Journal of Autonomous Intelligence (2024) Volume 7 Issue 2
doi: 10.32629/jai.v7i2.1255

ORIGINAL RESEARCH ARTICLE

Hardhat-wearing detection based on YOLOv5 in Internet-of-Things

Wanbo Luo[1], Rajeswari Raju[2], Khairul Khaizi Mohd Shariff[3], Ahmad Ihsan Mohd Yassin[3,*]

[1] *School of Electrical Engineering, Universiti Teknologi MARA, Shah Alam 40450, Malaysia*

[2] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Kuala Terengganu 21080, Malaysia*

[3] *Microwave Research Institute, Universiti Teknologi MARA, Shah Alam 40450, Malaysia*

*** Corresponding author:** Ahmad Ihsan Mohd Yassin, ihsan_yassin@uitm.edu.my

ABSTRACT

Worker safety is paramount in many industries. An essential component of industrial safety protocols involves the proper use of hardhats. However, due to lax safety awareness, many workers neglect to wear hardhats correctly, leading to frequent on-site accidents in China. Traditional detection methods, such as manual inspection and video surveillance, are inefficient and costly. Real-time monitoring of hardhat use is vital to boost compliance with hardhat usage and decrease accident rates. Recently, the advancement of the Internet of Things (IoT) and edge computing has provided an opportunity to improve these methods. In this study, two detection models based on You Only Look Once (YOLO) v5, hardhat-YOLOv5s and hardhat-YOLOv5n, were designed, validated, and implemented, tailored for hardhat detection. First, a public hardhat dataset was enriched to bolster the detection model's robustness. Then, hardhat detection models were trained using the YOLOv5s and YOLOv5n, each catering to edge computing terminals with varying performance capacities. Finally, the models were validated using image and video data. The experimental results indicated that both models provided high detection precision and satisfied practical application needs. On the augmented public dataset, the hardhat-YOLOv5s and hardhat-YOLOv5n models have a Mean Average Precision (mAP) of 87.9% and 85.5%, respectively, for all six classes. Compared with the hardhat-YOLOv5s model, Parameters and Giga Floating-point Operations (GFLOPs) of the hardhat-YOLOv5n model decrease by 74.8% and 73.4%, respectively, and Frame per Second (FPS) increases by 30.5% on the validation dataset, which is more suitable for low-cost edge computing terminals with less computational power.

Keywords: occupational safety & health; object detection; deep learning; data augmentation; edge computing terminal

ARTICLE INFO

Received: 18 September 2023
Accepted: 20 October 2023
Available online: 22 December 2023

1. Introduction

Hardhats are crucial safety equipment for workers in industrial settings, serving to shield the head from falling objects and high pressure, thus protecting lives in hazardous situations[1]. Regrettably, due to inadequate safety consciousness, some workers forgo the use of hardhats during daily inspections, equipment maintenance, and infrastructure construction, which often leads to injuries or even fatal accidents[2].

Industries involving aerial work, including construction, power, mining, and petroleum, underscore the importance of correct protective equipment usage. As a primary defense against external impact, hardhats are essential to worker safety[3]. From January 2016 to August 2018, more than 160,000 workplace safety accidents occurred across various industries in China, resulting in over 100,000 deaths. In the power industry alone, 170 accidents resulted in 232 deaths and direct economic losses of 22.9 million Chinese yuan[4].

In the construction industry, specifically, 2019 saw at least 773 safety accidents and 904 fatalities across China[5], a 5.31% and 7.62% increase in total accidents and fatalities, respectively, from the previous year. Notably, accidents caused by falling or colliding objects accounted for approximately 15.91% of total fatalities[6]. Among the 78 nationwide construction accidents, 53 fatalities involved workers not wearing hardhats, accounting for 67.95% of total casualties[7].

Conclusively, most work site fatalities stem from the lack of hardhat usage. Prompt detection and warnings for non-compliance can significantly reduce incidents and losses attributable to such negligence[8]. Some industries have implemented video surveillance systems to monitor workers' compliance, including hardhat usage[9]. Yet, manual surveillance, which often involves a single individual monitoring multiple video channels, is labor-intensive and prone to negligence and missed detections[10], highlighting the need to transition to an unattended intelligent video analysis system.

Recently, deep learning algorithms have shown promise in tasks such as image classification and object detection. Concurrently, edge computing, an expansion scheme of cloud computing, has been integrated with IoT to address these issues[11]. As a novel computing model, edge computing performs data processing, storage, and other tasks near users, providing efficient services[12,13]. The YOLO architecture is not only fast but also has high detection accuracy. Therefore, YOLO is introduced into IoT for hardhat detection at work sites via edge computing terminals, which can significantly mitigate risks associated with incorrect hardhat usage.

2. Literature review

Due to the rapid development of deep learning, many studies use deep learning methods to detect hardhat-wearing at work sites. Although these methods based on Faster Region-Convolutional Neural Network (RCNN), YOLOv3, and YOLOv4 had improved the algorithm to perform hardhat-wearing detection, their parameters and calculations are enormous, which were not conducive to deploying into edge computing terminals.

Recently, edge computing enabled more image processing tasks to be implemented at edge sides. Therefore, intelligent video surveillance at the edge has become a trend. Inspired by YOLO, Nguyen et al.[14] proposed a novel form of real-time human detection in 2021, focused on a good trade-off between accuracy and processing time. The trained model detected humans with accuracies of 95.05% and 96.81% on Raspberry PI 3B with 2 FPS. In 2022, Feng et al.[15] investigated the inference workflow and performance of the YOLO network in three different edge computing terminals, which were NVIDIA Jetson Nano, NVIDIA Jetson Xavier NX and Raspberry Pi 4B. The analysis results indicated that Jetson Nano was a trade-off edge computing terminal in terms of performance and cost. The trained model achieved up to 15 FPS of detected videos when running YOLOv4-tiny.

In 2020, Wu et al.[16] proposed a hardhat-wearing detection and identification method based on an improved Faster RCNN algorithm. The feature layers obtained in multiple stages were fused, and multi-scale detection was performed. Meanwhile, the size of the region proposal was modified to make the model optimal. The average detection accuracy rate of the five types of objects for workers wearing red, yellow, white, and blue color helmets and not wearing helmets reached 85.8%.

Wang et al.[17] proposed an improved YOLOv3 object detection algorithm in 2020. Combined with the YOLOv3 algorithm's objective function, this algorithm improved the Generalized Intersection over Union (GIoU) calculation method to design a new objective function to achieve Intersection over Union (IoU) local optimization as the local optimization of the objective function. The mAP of the improved YOLOv3 algorithm increased by 2.07% and 2.05% in the Visual Object Classes (VOC) 2007 public dataset and helmet-wearing dataset, respectively, compared with the YOLOv3 algorithm.

In the same year, Ge et al.[18] proposed a helmet-wearing detection method that integrates environmental features based on YOLOv4. To supplement the features lost in the convolution pooling process, under the condition that the output feature maps of 3 different sizes obtained by YOLOv4 were consistent with the receptive fields of the feature maps obtained by feature extraction from the original image, the two were added to fuse the high and low layer features to capture more detailed information. A 3 × 3 convolution operation was used on the fused feature map to reduce the aliasing effect of the fused feature map to ensure feature stability. The mAP reached 91.55%. Compared with YOLOv4, the mAP had increased by 5.2%.

To solve the problem of enormous model layers and parameters, two models for hardhat-wearing detection were trained respectively by using YOLOv5s and YOLOv5n to adapt to edge computing terminals with different computing power. In addition, some methods used open-sourced public datasets, and some used self-collected datasets. These datasets were usually annotated into only two classes, “person-with-helmet” and “person-no-helmet”. Therefore, four new classes were added to make the dataset label reasonable and scalable. The new dataset label consists of six classes, “helmet”, “head-with-helmet”, “person-with-helmet”, “head”, “person-no-helmet”, and “face”. Furthermore, these methods had low detection accuracy for occluded and crowded objects. Therefore, images of complex work scenes were added to the dataset to improve detection accuracy in this paper.

3. The YOLOv5 algorithm

3.1. Introduction to YOLO

The YOLO series is a popular one-stage object detection methodology, encompassing versions from YOLO to YOLOv5. Initially proposed by Redmon et al. at the conference on Computer Vision and Pattern Recognition (CVPR) in 2016, YOLO represented a pivotal shift in object detection algorithms[19]. YOLO innovatively merged the processes of region proposal generation and detection, effectively treating object detection as a regression problem and achieving one-stage detection. YOLO segmented the image into N × N sub-regions, predicting the probability, classification, and position offset of objects within each sub-region. Due to its simplistic structure, YOLO facilitated rapid operational speed.

In June 2020, YOLOv5 was open-sourced, featuring a fast detection speed and a substantially lightweight model size. YOLOv5 introduced five distinct object detection networks, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, tailored to cater to a range of applications. A comparison between various versions of YOLO is presented in **Table 1**, with the YOLOv5s model demonstrating the highest detection precision and fastest detection speed. The data represents various iterations of the YOLO object detection algorithm. Each version improves upon the speed or precision of its predecessor. The FPS is generally used to measure the detection speed of the model, and the larger the value, the faster the detection speed. The mAP is generally used to measure the detection accuracy of the model, and the larger the value, the higher the accuracy.

Table 1. Performance comparison of YOLO series object detection models.

Model	Network	FPS	VOC2007 (mAP/%)	VOC2012 (mAP/%)	COCO (mAP/%)
YOLO	VGG16	45	66.4	57.9	-
YOLO9000	Darknet19	40	78.6	73.5	21.6
YOLOv3	Darknet53	78	74.5	-	57.9
YOLOv4	CSPDarknet53	66	-	-	43.5
YOLOv5s	Conv2d (6 × 6) + CSPDarknet53	113	-	-	51.8
YOLO	VGG16	45	66.4	57.9	-

The original YOLO model, using a Visual Geometry Group 16 (VGG16) network, operates at 45 FPS, with 66.4% and 57.9% mAP on VOC2007 and VOC2012 datasets, respectively. YOLO9000, with Darknet19, yields 78.6% and 73.5% mAP on these datasets and introduces Common Objects in Context (COCO) evaluation at 21.6% mAP, albeit at a slightly slower 40 FPS. YOLOv3 enhances speed to 78 FPS with 74.5% mAP on VOC2007 and 57.9% mAP on COCO, while YOLOv4 focuses on COCO performance, reaching 43.5% mAP at 66 FPS. The latest, YOLOv5s, running on a Focus + Cross Stage Partial Darknet53 (CSPDarknet53) network, maximizes speed at 113 FPS with 51.8% mAP on COCO, illustrating the ongoing development of YOLO models balancing speed and precision.

3.2. YOLOv5 architecture

The architecture of the YOLOv5 model (**Figure 1**) is segmented into three core parts: the backbone, the neck, and the head.

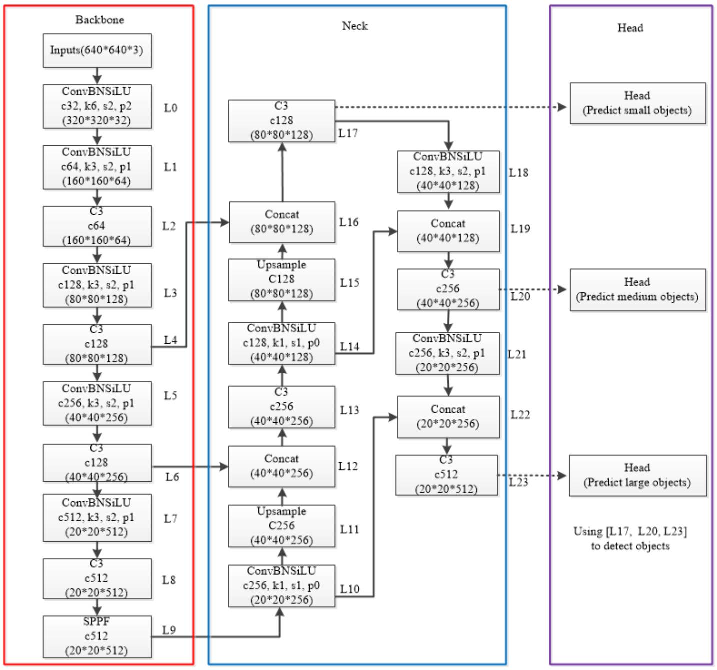


Figure 1. YOLOv5 architecture.

The backbone, based on the modified CSPDarknet53 structure, is a critical element designed to extract features from the input data. This feature extraction is fundamental to the model’s ability to identify and isolate relevant aspects of the data that are informative for the task at hand. Next, the neck, which incorporates the Spatial Pyramid Pooling Fast (SPPF) and the updated Cross Stage Partial-Path Aggregation Network (CSP-PAN) modules, performs feature fusion. Feature fusion is a crucial aspect of the object detection task, as it allows the model to combine features at various scales and levels of abstraction. This capability enhances the model’s robustness to variations in object size, shape, and orientation, leading to improved detection performance. The final part of the architecture, the head, is responsible for producing prediction results. This component of the model interprets the high-level features extracted and fused by the previous stages, turning them into a final prediction about the object's class.

The ConvBNSiLU layer consists of a convolutional layer, a batch normalization layer, and a Sigmoid Linear Unit (SiLU) activation function sequentially. The *c*, *k*, *s*, and *p* represent channel, kernel, stride, and padding. The upsampling layer represents an upsampling operation. The concatenation layer represents the operation of concatenating two layers. The C3 layer consists of different ConvBNSiLU, two BottleNeck structures, and Concat layers. Some C3 layers contain BottleNeck-1 structures, while others contain BottleNeck-2 structures. Larger YOLOv5 models, such as YOLOv5x, YOLOv5l, and YOLOv5m, have more repeated BottleNeck-1 or BottleNeck-2 structures in the C3 structure. In contrast, smaller YOLOv5 models, such as YOLOv5s and YOLOv5n, have fewer repeated BottleNeck-1 or BottleNeck-2 structures. Both BottleNeck structures consist of a ConvBNSiLU layer (k = 1) and a ConvBNSiLU layer (k = 3). The BottleNeck-1 adds a residual structure to the initial input, while the BottleNeck-2 does not.

YOLOv5 has some advantages compared to the previous version. First, the Conv2d (6 × 6) structure replaces the Focus structure, and the SPPF structure replaces the SPP structure, boosting efficiency. In addition, it uses various data augmentation techniques to improve the model's ability to generalize and reduce overfitting. Furthermore, it applies several sophisticated training strategies to enhance the model's performance. Overall, the design of the YOLOv5 model encapsulates a variety of architectural choices aimed at maximizing detection performance while maintaining flexibility in terms of input data and computational efficiency.

4. Methodology

4.1. Detection framework

Figure 2 presents a hardhat-wearing detection framework based on YOLOv5 and IoT. Initially, the management center uses a cloud server to train a model on a public hardhat detection dataset using YOLOv5. The trained model is then dispatched to edge computing terminals via IoT, where it is automatically deployed. These terminals perform hardhat-wearing detection. The management center can enhance the robustness of the detection model by retraining it using images from diverse complex scenarios, captured by the terminals. Subsequently, the refined model, once it meets the precision standards, is re-transmitted to the edge computing terminals for detection. By exploiting the computational power of the cloud computing center to train substantial data, the system can improve model training speed. Additionally, by having the edge computing terminals only send images of incorrectly worn hardhats to the management center, the system substantially reduces IoT transmission consumption.

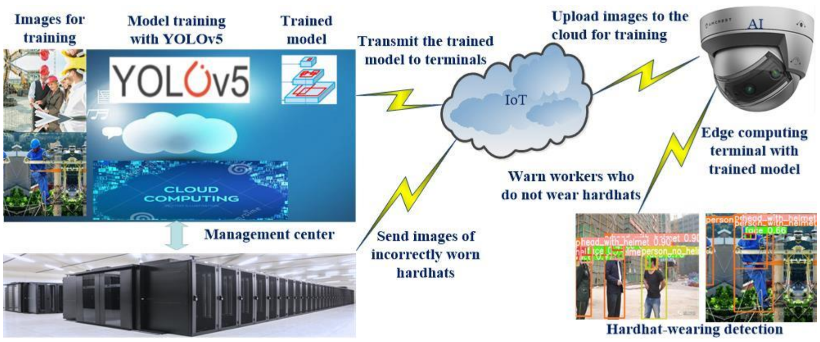
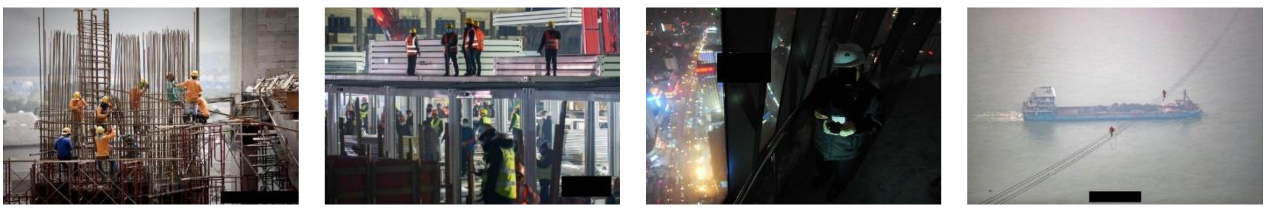


Figure 2. Hardhat-wearing detection framework based on YOLOv5 in IoT.

4.2. Data augmentation

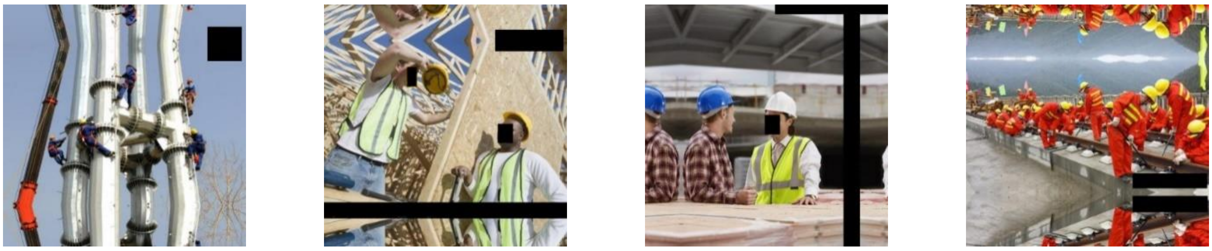
To improve the robustness of the trained model, images of complex work scenes were added to the dataset, such as images containing occluded, long-distance, crowded, and low-light objects. **Figure 3** shows sample images of complex work scenes included in the dataset.



(a) **(b)** **(c)** **(d)**

Figure 3. Sample images of complex work scenes: **(a)** sample image includes occluded objects; **(b)** sample image includes crowded objects; **(c)** sample image includes low-light objects; **(d)** sample image includes long-distance objects.

The original dataset's size was insufficient, prompting the use of data augmentation techniques to increase its volume. These methods encompassed image distortion, spatial translation, rotation, and flipping. **Figure 4** displays sample images after the application of these data augmentation techniques.



(a) **(b)** **(c)** **(d)**

Figure 4. Sample images implemented by data augmentation: **(a)** sample image after the application of image distortion; **(b)** sample image after the application of image rotation; **(c)** sample image after the application of image spatial translation; **(d)** sample image after the application of image flip.

4.3. Data annotation

At a typical worksite, hardhats usually come in blue, red, yellow, and white. Thus, these objects are primarily labeled based on their color. When annotating a "head-with-helmet" object, the bounding box should encompass the helmet, head, and neck. A "face" object should be included within the bounding box of the "head-with-helmet" object. The bounding box of a "person-with-helmet" object should enclose the "helmet", "head-with-helmet", and "face" objects. Objects labeled as "head" and "person-no-helmet" denote individuals not wearing hardhats.

The original public dataset contained 5000 images with bounding box annotations in the PASCAL VOC format for these three classes (person, head, and helmet), which had 25,501 labels in total[20]. After data augmentation, the new dataset had 6583 images. It was relabeled into six classes in YOLO format, which had 81,149 labels in total. The number of "helmet", "head-with-helmet", "person-with-helmet", "head", "person-no-helmet", and "face" labels were 20,803, 17,156, 16,687, 6015, 6405, and 14,083, respectively. The number of labels for each class increased significantly. In particular, many interfering images were added to improve accuracy, containing people wearing baseball caps or bamboo hats. A comparison of the number of labels in different datasets is shown in **Table 2**.

Table 2. Comparison of the number of labels in different datasets.

Label	Number (5000 images)	Number (5000 images of relabeled)	Number (6583 images of relabeled)	Comments
helmet	18,752	17,246	20,803	hardhat
head-with-helmet	0	14,379	17,156	workers who wear hardhats on their head
person-with-helmet	0	13,244	16,687	workers who wear hardhats
head	5525	5525	6015	workers who do not wear hardhats on their head
person-no-helmet	1224	4775	6405	workers who do not wear hardhats
face	0	12,671	14,083	face of workers

4.4. Model training

The experiment was conducted on the PyTorch framework. The hardware contained an Nvidia GeForce RTX 3060 Graphics Processing Unit (GPU) with 12 GB of graphic memory, a 13th generation Intel Core i7-13700KF Central Processing Unit (CPU) with 3.4 GHz frequency, and a Random Access Memory (RAM) with 32GB. The operating system was Windows 11 Professional Edition (22H2). The primary software components were Python 3.8.17, PyTorch 2.0.1, YOLOv5 7.0, and Compute Unified Device Architecture (CUDA) 11.8.

The dataset was divided into a training and a validation dataset. The training dataset contained 6083 images, and the validation dataset contained 500 images. All images were resized to 640 × 640 pixels, and the batch size was set to 16 during training. The model was trained for 100 epochs using pre-trained weights from the YOLOv5s.pt and YOLOv5n.pt files.

5. Results & discussion

5.1. Training results

The validation dataset had 7730 instances in total, including 2006 "helmet" instances, 1668 "head-with-helmet" instances, 1524 "person-with-helmet" instances, 595 "head" instances, 473 "person-no-helmet" instances, and 1464 "face" instances.

The training results of the hardhat-YOLOv5s model on the validation dataset are shown in **Table 3**. The hardhat-YOLOv5s model achieved good performance. The Precision, Recall, mAP50, and mAP50-95 of all six classes were 0.886, 0.806, 0.879, and 0.544, respectively. In addition, the mAP50 reached 0.879, indicating that good capabilities for object localization and classification can be obtained during the prediction process. From the prediction results, the mAP50 of all classes, except the "face" class, exceeded 88%. The Recall of the "face" class was 0.673, indicating that some face objects failed to predict.

Table 3. Training results with YOLOv5s.

Class	Instances	Precision	Recall	mAP50	mAP50-95
all	7730	0.886	0.806	0.879	0.544
helmet	2006	0.899	0.793	0.883	0.53
head-with-helmet	1668	0.939	0.817	0.921	0.592
person-with-helmet	1524	0.895	0.882	0.927	0.659
head	595	0.916	0.822	0.882	0.538
person-no-helmet	473	0.841	0.85	0.9	0.629
face	1464	0.824	0.673	0.76	0.32

The training results of the hardhat-YOLOv5n model on the validation dataset are shown in **Table 4**. The Precision, Recall, mAP50, and mAP50-95 of all six classes were 0.873, 0.774, 0.855, and 0.511, respectively. Therefore, the hardhat-YOLOv5n model also achieved good performance.

Table 4. Training results with YOLOv5n.

Class	Instances	Precision	Recall	mAP50	mAP50-95
all	7730	0.873	0.774	0.855	0.511
helmet	2006	0.883	0.776	0.863	0.504
head-with-helmet	1668	0.936	0.79	0.908	0.572
person-with-helmet	1524	0.894	0.843	0.912	0.614
head	595	0.889	0.793	0.856	0.499
person-no-helmet	473	0.85	0.825	0.885	0.596
face	1464	0.788	0.618	0.706	0.283

Figure 5 shows comparisons of Precision, Recall, mAP50, and mAP50-95 of the two models for all six classes. These metrics of the hardhat-YOLOv5s model were better than those of the hardhat-YOLOv5n model.

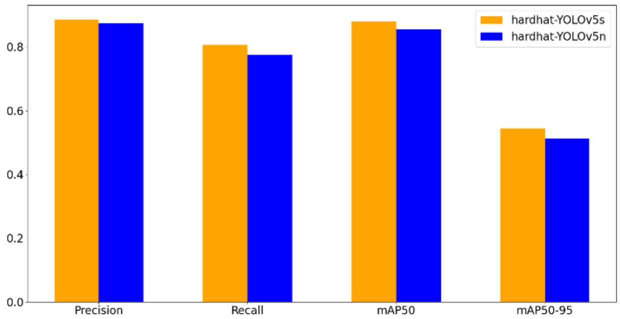


Figure 5. Comparison of training metrics of the two models.

Figure 6 shows the Precision-Recall curves of the two models. For the hardhat-YOLOv5s model, the mAP50 of “helmet” class, “head-with-helmet” class, “person-with-helmet” class, “head” class, “person-no-helmet” class, and “face” class were 0.883, 0.921, 0.927, 0.882, 0.900, and 0.760, respectively. Compared to the hardhat-YOLOv5s model, the mAP50 of these classes were 0.863, 0.908, 0.912, 0.856, 0.885, and 0.706, respectively, on the hardhat-YOLOv5n model. The average mAP50 of the hardhat-YOLOv5s and hardhat-YOLOv5n models for all six classes was 0.879 and 0.855, respectively.

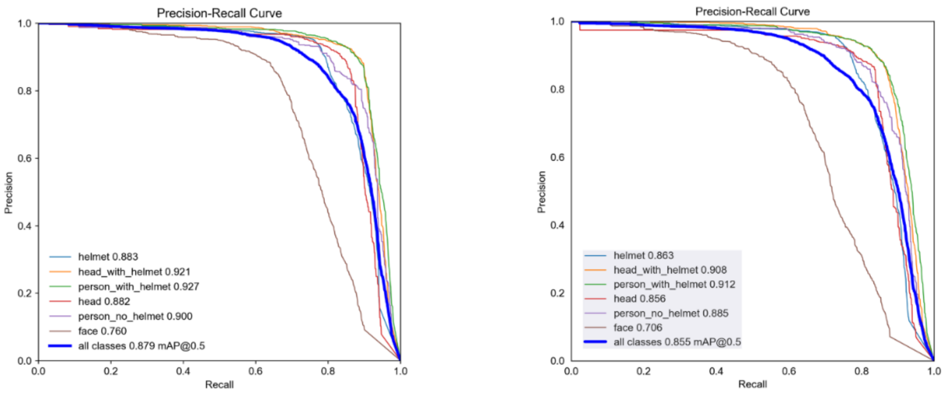


Figure 6. Precision-Recall curve of the two models: **(a)** Precision-Recall curve of hardhat-YOLOv5s; **(b)** Precision-Recall curve of hardhat-YOLOv5n.

Figure 7 shows the mAP50 metric comparison of each class on the two models. The mAP50 metric of each class on the hardhat-YOLOv5s model was better than the hardhat-YOLOv5n model.

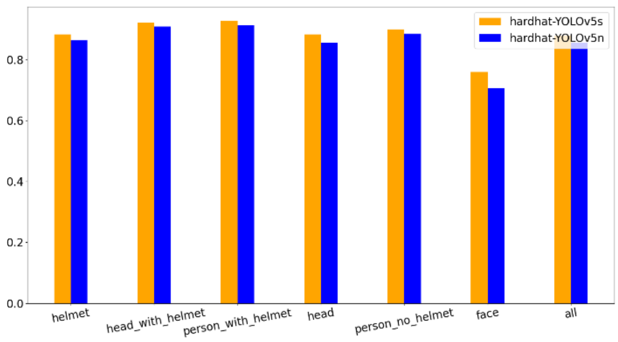


Figure 7. Comparison of the mAP50 metric of each class on two models.

5.2. Detection results

As a result of training with YOLOv5s and YOLOv5n, two distinct hardhat detection models, hardhat-YOLOv5s and hardhat-YOLOv5n, were obtained. Key performance metrics for these models are outlined in **Table 5**, showing the hardhat-YOLOv5s model achieving a higher mAP50 of 87.9% compared to the hardhat-YOLOv5n model’s mAP50 of 85.5%. The metric of Parameters indicates the space complexity of an algorithm, which means how much graphic memory the model occupies in the terminal device. The metric of GFLOPs indicates the time complexity of an algorithm, which is generally used as an indirect measure of the speed of a neural network model.

Table 5. Performance metrics of the two models.

Model	Layers	Parameters	GFLOPs	mAP@0.5
hardhat-YOLOv5s	157	7,026,307	15.8	0.879 (all classes)
hardhat-YOLOv5n	157	1,767,283	4.2	0.855 (all classes)

Both models had 157 layers. The hardhat-YOLOv5n model had 1,767,283 parameters. Compared to the hardhat-YOLOv5s model with 7,026,307 parameters, its parameters decreased by 74.8%. Therefore, the hardhat-YOLOv5n model occupied less graphic memory. The metrics of GFLOPs of the hardhat-YOLOv5s and hardhat-YOLOv5n models were 15.8 and 4.2, respectively. Compared to the hardhat-YOLOv5s model, the GFLOPs of the hardhat-YOLOv5n model decreased by 73.4%. So, the hardhat-YOLOv5n model theoretically had a faster detection speed than the hardhat-YOLOv5s model.

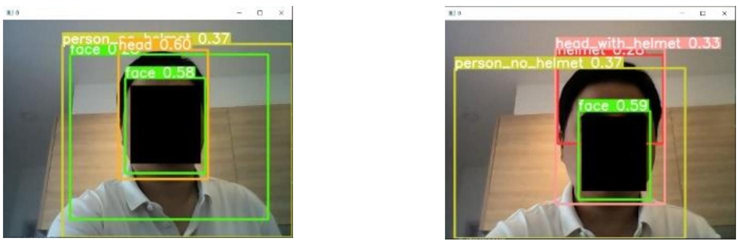
5.2.1. Model effectiveness evaluation

The effectiveness of the proposed method was evaluated using images from work sites. By inputting these images into the model, detection results were produced, and each object was enclosed by a bounding box accompanied by a confidence value. **Figure 8** depicts the detection results, wherein the bounding box's color is orange if the individual is wearing a hardhat, and yellow otherwise. Both models correctly predicted all objects in the same image.

(a) **(b)** **(c)**

Figure 8. Detection results using sample image: **(a)** sample image; **(b)** detection results of hardhat-YOLOv5s; **(c)** detection results of hardhat-YOLOv5n.

The method was also extended to detecting videos and live camera feeds. Real-time camera detection results are shown in **Figure 9**, where the hardhat-YOLOv5n model inaccurately predicted a hardhat with a confidence value of 0.28. On the contrary, the hardhat-YOLOv5s model correctly predicted the head with a confidence value of 0.60.



(a) **(b)**

Figure 9. Detection results using camera: **(a)** detection results of hardhat-YOLOv5s; **(b)** detection results of hardhat-YOLOv5n.

Figure 10 presents detection results using sample video, with the hardhat-YOLOv5s model demonstrating better accuracy in detecting long-distance objects, while the hardhat_YOLOv5n model missed these.

(a) **(b)**

Figure 10. Detection results using sample video: **(a)** detection results of hardhat-YOLOv5s; **(b)** detection results of hardhat-YOLOv5n.

5.2.2. Model speed evaluation

Generally, the time it takes a model to perform prediction on an image or a video is expressed in latency, the forward propagation time of the model. It contains the time spent on pre-processing, inference, and Non-Maximum Suppression (NMS) processes. The NMS is used to remove redundant prediction boxes. The shape of the image for detection is $3 \times 640 \times 640$. The Latency calculation formula is given in Equation (1). FPS is the inverse of Latency. The FPS calculation formula is given in Equation (2).

$$\text{Latency} = \text{pre-process} + \text{inference} + \text{NMS} \tag{1}$$

$$\text{FPS} = 1/\text{Latency} \tag{2}$$

The images of the validation dataset were respectively validated using the two models. **Table 6** shows the average detection speed per image for the two models on the validation dataset images. The hardhat-YOLOv5n model had a higher detection speed with an average FPS of 117.6 than the hardhat-YOLOv5n model with an average FPS of 90.1, increasing by 30.5%.

Table 6. Speed metrics on validation dataset images.

Model	Pre-process (ms)	Inference (ms)	NMS (ms)	Latency (ms)	FPS
hardhat-YOLOv5s	0.3	7.3	3.5	11.1	90.1
hardhat-YOLOv5n	0.5	4.5	3.5	8.5	117.6

6. Conclusion

This paper proposes two detection models based on YOLOv5 customized for hardhat wearing. The computing power of the cloud computing center is used to train the hardhat dataset with YOLOv5s and YOLOv5n to obtain two detection models, hardhat-YOLOv5s and hardhat-YOLOv5n. The two models can be transmitted to edge computing terminals through IoT to detect no-hardhat-wearing. Edge computing terminals perform detection tasks at work sides. Compared with traditional video surveillance methods, the terminals do not need to transmit images back to the cloud computing center, which can reduce IoT bandwidth consumption. The experimental results show that both models based on YOLOv5 have high accuracy. Compared with the hardhat-YOLOv5s model, the hardhat-YOLOv5n model has a higher detection speed. Therefore, the hardhat-YOLOv5n model is more suitable for edge computing terminals with lower computing power and cost. Meanwhile, by training different object datasets, the detection method can be used in many industries such as construction, power, mines, petroleum, etc.

Author contributions

Conceptualization, LW and AIMY; methodology, RR; software, LW; validation, LW, AIMY and KKMS; formal analysis, KKMS; investigation, LW; writing—original draft preparation, LW; writing—review and editing, LW, AIMY, RR and KKMS; data curation, LW; supervision, AIMY, RR and KKMS. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Chen Z, Su M. Improved mask R-CNN method for intelligent monitoring of helmet in power plant. In: Proceedings of the Photonics & Electromagnetics Research Symposium; 22 November 2021; Hangzhou, China.
2. Yang B, Qin H, Xie H, et al. Helmet detection under the power construction scene based on image analysis. In: Proceedings of the IEEE 7th International Conference on Computer Science and Network Technology; 19–20 October 2019; Dalian, China.
3. Chang X, Liu X. Fault tree analysis of unreasonably wearing helmets for builders. Journal of Jilin Jianzhu University 2018; 35(6): 65–69.
4. Zheng H. Research on Video Recognition of Safety Protection Measures for Electric Power Construction Personnel [Master's thesis]. Guangdong University of Technology; 2021.
5. Liang W, Jing Z, Zhou Z, et al. Research on detection algorithm of helmet wearing state in electric construction. In: Proceedings of the 14th National Academic Conference on Signal and Intelligent Information Processing and Application; 11 April 2021; Beijing, China.
6. Sun C, Wang Y, Zhao J. Research on online detection technology of construction workers' hardhats. Science and Technology and Innovation 2021; 19: 85–86.
7. Xu S, Wang Y, Gu Y. Construction site safety helmet wearing detection based on improved region convolutional neural network. Computer Engineering and Design 2020; 4(5): 1385–1389.
8. Lv M. Analyzing the monitoring technology of helmet wearing on construction site. Low Carbon Word 2018; 7: 205–206.
9. Liao M, Dan W, Lai Q. The communication structure of integrated automation system in 220KV conventional substation. Electric Engineering 2018; 7: 76–78.
10. Wang Y, Su B, Zhao H. The concept and development trend of power Internet of Things (Chinese). Telecommunications Science 2010; 26(S3): 9–14.
11. Shi W, Cao J, Zhang Q et al. Edge computing: vision and challenges. IEEE Internet of Things Journal 2016; 3(5): 637–646.
12. Wang S, Zafer M, Leung KK. Online placement of multi-component applications in edge computing environments. IEEE Access 2017; 5: 2514–2533.
13. Shi W, Sun H, Cao J. Edge computing—An emerging computing model for the Internet of Everything era. Journal of Computer Research and Development 2017; 54(5): 907–924.
14. Nguyen HH, Ta TN, Nguyen NC, et al. YOLO Based Real-Time Human Detection for Smart Video Surveillance at the Edge. In: Proceedings of IEEE Eighth International Conference on Communications and Electronics (ICCE); 13 January 2021; Phu Quoc Island, Vietnam.
15. Feng H, Mu G, Zhong S, et al. Benchmark Analysis of YOLO Performance on Edge Intelligence Devices. Cryptography 2022; 6(2): 16.
16. Wu D, Wang H, Li J. Safety helmet detection and identification based on improved faster RCNN. Information Technology and Informatization 2020; 1: 17–20.
17. Wang B, Li WJ, Tang H. Improved YOLOv3 algorithm and its application in helmet detection. Computer Engineering and Applications 2020; 56(9): 33–40.
18. Ge Q, Zhang Z, Yuan L, et al. Safety helmet wearing detection method of fusing environmental features and improved YOLOv4. Journal of Image and Graphics 2021; 26(12): 2904–2917.
19. Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 27–30 June 2016; Las Vegas, USA.
20. Safety Helmet Detection. Available online: https://www.kaggle.com/andrewmvd/hard-hat-detection (accessed on 24 September 2023).