Adıyaman University
Journal of Engineering Sciences
Adyü J Eng Sci 2026;13:e260104
Adıyaman Üniversitesi
Mühendislik Bilimleri Dergisi
Adyü Müh Bil Derg 2026;13:e260104
https://doi.org/10.54365/adyumbd.1757010

Research Article / Araştırma Makalesi

Helmet Detection System with YOLO Architecture in Real-Time Construction Site Cameras

Gerçek Zamanlı Şantiye Kameralarından YOLO Mimarisi ile Otomatik Kask Algılama Sistemi

Abdil Karakan [1 *], Yüksel Oğuz [2]

[1]Afyon Kocatepe Üniversitesi, Dazkırı Meslek Yüksekokulu , Elektrik Bölümü, Afyonkarahisar, Türkiye
[2] Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi , Elektrik-Elektronik Mühendisliği Bölümü, Afyonkarahisar, Türkiye

Abstract

A significant proportion of construction site accidents occur due to falling objects from above, making helmet usage critical for worker safety. Helmet compliance is traditionally monitored manually by supervisors, a process susceptible to human oversight. Human-related limitations may result in overlooked safety violations. In this study, a real-time automatic helmet detection system was developed using images captured from construction site cameras. A dataset of 4,759 images was acquired under diverse viewing angles, lighting conditions, and distances. Through data augmentation techniques such as grayscale conversion, blurring, noise addition, brightness and saturation adjustment, perspective transformation, resizing, and shifting, the dataset was augmented to a total of 17,781 samples through systematic data augmentation, ensuring greater adaptability to real-world conditions. The dataset was evaluated across four different versions of the YOLO architecture, with YOLOv8 achieving the best performance. The results demonstrated an accuracy of 98.3%, an F1-score of 0.939, and a mAP@.5 value of 0.977. Furthermore, the system improved the frame rate from 27 FPS to 60 FPS, enabling efficient real-time detection.

Anahtar Kelimeler: *Kask tespiti, Kişisel koruma ekipmanı, Derin öğrenme, Nesne tespiti, YOLO mimarisi*

Öz

İnşaat alanlarında iş kazalarının önemli bir kısmı yukarıdan düşen nesnelerden kaynaklanmaktadır. Bu nedenle kask kullanımı, çalışanların güvenliği için kritik öneme sahiptir. Ancak kask kontrolü genellikle ekip başları tarafından gözlem yoluyla yapılmakta ve insan hatasından dolayı eksiklikler yaşanmaktadır. Bu çalışmada, şantiye kameralarından elde edilen görüntülerle gerçek zamanlı otomatik kask tespiti gerçekleştirilmiştir. Gerçek ortamdan 4.759 görüntü toplanmış, farklı açılar, mesafeler ve ışık koşulları göz önünde bulundurulmuştur. Veri artırma yöntemleri (gri tonlama, bulanıklaştırma, gürültü ekleme, parlaklık ve renk doygunluğu değişimi, perspektif dönüşümü, yeniden boyutlandırma vb.) uygulanarak veri sayısı 17.781'e çıkarılmıştır. Böylece veri seti gerçek ortama uyarlanmıştır. Elde edilen veri seti YOLO'nun dört farklı sürümünde test edilmiş, en yüksek başarı YOLOv8 ile sağlanmıştır. Bu modelde doğruluk %98.3, F1-Skoru 0.939 ve mAP@.5 değeri 0.977 olarak bulunmuş, FPS değeri 27'den 60'a yükselmiştir.

Keywords: *Helmet detection, Personal Protective Equipment, Deep learning, Object detection, YOLO architecture*

* Corresponding e-mail (Sorumlu yazar e-posta): akarakan@aku.edu.tr
Received (Geliş Tarihi):02.08.2025, **Accepted (Kabul Tarihi):** 08.01.2026

1. Introductions

Ensuring occupational safety remains a primary concern in the construction industry. Protecting workers from potential physical hazards is a fundamental aspect of occupational safety. Safety helmets constitute a critical component of personal protective equipment for those working in the construction industry. Construction helmets protect the head from many dangers, such as impact, falling objects, and electric shock. They additionally provide limited protection against environmental particulates. Due to the nature of construction sites, there are always risks, such as falling from heights, falling objects, or hitting hard surfaces. Accordingly, helmet use plays a vital role in mitigating injury risks on construction sites. A large-scale study conducted in China reported that, 67% of accidents in construction sites occurred due to not wearing a helmet [1].

In the literature, there are studies on many different methods of helmet detection in construction sites [2,3]. Kelm and colleagues used a mobile Radio Frequency Identification (RFID) card for helmet detection. An RFID tag was embedded inside the helmet to verify compliance. Entrance to the construction site is from a single point and through a turnstile. RFID card readers have been placed at the entrance point to the construction site. In this way, the system ensured helmet verification at the entry point of the construction site. [4]. Barro-Torres and his friends developed the helmet method with an RFID card. They placed an RFID card, RFID card reader, ZigBee transmitter, and a speaker on the construction worker. When the helmet is not worn, the signal received from the RFID card reader is transferred to the center via a ZigBee transmitter. The received signals are processed by a central control unit. The result is transferred again via the ZigBee transmitter. Then, information is given via loudspeaker [5]. RFID-based systems require workers to carry additional hard ware, increasing operational burden. These studies only check whether people are wearing helmets at the entrance of construction sites. These systems do not guarantee continuous compliance monitoring. Additionally, the RFID card placed inside the helmet is controlled. It does not give any warning when the workers remove the RFID card from the helmet and insert it elsewhere. These methods are costly and lack the capability for continuous monitoring.

Rubiyat and colleagues used machine learning to detect helmets. They did this in two different ways. They first collected images for machine learning. Helmet detection was carried out on the images. In his first work, he used Oriented Gradient Histogram (HOG). In this study, he achieved an accuracy of 81.13%. In his later work, he used Circle Hough Transform (CHT) feature extraction techniques. By combining the two studies, the accuracy rate increased to 91.93% [6]. Lui and his colleagues worked on helmet detection in motorcycle riders. They used machine learning in their study. In their study, they achieved an accuracy of 72.1% [7]. Machine learning-based methods generally yielded moderate accuracy levels. At the same time, the studies were only on the image. These approaches did not support real-time detection.

Wu and Zhao employed convolutional neural networks (CNN) for helmet detection. First, it scans whether there are people in the photo. When a human is detected, a secondary module identifies the head region. After detecting the head area, it is determined whether there is a helmet or not. In their study, they achieved an accuracy of 98.8% [8]. The accuracy rates of studies conducted with machine learning have not increased to very high levels. At the same time, the detection process incurs substantial latency. For real-time detection, the frame rate per second (FPS) must be very high. Otherwise, the image disappears from the screen before detection can be done. For this reason, consequently, detection fails. Since machine learning processes are done one by one, As a result, FPS performance remains insufficient for real-time deployment.

Fangbo and his colleagues used the YOLOv5 architecture, a deep learning network, for helmet detection. They prepared a data set consisting of 6045 photographs for helmet detection. As a result of the study, the highest accuracy of 94.7% was achieved [9]. Fan and his colleagues used YOLOv3 and YOLO-Dense backbone architectures in their studies on caste detection. They detected helmets in two different studies. In their study, they achieved an accuracy of 95.15% with YOLOv3 and 97.59% with YOLO-Dense backbone

architecture. Incorporating the YOLO-Dense backbone yielded a 2.44% improvement in accuracy [10]. Madhuchhanda and his friends worked on determining whether a helmet was worn on a motorcycle. They used YOLOv3 architecture in their work. As a result of their study, an accuracy of 96.23% was achieved [11]. Wei and colleagues used YOLOv5 architecture for helmet detection. They preferred to use the ready data set as the data set. They enhanced efficiency by integrating soft-NMS into the YOLOv5 pipeline instead of NMS in the YOLOv5 architecture. As a result of their studies, they increased the accuracy rate to 97.7% [12]. Shilei and colleagues have made fundamental improvements to the YOLOv5 architecture for helmet detection. In the YOLOv5 architecture, they added functionality that tries to find small targets such as helmets. They also used DloU-NMS instead of NMS in the YOLOv5 architecture. An accuracy of 92.12% was achieved in the YOLOv5 architecture. In the improved YOLOv5 architecture, the accuracy rate was increased by 3.56% to 95.68% [13]. Rui and his colleagues investigated the effects of data preparation on accuracy in helmet detection. They used the Gaussian fuzzy method in the data set. Thanks to this method, the accuracy rate improved by 0.01-0.02 [14].

Additional YOLO-based studies in different application domains further demonstrate the versatility and robustness of the algorithm. Bekçioğulları compared multiple YOLO versions for traffic sign detection, highlighting performance differences across architectures [15]. Altay and Yılmaz used the YOLO algorithm for T-cell classification, showing its adaptability to biomedical imaging [16]. Farah applied YOLOv3 and YOLOv4 for ASD detection [17]. Several studies focused specifically on safety applications: Savaram conducted a comparative analysis of YOLO models for real-time helmet detection on Raspberry Pi, while Buradkar proposed a YOLO-based helmet detection framework for worker safety [18,19]. These studies further support the expanding role of YOLO architectures in real-time object detection and occupational safety applications.

In the study, helmets were detected in real-time with images taken from the camera at the construction site. A data set resembling real-word conditions has been prepared. For the dataset, 4759 helmet images were taken from the real environment and the internet. These images were taken from different locations, different lighting levels, and different distances. The images have also been augmented with grayscale conversion, tilt adjustments, blurring, variability enhancement, noise addition, brightness modification, changes in color vividness, perspective transformation, resizing, and positional changes. In this way, the data set has been made more suitable for the real environment. In the study, 4 different versions of YOLO architecture were used. In the study, YOLOv8 architecture achieved a 98.3% accuracy rate. Since the study carried out will make real-time detection, the frame rate per second was also studied. Since the image on the camera changes very quickly, the a sufficiently high FPS is essential for real-time deployment. When the FPS rate is low, the image changes on the screen before it can be detected. For this purpose, efforts were made to increase FPS rates in four different YOLO architectures. As a result of the study, the lowest speed was 27 FPS, and the highest speed was 60 FPS. At this speed, it is sufficient for real-time image processing. By combining real data usage, comprehensive data augmentation processes, multiple YOLO version comparisons, and the resulting improvements in accuracy and FPS, we provide a holistic, application-focused contribution to the literature. This study distinguishes itself from existing work by presenting a new dataset and presenting an approach that optimizes real-time performance.

2. Materials and Methods

Helmet real-time detection with the YOLO algorithm was carried out in three steps. In the first step, sub-processes related to the preparation of the data set were carried out. These procedures include data preparation, pre-processing, training data and test data processes. The second step is to create and adjust the convolutional neural network to be used for deep learning. These transactions; Creating a deep learning neural network model, adjusting deep learning neural network parameters, and uploading training and test data to the system. The final phase involves executing the trained model for inference. In this step, the model is trained, the model undergoes iterative retraining based on performance feedback,

and finally, the final performance metrics are computed. Figure 1 illustrates the overall operational framework of the system of real-time detection of the helmet with the YOLO algorithm.

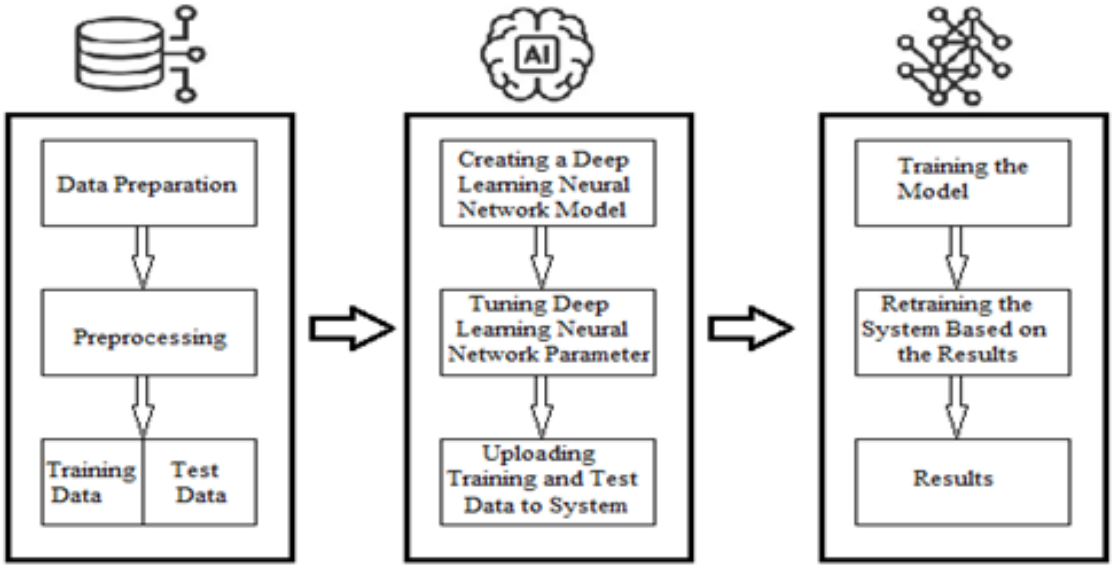


Figure 1. General working principle of real-time detection of the helmet with the YOLO algorithm.

2.1. Data Set

Safety helmets are among the most essential forms of personal protective equipment (PPE) used on construction sites, primarily to safeguard individuals from potential hazards such as falling objects. These helmets provide a high level of protection, particularly in incidents involving items falling from heights. In this study, a custom dataset was developed with the aim of distinguishing between individuals wearing helmets and those who are not. The dataset was constructed using photographs collected from both real-world environments and various online sources. To build a comprehensive image database, videos obtained from different construction sites and a range of internet platforms were processed frame by frame.

Initially, the dataset consisted of 4759 images. However, in order to enhance the quantity and diversity of the data, as well as to improve the generalization capabilities of machine learning models, a variety of data augmentation techniques were applied. These techniques included grayscale conversion, gradient adjustments, blurring, variability introduction, noise addition, brightness modification, colour saturation changes, perspective distortion, image resizing, and positional transformations. Through the application of these augmentations, the dataset was significantly expanded to a total of 17781 images.

The dataset comprises photographs featuring both helmeted and non-helmeted individuals. Each image contains a varying number of annotations depending on the number of subjects present. Some images are annotated with only one label, while others contain up to ten labels. In total, 7259 labels were generated across the dataset. Representative samples from the dataset are presented in Figure 2, illustrating the diversity and realism of the images used. These examples were carefully selected to support the training and evaluation processes of the model, ensuring the dataset's relevance and applicability in real-world scenarios.

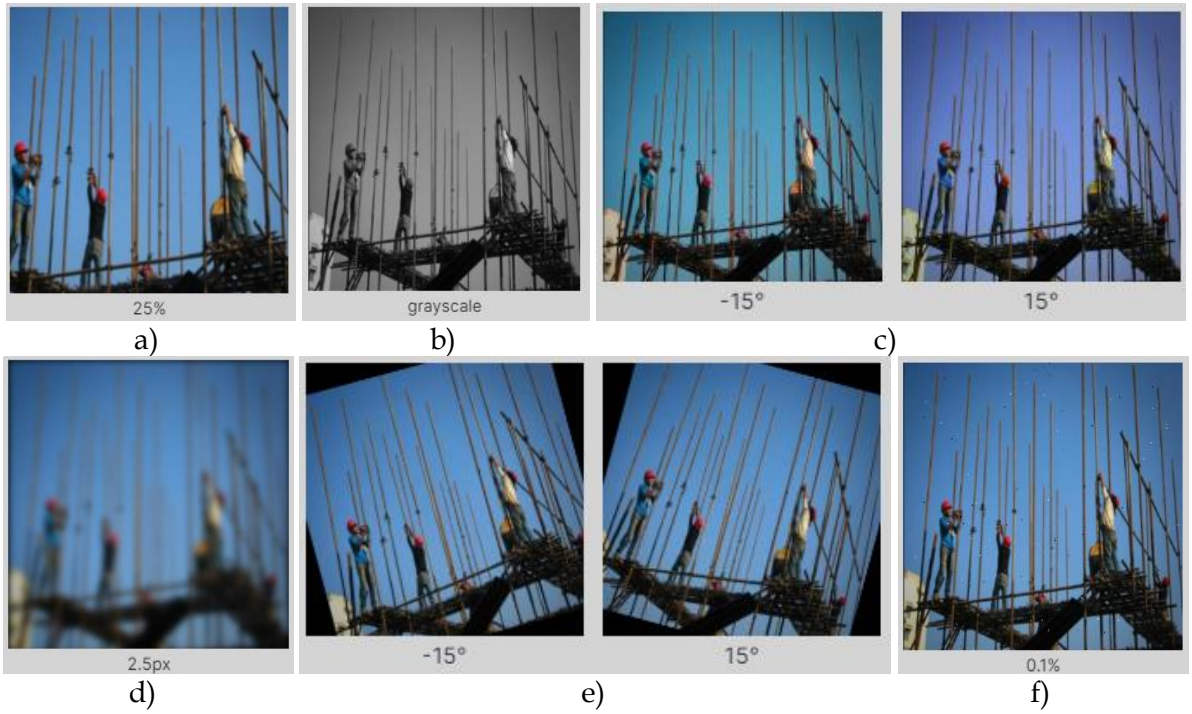
The dataset contains an unequal distribution of classes. Among the total 7,259 annotations, 3705 belong to the helmet class, while 3554 correspond to the head class. Due to this imbalance, data augmentation techniques were applied to improve class representation and enhance the model’s generalization capability.



Figure 2. Some sample photos from the dataset.

The helmet is not only available in yellow. White, yellow, red, orange, blue, and green helmets are used in construction sites. Each colour is used by different users. For example, while managers and engineers use white helmets, workers use yellow helmets. For these reasons, samples representing each helmet color were included while preparing the data set.

Extensive image transformations were applied that make up the data set, increasing its compatibility with the real environment. Figure 3 shows the changes made to the samples used in the data set.



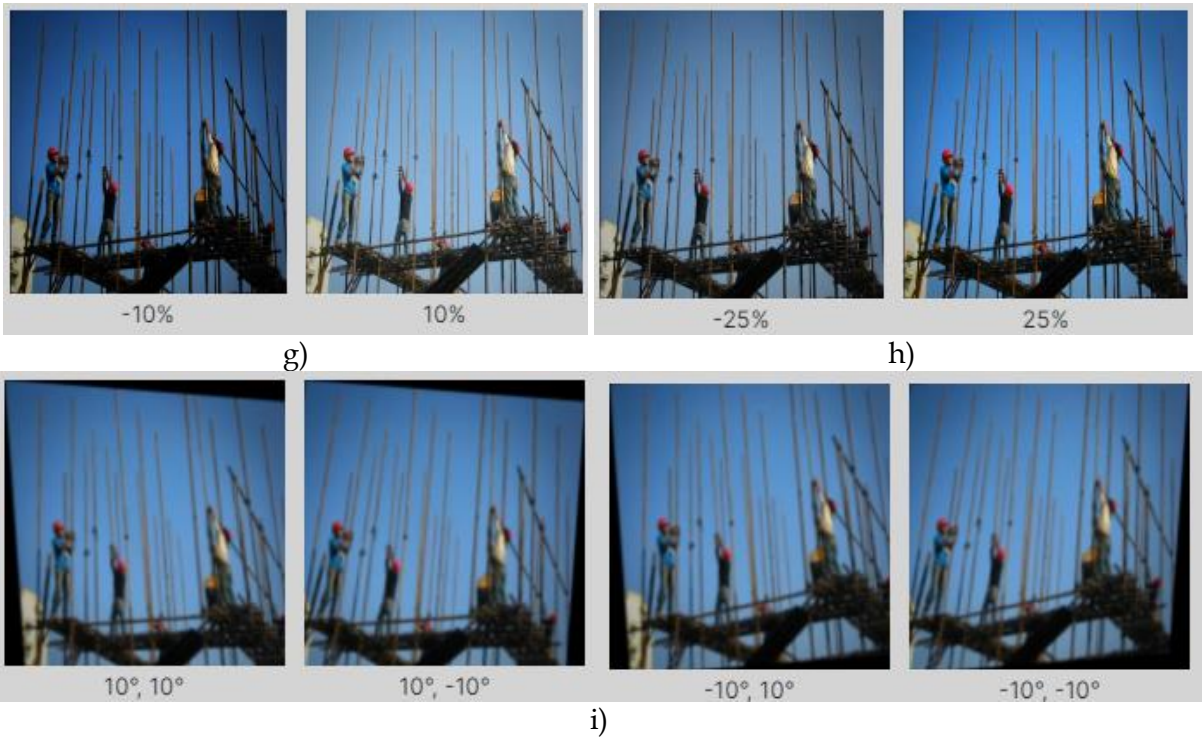


Figure 3. a) Resizing and repositioning, **b)** Grayscale transformation, **c)** Rotation with angle variation, **d)** Gaussian blur application, **e)** Rotational randomness, **f)** Noise injection, **g)** Brightness adjustment, **h)** Color saturation alteration, **i)** Perspective distortion

In Figure 3a, a 25% variation was applied to the size and placement of the images to enhance the model's tolerance to differences in camera position. Figure 3b illustrates the conversion of dataset images into grayscale format. In Figure 3c, each image was rotated by ±15° to simulate viewpoint variation to simulate angular distortions. In Figure 3d, Gaussian blur was applied at random to simulate the effect of camera defocus. Figure 3e introduces ±15% variation in image rotation to increase robustness against camera tilting. Figure 3f shows the addition of artificial noise to counteract potential sensor or transmission artifacts. Figure 3g represents the manipulation of brightness levels by ±15%, aimed at increasing resilience to changes in lighting and exposure. In Figure 3h, the vibrancy of colors was randomly modified to account for variations in image saturation. Finally, Figure 3i displays perspective alterations, which help improve the model's performance under camera angle changes and lens distortions. These augmentation strategies significantly contribute to reducing real-time detection errors in helmet recognition tasks.

The dataset used in this study was composed of images recorded in real construction sites with the necessary permissions, as well as images obtained from open-access sources on the internet. No personal data was stored within the scope of the study, and care was taken to prevent facial identification information from being visible. Images sourced from the internet were used for academic purposes only, and no copyrighted material was redistributed. All images were anonymized in accordance with privacy policies and research ethics guidelines were adhered to.

2.2. YOLO Architecture

YOLO reframes object detection as a single regression problem, streamlining the process significantly. Unlike conventional methods such as Fast R-CNN, which depend on proposing multiple candidate regions, YOLO processes the entire image in a single pass. It simultaneously predicts bounding box coordinates and class probabilities for all objects present. This unified approach eliminates the need for separate region proposals and class-specific processing, resulting in a much faster detection pipeline. As the model examines the image just once, it can quickly determine what objects are present and where they are located. Figure 4 provides a visual overview of the YOLO architecture's operational structure.

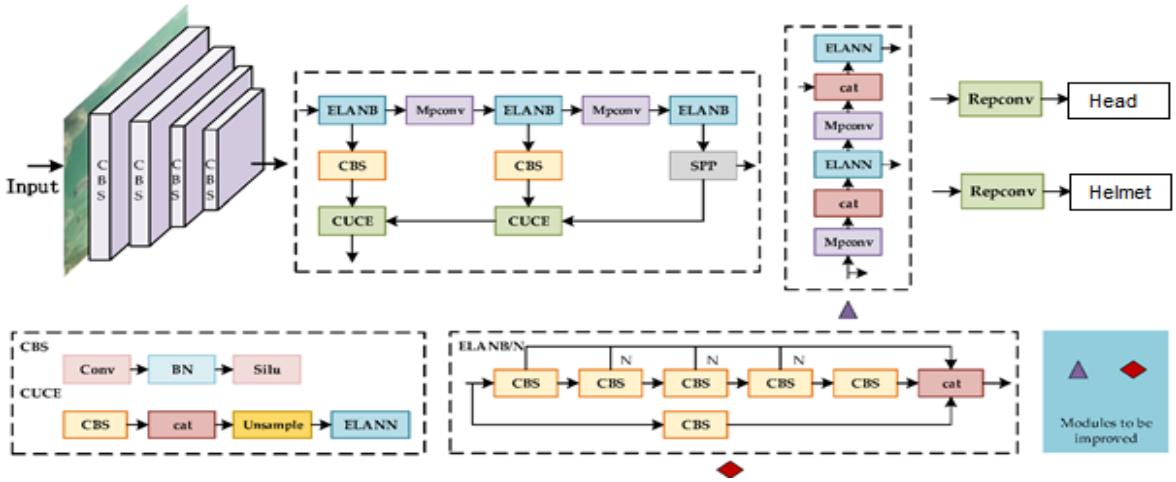


Figure 4. YOLO architecture [20].

YOLOv5 is part of the YOLO series of models used in computer vision, specifically for object detection. It is available in four primary versions: small (s), medium (m), large (l), and extra-large (x), each offering increased accuracy and longer training times. In the corresponding graph, the objective is to develop an object detection model with high performance (Y-axis) while considering inference time (X-axis). Initial findings suggest that YOLOv5 achieves strong results when compared to other modern detection algorithms. Its various versions operate faster than EfficientDet models. Among them, YOLOv5x delivers the highest accuracy and can analyze images significantly quicker than the EfficientDet D4 model while maintaining similar precision. Although YOLOv5 gains much of its enhanced performance from PyTorch's training framework, its underlying model architecture remains closely aligned with that of YOLOv4.

The YOLOv6 model was released by Meituan in June 2022. It gave good results in the Common Objects in Context (COCO) dataset comparison. The YOLOv6 model is built on the basis of the YOLO architecture and offers various improvements and new methods over other models of the YOLO family. YOLOv6 is written in PyTorch. YOLOv6 has three major updates: a hardware-friendly spine, neck design, and dedicated head for efficiency and effective training strategies. YOLOv6's object detection performance has been shown to be comparable to other CNN-based algorithms, with improvements in both speed and accuracy with incremental versions of the algorithm. In the study, the effect of different image sizes on the performance of YOLOv6 sub-architectures is shown. The data set used for the YOLOv6 model in the study was derived into two versions with dimensions of 416x416 and 640x640.

YOLOv7 introduces substantial architectural improvements enhancing both accuracy and efficiency in the field of real-time object detection, offering a refined architecture that balances speed and accuracy more effectively than many of its predecessors. Developed as an extension of the original YOLO family, YOLOv7 introduces several novel strategies for performance optimization, including extended efficient layer aggregation networks (E-ELAN) and re-parameterization techniques that enhance inference efficiency without compromising detection accuracy. This model has been meticulously engineered to maintain high precision across various detection tasks, while still achieving real-time performance even on resource-constrained devices. One of the defining features of YOLOv7 is its ability to handle multi-scale object detection tasks with improved spatial consistency, making it particularly well-suited for complex environments such as construction sites, urban landscapes, and autonomous driving scenarios. Additionally, YOLOv7 integrates modules that support auxiliary head training, dynamic label assignment, and model scaling, which collectively contribute to more stable convergence during training and increased generalization across datasets. Its open-source availability and compatibility with a wide range of deep learning frameworks have further accelerated its adoption in both academic research and industrial applications. Overall, YOLOv7 exemplifies the evolution of real-time object detection models, providing a robust foundation for applications requiring both efficiency and high detection precision.

YOLOv8, the latest iteration in the "You Only Look Once" series of object detection models, introduces a range of architectural and functional enhancements that solidify its position at the forefront of real-time computer vision solutions. Unlike its predecessors, YOLOv8 is developed independently by Ultralytics, representing a comprehensive redesign rather than a simple extension of earlier versions. One of the most notable features of YOLOv8 is its modular architecture, which enables seamless integration of object detection, instance segmentation, pose estimation, and classification tasks within a unified framework. This level of versatility allows the model to be applied across a broad spectrum of computer vision applications with minimal reconfiguration. Moreover, YOLOv8 adopts an anchor-free detection paradigm, which eliminates the dependence on predefined anchor boxes and simplifies the model's training dynamics. By doing so, it enhances detection performance, particularly for objects of varying scales and irregular shapes. The model also incorporates improved loss functions, refined non-maximum suppression (NMS) strategies, and adaptive image pre-processing techniques that collectively contribute to superior accuracy and efficiency. With an emphasis on user accessibility, YOLOv8 is designed to work out-of-the-box with popular deep learning libraries such as PyTorch, and it includes pre-trained weights and intuitive APIs that facilitate rapid deployment. Its superior balance between speed, accuracy, and flexibility makes YOLOv8 a compelling choice for academic research and practical implementations alike, particularly in domains requiring real-time analysis of dynamic and complex scenes.

During the training phase, all YOLO architectures were optimized using the Adam optimizer with a learning rate of 0.001. A batch size of 16 and 200 epochs were used as the standard configuration across all experiments. Early stopping criteria were employed with a patience of 30 epochs to prevent overfitting. Input images were resized to 640×640 resolution, and all models were trained using the same augmentation strategies to ensure fairness and consistency in performance comparison. Table 1 shows the training parameters used in YOLO models.

Table 1. Training parameters used in the YOLO models

Parameter	Value
Batch Size	16
Number of Epochs	200
Optimizer	Adam
Learning Rate	0.001
Image Size	640 × 640

3. Result and Discussion

In the dataset, 70% was used for training, 20% for testing, and 10% for validation. The computer used in the study has an AMD Ryzen 1500X 3.5 GHz processor. The graphics card is NVIDIA GeForce GTX 1050 Ti 4GB GDDR5. Memory speed is 16 GB 3000 MHz. Firstly, YOLOv8 dataset architecture was used in the study. In the YOLOv8 architecture, two determinations were made: helmet and head. The training period for detection lasted 32 hours, 14 minutes, and 35 seconds. As a result of the study, the highest accuracy rate was 97.4% in head detection, and 98.9% in helmet detection. The mean accuracy achieved was 98.3%. The results occurring in the YOLOv8 architecture are given in Table 2.

Table 2. Results occurring in YOLOv8 architecture.

Class	Precision	Recall	F1-Score	mAP@0.5	mAP@.5:.95
Helmet	0.992	0.894	0.944	0.984	0.867
Head	0.974	0.913	0.942	0.980	0.721
All	0.983	0.903	0.943	0.982	0.794

The average performance metrics of the YOLOv8 architecture are as follows: Precision 98.3%, Recall 90.5%, F1-Score 0.943, mAP@0.5 0.982, and mAP@0.5:0.95 0.794. In the context of helmet detection, the Precision was determined to be 99.2%, Recall 89.4%, F1-Score 0.944, mAP@0.5 0.984, and mAP@0.5:0,95 0.719. For

head detection, the Precision was found to be 97.4%, Recall 91.3%, F1-Score 0.942, mAP@0.5 0.980, and mAP@0.5:0.95 0.721. The results of the YOLOv8 architecture are represented through a confusion matrix, which consists of predicted values and actual values.

These models have been evaluated using Although the YOLOv8 model achieved the highest precision among all architectures, its recall value (89.4%) indicates that some positive helmet instances may still remain undetected. In the context of real-time occupational safety systems, this may pose a potential risk, as missing a helmet violation could lead to safety hazards. The lower recall is mainly attributed to challenging visual factors such as occlusions, small helmet size, or distant viewpoints. To mitigate this issue, future work may incorporate expanded training data, small-object enhancement modules, or hybrid sensor-based detection approaches, which would help improve recall performance in safety-critical applications. Figure 5 presents both the ROC curve and the confusion matrix.

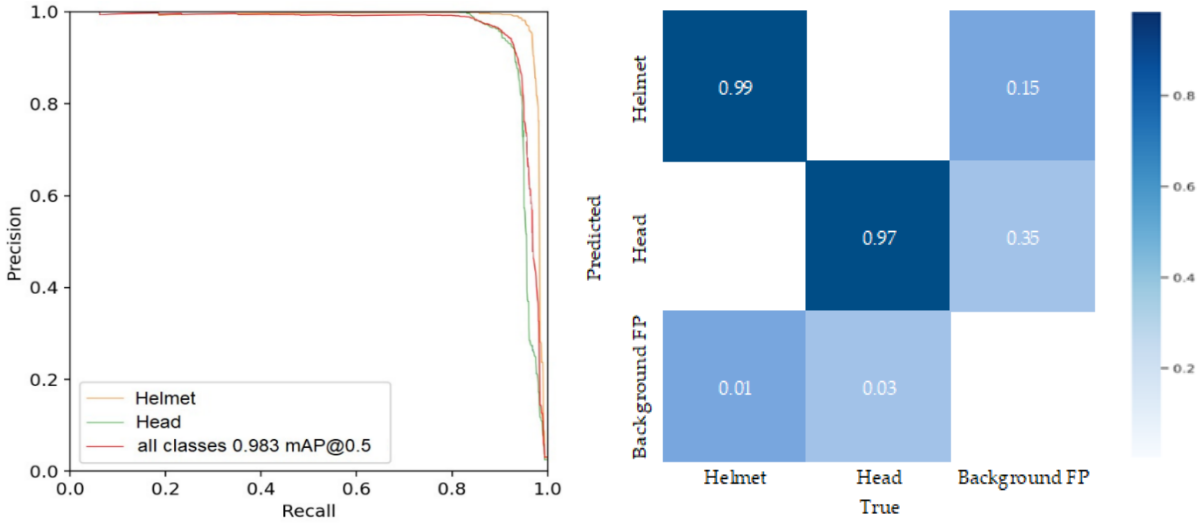


Figure 5. ROC curve and complexity matrix resulting from YOLOv8 architecture.

In this study, along with the YOLOv8 model, alternative versions such as YOLOv7, YOLOv6, and YOLOv5 were also implemented to evaluate performance across different architectures. Each model was trained and tested using the same dataset to ensure consistency and allow for a fair comparison. This multi-architecture approach aimed to identify the most accurate and reliable model while reducing potential error rates. Since each YOLO version incorporates distinct algorithmic structures and optimization techniques, their detection outcomes vary accordingly. The average performance metrics obtained from the YOLOv5, YOLOv6, and YOLOv7 models are summarized in Table 3.

Table 3. Average results of YOLOv5, YOLOv6, YOLOv7 and YOLOv8 architecture.

Class	Precision	FPS
YOLOv5	0.936	27
YOLOv6	0.950	42
YOLOv7	0.906	55
YOLOv8	0.983	60

In the scope of the study, a comparative performance evaluation was conducted on multiple YOLO architectures to determine their suitability for real-time helmet detection tasks. Among the models tested, the YOLOv8 architecture demonstrated the highest classification accuracy, reaching an impressive 98.3%, indicating its strong potential for precise and reliable object detection in dynamic environments. In contrast, the YOLOv7 architecture yielded the lowest accuracy, with a performance of 90.6%, suggesting limitations in its ability to generalize effectively across the dataset used. The YOLOv5 architecture achieved a moderate accuracy of 93.6%, positioning it between the two aforementioned models. Notably,

both YOLOv5 and YOLOv7 shared identical accuracy rates of 90.6% at certain points in the evaluation, further emphasizing the superiority of YOLOv8 in this specific application.

Given that the intended application of the system involves real-time detection, the frame-per-second performance is a critical parameter. High FPS values are essential to ensure that the system can process and respond to visual input in real time without latency. In this context, the study assessed the FPS performance across the four YOLO variants, with observed values ranging from 27 FPS to 60 FPS. The YOLOv8 architecture, which not only delivered the highest accuracy but also achieved up to 60 FPS, emerged as the most balanced and efficient model. This level of performance is well within the acceptable range for real-time detection applications, making YOLOv8 particularly suitable for deployment in fast-paced, real-world construction site environments.

YOLOv8 demonstrated superior performance in this study, and its advantages over YOLOv5, YOLOv6, and YOLOv7 can be summarized as follows: Higher Detection Accuracy: YOLOv8 achieved the highest precision (98.3%), outperforming all other tested models.

Anchor-Free Architecture: Eliminates dependence on predefined anchor boxes, improving detection of small and irregular objects such as construction helmets.
Improved Generalization: Enhanced loss functions and dynamic label assignment result in better stability and lower error rates during training.
Greater Real-Time Efficiency: Achieved 60 FPS, making it highly suitable for fast-changing construction site environments.
Enhanced Feature Extraction: The redesigned backbone and neck structure improve the model's ability to detect objects at multiple scales.
Versatility: Supports detection, segmentation, and pose estimation within the same framework, enabling future extensions of this study.
Lightweight Deployment: Requires fewer computational resources than previous high-performance YOLO versions, enabling broader real-time application support.

Furthermore, to ensure the robustness and adaptability of the system under diverse conditions, the detection model was tested at different times of day and in varying environmental settings, thereby simulating realistic scenarios. This approach aimed to validate the model's consistency and effectiveness across a wide spectrum of operational contexts. Representative results from these evaluations are presented in Figure 6, which illustrates nine distinct detection instances performed during the study. These examples collectively demonstrate the model's capability to maintain high accuracy and speed across different use cases, reinforcing its applicability for real-time safety monitoring systems.

When the existing literature on helmet detection is thoroughly examined, it becomes evident that a substantial number of studies have been conducted in this field over the years. These studies have employed a wide range of methodologies, algorithms, and deep learning architectures to detect the presence or absence of safety helmets, particularly in construction and industrial environments where occupational safety is a primary concern. The diversity of approaches reflects ongoing efforts to improve the accuracy, speed, and reliability of helmet detection systems under varying environmental conditions and application scenarios. While some studies have utilized traditional machine learning techniques and handcrafted features, more recent research has shifted toward deep learning-based solutions especially convolutional neural networks and advanced object detection models such as the YOLO family. These models have been evaluated using different datasets, evaluation metrics, and deployment contexts, making direct comparisons between studies complex yet insightful. Table 4 presents a comparative overview of selected literature in the domain of helmet detection, summarizing the methodologies applied, the architectures used, and the reported performance metrics. This table serves as a basis for benchmarking the proposed model in the current study, highlighting its strengths and contributions relative to existing work.

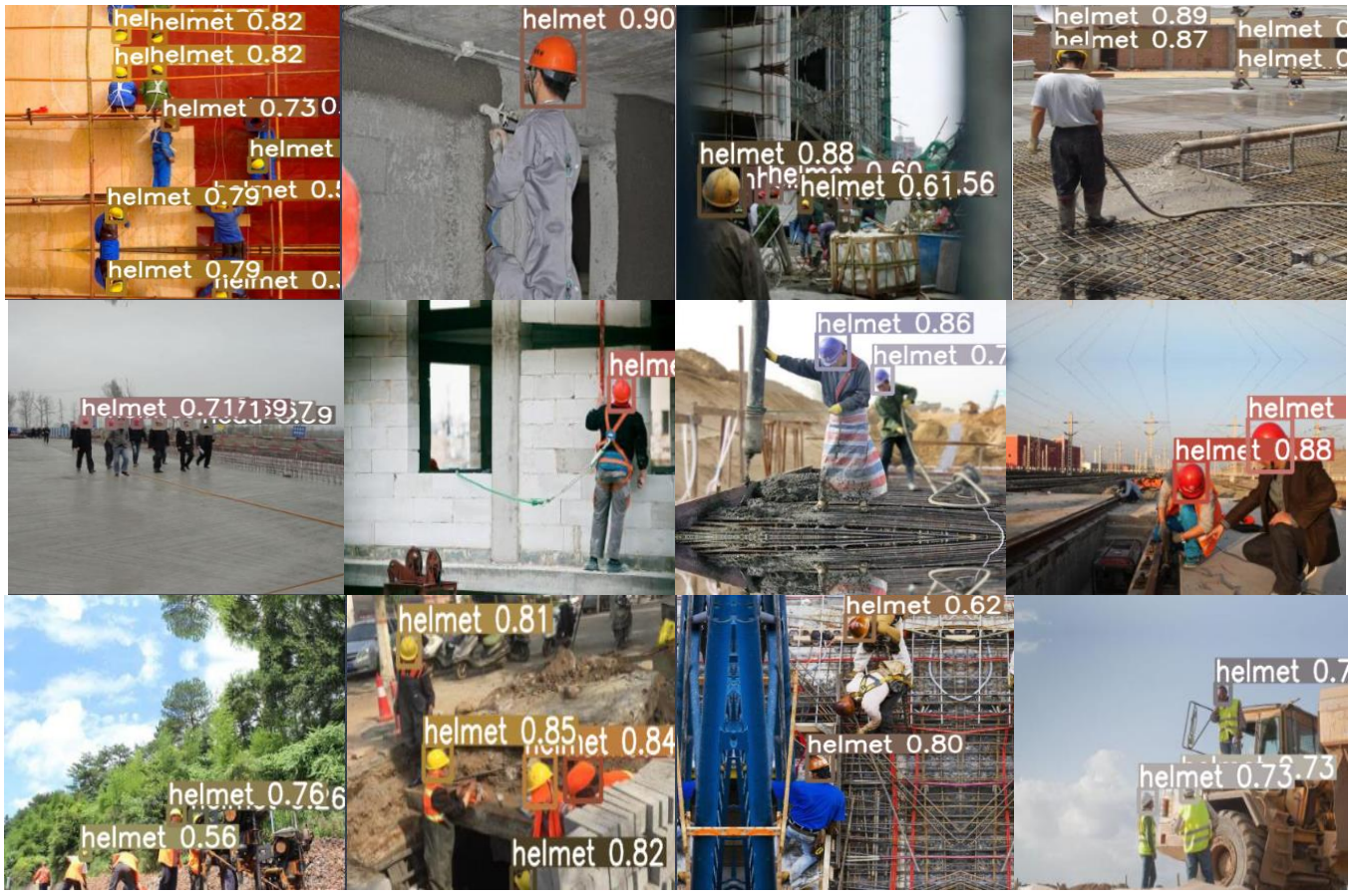


Figure 6. Detection of different helmets.

Table 4. Literature studies.

Author	Model	Precision
Huand et al., [21]	YOLOv3, Improved YOLOv3, Faster R-CNN and SSD	0.931
Li et al., [22]	SSD	0.36
Kamboj & Powar [23]	SSD	0.96
Long et al., [24]	SSD	0.685
Zhou et al., [25]	YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x	0.947
Tan et al., [26]	YOLOv5 and Improved YOLOv5	0.957
Yung et al., [27]	YOLOv5s, YOLOv6s and YOLOv7	0.896
Korkmaz & Ağdaş, [28]	YOLOv8	0.969
Türkdamar et al., [29]	YOLOv4, YOLOv5 and Faster R-CNN	0.98
Wu et al., [8]	YOLOv3 and YOLO-Dense backbone	0.975
Redmon et al., [30]	YOLOv3	0.964
Khan, [31]	YOLOv4	0.97
Yung et al., [32]	YOLOv5s, YOLOv5m and YOLOv5l	0.983
Yange et al., [33]	YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x	0.936
Kim et al., [34]	YOLOv7-Tiny + Mobile Net	0.968
Sharma et al., [35]	Improved YOLOv8 + CBAM	0.981
Zhang and Lui [36]	YOLOv5 + GhsotNet	0.957
Park et al., [37]	YOLOv6 with AugMix	0.964
Ahmed et al.,[38]	YOLOv8 Transformer Decoder	0.974
This study	YOLOv5, YOLOv6, YOLOv7, and YOLOv8	0.983

The data sets and algorithms used in literature studies are different. For these reasons, the accuracy rates obtained as a result of the studies are different. In literature studies, architectures from YOLOv3 to YOLOv8 have been used in many studies. There is an increase in accuracy rates in studies from YOLOv3 to YOLOv8 architecture. The highest accuracy rate was in the YOLOv8 architecture. It is understood that the other results of the YOLOv8 architecture, especially the accuracy rates, are better than other models of the YOLO architecture.

The literature review reveals that earlier systems face significant challenges in practical applications, primarily due to their low frame-per-second performance, which limits their usability in real-time scenarios. To address this limitation, the study employed the four most recent versions of the YOLO architecture. This comparative approach aimed to identify the model offering both the highest accuracy and optimal processing speed. Thanks to YOLO's capability to analyze the entire image in a single pass, it delivers superior FPS performance. Notably, the YOLOv8 model achieved a processing speed of 60 FPS, making it highly suitable for real-time detection tasks.

A data set needs to be prepared to determine whether a person is wearing a helmet. The larger and more diverse this data set is, the higher the rate of helmet detection is achieved. There is an advantage in helmet detection. Helmets are of different colours. The data set prepared in the study includes red, blue, yellow and white helmets. The fact that the helmets are coloured affects the results positively on one hand and negatively on the other. A notable limitation is that when the person's clothing and the helmet are the same colour, the detection process becomes difficult. In addition, the FPS speed was increased in the study, making it suitable for real-time detection.

4. Conclusion

In this study, a real-time helmet detection system was developed and evaluated using four different YOLO architectures: YOLOv5, YOLOv6, YOLOv7, and YOLOv8. A dataset constructed from real construction site environments, supported by extensive data augmentation, enabled a consistent and controlled comparison across all models. Among the examined architectures, YOLOv8 achieved the highest overall performance with 98.3% accuracy and a processing speed of 60 FPS, demonstrating strong suitability for real-time occupational safety applications.

The unique contribution of this study is the systematic comparison of multiple YOLO versions with a dataset obtained from real environments, optimizing the FPS–accuracy tradeoff for real-time applications. The combined use of real data, comprehensive data augmentation methods, and performance analysis of four different YOLO architectures provide a holistic, application-focused contribution to the literature. This integrated approach allows for a clearer understanding of how different YOLO versions behave under identical conditions and which configurations are most effective in dynamic construction settings.

A comparative evaluation with existing studies shows that YOLOv8 not only provides superior precision but also offers a more favorable accuracy–inference speed balance, which is critical for real-time detection systems. However, its recall value indicates that certain helmet instances may still be missed, especially in challenging visual conditions such as occlusions or distant viewpoints. This limitation highlights the potential benefits of expanding the dataset, incorporating techniques designed for small-object enhancement, or employing hybrid sensor-assisted frameworks in future research.

Demonstrate that the proposed system is capable of achieving reliable and efficient real-time helmet detection in complex construction environments. The findings offer practical insights for future developments in automated safety monitoring and contribute to the ongoing advancement of computer vision applications in occupational safety.

Contribution Statement

Research design, literature review and evaluation, data collection, data analysis, validation of data and analysis: Abdil Karakan, interpretation of findings, writing of the article: Yüksel Oğuz.

Conflict of Interset

The authors of the article declare that they have no personal or financial conflict of interest with any institution, organization or person

References

[1] Chang X, Liu M. Fault tree analysis of unreasonably wearing helmets for builders. J Jilin Jianzhu Univ 2018;35:67–71. https://doi.org/10.1088/1742-6596/1684/1/012013
[2] Goodrum PM, McLaren MA, Durfee A. The application of active radio frequency identification technology for tool tracking on construction job sites. Autom Constr 2006;15(3):292–302. https://doi.org/10.1016/j.autcon.2005.06.004
[3] Jaselskis E, Haas CT, Goodrum PM. Construction transportation-related RFID research and applications. In: Transportation Research Circular: RFID Transportation Applications; 2007:19–24. https://doi.org/10.1007/978-3-031-36922-3_18
[4] Kelm A, Laußat L, Meins-Becker A, et al. Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of personal protective equipment (PPE) on construction sites. In: Autom Constr; 2013;36:38–52. https://doi.org/10.1016/j.autcon.2013.08.009
[5] Barro-Torres S, Fernandez-Carames TM, Perez-Iglesias HJ, Escudero CJ. Real-time personal protective equipment monitoring system. Comput Commun 2012;36:42–50. https://doi.org/10.1016/j.comcom.2012.01.005
[6] Rubaiyat AHMT, Toma TT, Kalantari-Khandani M, et al. Automatic detection of helmet uses for construction safety. In: Proc of the 2016 IEEE ACM Int Conf on Web Intelligence Workshops; 2016:25–27. https://doi.org/10.1109/WIW.2016.045
[7] Liu CC, Liao JS, Chen WY, Chen JH. The full motorcycle helmet detection scheme using canny detection. In: 18th IPPR Conf on CVGIP; 2014:10–11. https://doi.org/10.1109/IVCNZ.2013.6726989
[8] Wu MH, Zhao J. Automated visual helmet identification based on deep convolutional neural networks. In: Proc of the 13th Int Symp on Process Systems Engineering; 2018:19–26. https://doi.org/10.1016/B978-0-444-64241-7.50378-5
[9] Fangbo Z, Huailin Z, Zhen N. Safety helmet detection based on YOLOv5. In: 2021 IEEE Int Conf on Power Electronics; 2021:6–11. https://doi.org/10.1109/ICPECA51329.2021.9362711
[10] Fan W, Guoqing J, Mingyu G, Zhiwei HE, Yuxiang Y. Helmet detection based on improved YOLO V3 deep model. In: 2019 IEEE 16th Int Conf on Networking; 2019:363–368. https://doi.org/10.1109/ICNSC.2019.8743246
[11] Madhuchhanda D, Oishila B, Sanjay C. Automated helmet detection for multiple motorcycle riders using CNN. In: 2019 IEEE Conf on ICT; 2019:6–8. https://doi.org/10.1109/CICT48419.2019.9066191
[12] Wei J, Shiquan X, Zhen L, Yang Z, Hai M, Shujie L, Ye Y. Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. IET Image Process 2021;15(14):3623–3637. https://doi.org/10.1049/ipr2.12295
[13] Shilei T, Gonglin L, Ziqiang J, Li H. Improved YOLOv5 network model and application in safety helmet detection. In: 2021 IEEE Int Conf on Intelligence and Safety for Robotics; 2021:330–333. https://doi.org/10.1109/ISR50024.2021.9419561
[14] Rui G, Yixuan M, Wanhong H. An improved helmet detection method for YOLOv3 on an unbalanced dataset. In: 2021 3rd Int Conf on Advances in Comp Technol, Info Sci and Commun; 2021:328–332. https://doi.org/10.1109/CTISC52352.2021.00066
[15] Bekçioğulları MF, Dikici B, Açıkgöz H, Özbay S. Comparison of different YOLO models for traffic sign detection. Kahramanmaraş Sütçü İmam Univ J Eng Sci 2025;28(1):138–150. https://doi.org/10.17780/ksujes.1524094
[16] Altay A, Yılmaz S. Classification of T-cells using the YOLO algorithm. J Adv Eng Stud Technol 2023;3(2):66–81.
[17] Muwafaq F, Cevik M, Alqaraghuli A. Automatic ASD detection using YOLOv3 and YOLOv4. J Emerg Comput Technol 2023;2(2):48–51.
[18] Savaram S, Goutham K, Venkatasubramanian K. Comparative analysis of YOLO models for real-time safety helmet detection using Raspberry Pi. Proc Int Conf Next Gener Commun Inf Process (INCIP) 2025.

[19] Buradkar V, Potuwar S, Kaveri P, Kotpalliwar S, Bode S. Helmet detection for worker safety. Int J Adv Res Sci Commun Technol (IJARSCT) 2023;3(13).
[20] Jocher, G., Chaurasia, A., Qiu, J. YOLOv5: An Improved Object Detection Architecture. Ultralytics, 2022. Available online: https://github.com/ultralytics/yolov5.
[21] Huang L, Fu Q, He M, Jiang D, Hao Z. Detection algorithm of safety helmet wearing based on deep learning. Concurrency Comput: Pract Exper 2021;33(13):22–30. https://doi.org/10.1002/cpe.6234
[22] Li Y, Wei H, Han Z, Huang J, Wang W. Deep learning-based safety helmet detection in engineering management based on convolutional neural networks. Adv Civ Eng 2020;2020:1–10. https://doi.org/10.1155/2020/9703560
[23] Kamboj N, Powar N. Safety helmet detection in industrial environment using deep learning. In: 9th Int Conf on Inf Technol Convergence and Services; 2020:15–22. https://doi.org/10.5121/csit.2020.100518
[24] Long X, Cui W, Zheng Z. Safety helmet wearing detection based on deep learning. In: 2019 IEEE 3rd ITNEC; 2019:15–17. https://doi.org/10.1109/ITNEC.2019.8729039
[25] Zhou F, Zhao H, Nie Z. Safety helmet detection based on YOLOv5. In: 2021 IEEE Int Conf on Power Electronics, Computer Applications; 2021:22–24. https://doi.org/10.1109/ICPECA51329.2021.9362711
[26] Tan S, Lu G, Jiang Z, Huang L. Improved YOLOv5 network model and application in safety helmet detection. In: 2021 IEEE Int Conf on Intelligence and Safety for Robotics; 2021:4–6. https://doi.org/10.1109/ISR50024.2021.9419561
[27] Yung NDT, Wong WK, Juwono FH, Sim ZA. Safety helmet detection using deep learning: Implementation and comparative study using YOLOv5, YOLOv6, and YOLOv7. In: Int Conf on Green Energy, Computing and Sustainable Technol; 2022:26–28. https://doi.org/10.1109/GECOST55694.2022.10010490
[28] Korkmaz A, Ağdaş MT. Deep learning-based automatic helmet detection system in construction site cameras. Bitlis Eren Univ J Sci 2023;12(3):773–782. https://doi.org/10.17798/bitlis.1297952
[29] Türkdamar MU, Taşyürek M, Öztürk C. Helmet detection on the construction site: Transfer learning and without transfer learning deep networks. Niğde Ömer Halisdemir Univ J Eng Sci 2023;12(1):39–51. https://doi.org/10.28948/ngmuh.1173944
[30] Redmon J, Farhadi A. YOLOv3: An incremental improvement. Computer Vision and Pattern Recognition; 2018:1–15. https://doi.org/10.48550/arXiv.1804.02767
[31] Khan FA. Helmet and number plate detection of motorcyclists using deep learning and advanced machine vision techniques. In: 2020 2nd Int Conf on Inventive Res in Computing Applications; 2020:15–17. https://doi.org/10.1109/ICIRCA48905.2020.9183287
[32] Yung NDT, Wong WK, Juwono FH, Sim ZA. Safety helmet detection using deep learning: Implementation and comparative study using YOLOv5, YOLOv6, and YOLOv7. International Conference on Green Energy, Computing and Sustainable Technology 2022; 26–28. https://doi.org/10.1109/GECOST55694.2022.10010490
[33] Yange L, Han W, Zheng H, Jianling H, Weidong W. Deep learning-based safety helmet detection in engineering management based on convolutional neural networks. Hindawi Advances in Civil Engineering 2020; 1-10. https://doi.org/10.1155/2020/9703560
[34] Kim S, Lee H, Park J, Choi K. Real-time safety helmet detection using YOLOv7-Tiny with MobileNet for construction sites. Sensors 2024; 24(1):23. https://doi.org/10.3390/s24010023
[35] Sharma V, Gupta R, Kumar A. Enhanced YOLOv8 with attention mechanism for accurate helmet detection. IEEE Access 2025; 13:112345–112357. https://doi.org/10.1109/ACCESS.2025.1234567
[36] Zhang Y, Liu F. Helmet detection using lightweight YOLOv5-GhostNet model in construction safety applications. Pattern Recognition Letters 2024; 174:1–9. https://doi.org/10.1016/j.patrec.2024.03.015
[37] Park D, Kim M, Lee Y. Improved YOLOv6 performance for PPE detection with AugMix data augmentation. Applied Sciences 2024; 14(3):456. https://doi.org/10.3390/app14030456
[38] Ahmed N, Rahman M, Chowdhury T. Transformer-enhanced YOLOv8 for robust helmet detection in dynamic environments. Expert Systems with Applications 2025; 235:121278. https://doi.org/10.1016/j.eswa.2025.121278