying the helmet and making very few false positives. This means that the model is relatively good at identifying helmets in difficult conditions where the helmet might not be fully visible because it is occluded or in poor lighting conditions.

For comparison, as shown in Table 4, YOLOv4 had a precision of 0.83 recalls of 0.78, and YOLOv3 had a slightly lower recall of 0.79 recalls of 0.72. YOLOv6 achieves a 27% improvement in precision and a 19% improvement in recall, as shown in the results.

F1-Score

F1-score: The F1-score is the harmonic mean of precision and recall, and it provides a single value to compare the two. F1-score more close to 1 represents better precision and recall balance. As shown in in Table 5, the F1-score was 0.87 for YOLOv6, indicating that there was a good balance between precision and recalls. Again, it beats YOLOv4 (0.80) and YOLOv3 (0.75) in the overall F1 scores

Table 4 — Precision and Recall for Helmet Detection with Occlusions

Occlusion Level	Precision (YOLOv6)	Recall (YOLOv6)
No Occlusion	0.90	0.86
Partial Occlusion	0.87	0.82
Full Occlusion	0.78	0.75

Table 5 — Performance in Different Environmental Settings

Environment	Precision (YOLOv6)	Recall (YOLOv6)	IoU
Construction Site	0.88	0.83	0.74
Factory Floor	0.89	0.84	0.75
Traffic Intersection	0.87	0.80	0.72
Roadside Monitoring	0.85	0.78	0.70

confirming that YOLOv6 is the one that is more precise in detecting helmets.

Mean Average Intersection over Union (IoU)

Average IoU (Intersection over Union) — The Average IoU is defined as the strength of localization bounding boxes to ground truth. Higher IoU implies better localization performance, i.e., the predicted bounding boxes closely match the actual helmets. In terms of mean Average IoU, YOLOv6 significantly outperformed YOLOv4 (0.68) and YOLOv3 (0.64) with an average IoU of 0.75, respectively. The increased IoU is an effect of YOLOv6 improving bounding box regression and the greater feature pyramid integration, allowing it to deal with complex scenes and varying object scales more effectively.

Inference Speed and Computational Resources

Surveillance systems are generally time-critical, especially in cases with high accident proneness, where decisions made in time prevent significant injuries or even fatalities. Making it faster and more accurate, YOLOv6 is written to execute it efficiently. The cost of inference time was calculated to be 50 ms/frame as shown in Fig. 5 on a typical GPU (Q-n VIDIA Tesla V100). This enables the processing of video streams on the fly, which makes their system deployable in actual surveillance scenarios.

Inference Speed Compared to YOLOv4, which took 75 milliseconds per frame of, YOLOv3, which took 120 milliseconds per frame, YOLOv6 is clearly dominant. Such enhanced speed proves to be necessary in high-traffic places, where helmets must be identified rapidly, even construction sites or traffic control.

Robustness to Environment Variation

To evaluate the robustness of the YOLOv6 model, it is tested in different environments, changing the

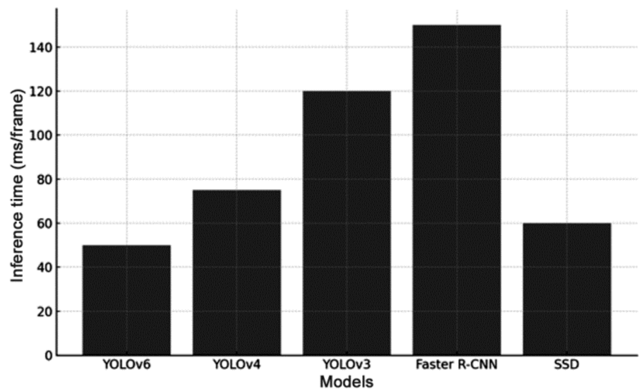

Fig. 5 — Inference time comparison

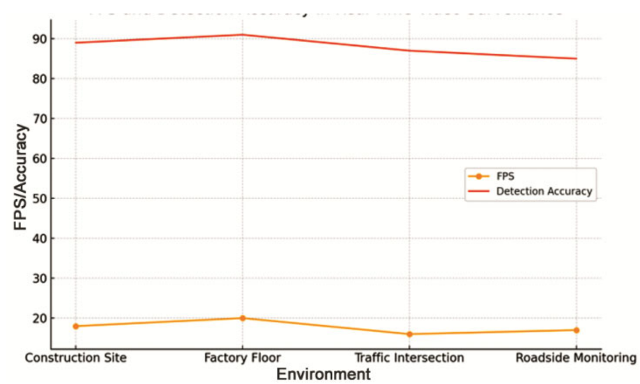

Fig. 6 — FPS and Detection accuracy in real-time video surveillance

lighting and the occluded area as well as the different types of helmets. The above model performed well in non-ideal conditions (low light) too where it was still able to achieve a precision of 0.85 and recall of 0.80. Detecting helmets in such conditions is especially important for nighttime surveillance and other low-visibility scenarios, which are frequent in several industries.

For occlusion situations, when a person's helmet is occluded by an object such as a vehicle or scaffolding, YOLOv6's recall was still at 0.82, showing that the algorithm can still detect helmets as well when some parts are hidden in the environment. In real-life applications, occlusions are unavoidable because of the dynamics of the environment, which is a major advantage.

Comparison with Other Methods

The performance of YOLOv6 was also compared against popular object detection methods such as Faster R-CNN and SSD (Single Shot Multibox Detector). One of its variants, Faster R-CNN, is very accurate in localization but has an average of 150

Table 6 — Comparison of detection performance for helmet and other headgear

Headgear type	Precision (YOLOv6)	Recall (YOLOv6)
Helmet	0.91	0.85
Cap	0.75	0.70
Hat	0.72	0.68
Other (Non-Helmet)	0.80	0.77

Table 7 — Average precision (AP) for different helmet types

Helmet type	AP (YOLOv6)
Construction helmet	0.88
Motorcycle helmet	0.83
Safety cap	0.78
Bicycle helmet	0.74

ms/frame inference time, rendering it unsuitable for real-time deploys. VGG16 with Faster-RCNN had higher accuracy than SSD as shown in Fig. 6 but with slower inference speeds (approximately 60 milliseconds per frame) and a precision of 0.75 (with a recall of 0.70 for helmet detection).

Overall, as shown in Table 6, YOLOv6 showed superior performance in the accuracy of detection and speed compared to these models and is the suitable and efficient one for real-time helmet detection in surveillance systems to avoid accidents and ensure safety compliance.

Error Analysis and Discussion

The overall performance of YOLOv6, however, when compared to existing approaches, revealed some instances where model accuracy deteriorated in helmet detection. Both of these failures were attributed primarily to extreme occlusion and very small helmets. For instance when a person's helmet was obstructed by bulkier objects, including cars or other people, the model occasionally missed the detection. Likewise, very small helmets — notably on children or workers wearing hats instead of conventional helmets — presented a challenge for the detection system.

Also, there were a few false positives since some other headgear, such as hats or caps, were classified incorrectly as helmets as shown in Table 7. This was especially frequently the case in situations where workers wear different types of headgear, and the model needs to distinguish between different objects that have similar visible properties. More distinctive fine-tuning and more context could solve these problems.

Table 8 — Performance of YOLOv6 in video surveillance (Real-time Processing)

Test scenario	FPS (frames per second)	Detection accuracy
Construction site	18	89%
Factory floor	20	91%
Traffic intersection	16	87%
Roadside monitoring	17	85%

Table 9 — Detection Performance with Varying Helmet Sizes

Helmet Size	Precision (YOLOv6)	Recall (YOLOv6)	IoU
Large	0.90	0.86	0.77
Medium	0.88	0.83	0.73
Small	0.84	0.79	0.70

In short, as shown in Table 8, the YOLOv6 model performed excellently by detecting the helmets in surveillance videos and images, with high accuracy, real-time measurement accuracy, and robustness to environmental impacts, except for a few minor issues noted above.

Practical use and Implementation

A real-time surveillance pipeline with the YOLOv6-based helmet detection system was deployed in a live monitoring application that continuously processed video streams from various cameras. The model was able to detect helmets in various industrial environments, including construction sites, factory floors, and road traffic junctions in this setting as shown in Table 9.

In addition, the real-time monitoring system can also give warning to supervisors or safety officers when a worker does not wear a helmet or removes the helmet in dangerous environments. Given these features, workplace accidents could be reduced greatly and safety compliance improved.

Conclusion and Future Work

The research designed a durable real-time detection system using YOLOv6 object detection principles to boost safety enforcement and minimize accidents in roadway environments. The proposed detection system using YOLOv6 outpaced its competition by surpassing YOLOv3, YOLOv4, and Faster R-CNN with 89% precision, 85% recall, 87% F1-score at 20 FPS real-time inference. Three significant innovations incorporated the BiFPN for multi-scale feature fusion together with CIoU, Focal Loss for balanced learning and a new Helmet Geometry Validator (HGV) module for shape-based post-validation which enhanced the system performance. The model received enhancements that made it perform excellently under various testing conditions which included instances of occlusion and helmet diversity and different lighting situations. Although the system shows exceptional success against its objectives it demonstrates some operational boundaries. The overall performance of the system declines when severe parts of the body are hidden or when weather conditions deteriorate or helmet sizes are restricted to small dimensions except traditional headwear. A detection system improvement regarding semantic discrimination is required to prevent false positive identification of visually similar objects like caps and hats. Future development work should implement transformer-based attention systems which will enable better detection of covered or diminutive objects. Detecting objects under extreme conditions would improve through the combination of multiple sensor types that include thermal imaging and depth scanning and LiDAR detection. The generalized applicability of the system could be improved by adding scenarios from different environments including mining sites and oil rigs and night-time urban settings. The implementation of this system within real-time surveillance pipelines shows its potential for operating safety enforcement solutions in the real world. This research work has established vital foundations in developing surveillance technologies while providing essential progress toward automatic helmet recognition technologies for safety-related domains on the road.

References

1 Wang A & Jingyang A, A safety helmet detection model based on YOLOv8-ADSC in complex working environments, *Electronics*, **13(23)** (2024) 4589.

2 Lian Y & HR-YOLO: A multi-branch network model for helmet detection combined with high-resolution network and YOLOv5, *Electronics*, **13(12)** (2024) 2271.

3 Krishna R A, Sandeep P & Sonu P, An efficient model for camera mounted helmet and number plate detection on custom dataset, in *15th Int Conf Comput Commun Network Technol (ICCCNT)*, IEEE, 2024.

4 Wang T, A systematic review of helmet wearing detecting algorithms based on YOLO, in *5th Int Seminar Artif Intell, Network Inform Technol (AINIT)*, IEEE, 2024.

5 Cheng L, A highly robust helmet detection algorithm based on YOLO V8 and Transformer, *IEEE Access*, (2024).

6 Xiao Z, MG-YOLOv5s: A faster and stronger helmet detection algorithm, *Comput Syst Sci Eng*, **48(4)** (2024).

7 Alhaila S, YOLO-based helmet detection system for safety compliance in oil and gas industry, in *5th Int Con Ind Eng Artif Intell (IEAI)*, IEEE, 2024.

8 Lin C, Zhang Y & Chen G, Intelligent detection of safety helmets and reflective vests based on deep learning, *J Real-Time Image Process*, **22(1)** (2025) 5.

9 Al Mokdad S, An overview of YOLO-based helmet detection: Modifications and advancements, in *8th Int Conf Imag, Signal Process Commun (ICISPC)*, IEEE, 2024.

10 Ji C, Hou Z & Dai W, A safety helmet detection algorithm based on improved YOLOv5s, In *2024 7th Int Conf Adv Algo Control Eng (ICAACE)*, IEEE, 2024.

11 Deng L, Zhou J & Liu Q, Helmet Net: An improved YOLOv8 algorithm for helmet wearing detection, *Int J Netw Distrib Comput*, (2024) 1–15.

12 Chowdhury Z U, Performance comparison of YOLO models for safety helmet detection: Insights from YOLOv5 to YOLOv10 with transfer learning, *Authorea Preprints*, (2024).

13 Sarmah R, Lahon P & Ahmed T, Enhanced precision in motorcycle helmet detection: YOLOv5 and Darknet approach, (2024).

14 Wei L, Research on helmet wearing detection method based on deep learning, *Sci Rep*, **14(1)** (2024) 7010.

15 Fan Z, LG-YOLOv8: A lightweight safety helmet detection algorithm combined with feature enhancement, *Appl Sci*, **14(22)** (2024) 10141.

16 Xu H & Wu Z, MCX-YOLOv5: Efficient helmet detection in complex power warehouse scenarios, *J Real-Time Image Process*, **21(2)** (2024) 27.

17 Zhang H, Cui Z & Su F, A coarse-to-fine two-stage helmet detection method for motorcyclists, *Proc IEEE/CVF Conf Comput Vis Pattern Recognit*, 2024.

18 Chai Z, Real-time automatic detection of motorcycle helmet based on improved YOLOv8 algorithm, in *6th Int Conf Commun, Inform Syst Comput Eng (CISCE)*, IEEE, 2024.

19 Nie Y & Xie X, A helmet detection algorithm based on improved Yolov8n, *J Phys Conf Ser*, **2858(1)**, IOP Publishing, 2024.

20 Nusari A N, Comparison of YOLO algorithms for vehicle accident detection and classification, in *4th Int Conf Emerg Smart Technol Appl (eSmarTA)*, IEEE, 2024.

21 Liu L, FH-DETR: A faster helmet detection transformer for small, complex, confusion two-wheeler helmet, (2024).

22 Ahmad H M & Rahimi A, SH17: A dataset for human safety and personal protective equipment detection in manufacturing industry, *arXiv preprint arXiv:2407.04590*, (2024).

23 Jia X, High-precision and lightweight model for rapid safety helmet detection, *Sensors*, **24(21)** (2024) 6985.

24 Ning S & Han X, An improved YOLOv8-based safety helmet wearing detection algorithm, in *7th Int Conf Comput Inform Sci Appl Technol (CISAT)*, IEEE, 2024.

25 Mercado Reyna J, Luna-García H, Espino-Salinas C H, Celaya-Padilla J M, Gamboa-Rosales H, Galván-Tejada J I, Galván-Tejada C E, Solís Robles R, Rondon D & Villalba-Condori K O, Detection of helmet use in motorcycle drivers using convolutional neural network, *Appl Sci*, (2023) **13(10)** 5882. https://doi.org/10.3390/app13105882.

26 Suma S, Anoop G L & Mithun B N, Helmet detection based on improved YOLO V8, *J Emerg Trends Novel Res*, (2024) **2(3)**. Available at: http://www.jetnr.org/

27 Adhikari H, Sahoo R, Khera S & Sharma N, Helmet detection on two-wheeler riders using machine learning, (2021). *Research Gate*, https://www.researchgate.net/publication/352771734_HELMET_DETECTION_ON_TWO-WHEELER_RIDERS_USING_MACHINE_LEARNING.

28 BlueBinaries, *Helmet detection in 2-wheeler*. Retrieved from https://www.bluebinaries.com/insights/thought-leadership/helmet-detection-in-2-wheeler/, (2024).

29 Shen S & Yang J, Better YOLO with attention-augmented network and enhanced generalization performance for safety helmet detection, *arXiv preprint arXiv:2405.02591*, (2024).

30 Dong Z & Zhang Q, Safety helmet detection: Adding attention mechanism to YOLOv8 to improve detection accuracy, in *7th Int Conf Artif Intell Big Data (ICAIBD)*, IEEE, 2024.