Hardhat-YOLO: A YOLOv5-based Lightweight Hardhat-Wearing Detection Algorithm in Substation Sites

Wanbo Luo[1], Ahmad Ihsan Mohd Yassin[2]*, Khairul Khaizi Mohd Shariff[3], Rajeswari Raju[4]
School of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia[1]
Department of Artificial Intelligence, Leshan Vocational and Technical College, Leshan, China[1, 2]
Microwave Research Institute, Universiti Teknologi MARA, Shah Alam, Malaysia[2, 3]
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Kuala Terengganu, Malaysia[4]

***Abstract*—Accidents at substation sites have occurred frequently in recent years due to workers violating power safety regulations by not wearing hardhats. Therefore, it is necessary to provide real-time warnings when detecting workers without hardhats. Nevertheless, the deployment of deep learning-based algorithms necessitates the utilization of a multitude of parameters and computations, which consequently engenders an augmented expenditure on hardware. Therefore, using a lightweight backbone can address this issue well. This paper explored methods, such as deep learning, power Internet of Things (PIoT), and edge computing and proposed a lightweight and effective method called hardhat-YOLO for hardhat-wearing detection. First, the MobileNetv3-small backbone replaced the backbone of You Only Look Once (YOLO) v5s to reduce parameters and increase detection speed. In addition, the Convolutional Block Attention Module (CBAM) was effectively integrated into the network to improve detection precision. Finally, the hardhat-YOLO model, trained with a customized dataset, was transmitted to edge computing terminals in substations through PIoT for hardhat-wearing detection. Compared to the YOLOv5s model, the Parameters and Giga Floating Point Operations (GFLOPs) of the proposed model decreased by about 35.5% and 54.4%, respectively, and Frame per Second (FPS) increased by 17.3% approximately. The experimental results indicated that the hardhat-YOLO model achieved a Mean Average Precision of 83.3% at 50% intersection over union (mAP50), correctly and effectively conducting hardhat-wearing detection tasks.**

***Keywords*—Hardhat-wearing detection; You Only Look Once (YOLO); MobileNet; Substation; power Internet of Things (PIoT)**

I. Introduction

Electricity is a crucial industry for ensuring national development and daily life. The power industry has experienced significant growth, resulting in widespread coverage of power systems [1]. The State Grid Corporation of China (State Grid) has a vast transmission range with numerous power work sites. Additionally, various types of electrical equipment operate in substation sites, making working in these areas a high-risk activity [2]. As worker safety is of utmost importance, it is imperative that all workers strictly adhere to the safety regulations, including wearing a hardhat [3]. The hardhat serves as a critical safety measure at the power work site, dispersing the impact force of falling objects through its shell and further buffering and absorbing the impact force through its interior to ensure the safety of workers' heads [4].

The statistical analysis of accidents in the power industry between 2015 and 2020 in China revealed that 131 accidents occurred during power production, accounting for 60%, while 69 accidents occurred during power construction, accounting for 29%, and 26 accidents occurred during power technology improvement, accounting for 11%. The analysis of the causes of casualties in power operations between 2014 and 2018 showed that illegal operations accounted for 73.68%, equipment accounted for 7.37%, natural causes accounted for 7.37%, and other causes accounted for 11.58% [5]. The survey data on personal injury accidents in the past decade reveals that the leading causes of fatalities in safety accidents are falling utility poles, object impacts, falls from heights, and electric shocks. The most harmful incident, according to the statistics, is when a worker is struck on the head by a falling object, resulting in head and neck injuries or death. 80% of the casualties were caused by non-compliance with safety regulations and failure to take safety precautions [6].

A video monitoring system has been installed in substation sites with fundamental functions, including real-time display and playback of historical video data. However, it lacks an alarm function for abnormal conditions. Therefore, an unattended system that can rapidly detect hardhat wearing should be urgently implemented to provide early warnings and reduce the occurrence of accidents. With the construction and development of the smart grid, power system equipment has become more intelligent [7]. The Power Internet of Things (PIoT) is formed by integrating the smart grid and the Internet of Things (IoT), making it a strategic transformation technology for the State Grid [8]. PIoT's superior advantages have attracted the attention of many researchers, leading to its introduction into all production chains [9-11].

In PIoT, many services require lower latency, which the cloud-centric computing model struggles to meet. Edge computing has been proposed as an expansion scheme of cloud computing to address this issue [12]. Edge computing terminals collect, process, and store data on edge sides [13]. The terminals have data processing capabilities without uploading all data to the cloud center, which saves the transmission

*Corresponding author.

bandwidth of PIoT significantly. Therefore, edge computing terminals are more suitable for deploying in substations for hardhat-wearing detection.

Compared to resource-rich cloud computing platforms, edge platforms still face challenges such as inferior hardware performance, power consumption sensitivity, and limited computing power. Deep learning models typically consist of hundreds of layers and millions of parameters, such as YOLOv5s with 214 layers and 7,035,811 parameters. Meanwhile, the running process occupies significant memory resources of the computing platform and requires powerful floating-point computing capabilities.

Terminals in substations are typically installed in fixed locations. The size of the object captured by the camera does not vary significantly. Selecting a model with fewer parameters and computations could improve the detection speed. Therefore, taking advantage of the capacity and speed of PIoT, a lightweight model can be deployed to edge computing terminals with limited resources to energize hardhat-wearing detection tasks.

The rest of this paper is organized as follows: Section II presents a review of recent relevant literature, followed by the introduction of YOLOv5 and MobileNet in Section III. The methodology is presented in Section IV. Section V discusses the results & discussion. Finally, concluding remarks are proposed in Section VI.

II. Related Work

To solve the problems of slow detection speed and low detection accuracy, Xiao et al. [14] proposed a helmet-wearing detection method based on an improved Single Shot MultiBox Detector (SSD) in 2020. The MobileNetV3-small backbone replaced the Visual Geometry Group 16 (VGG) backbone of the SSD detection algorithm to reduce model parameters. Furthermore, the proposed method utilized the Feature Pyramid Network (FPN) structure to combine abstract and detailed shallow features for improved detection accuracy. The proposed method achieved a detection speed of 108 FPS and a 0.5% increase in mAP50 compared to the SSD algorithm.

In 2021, Chen et al. [15] proposed a method for detecting helmet-wearing based on EfficientDet. The proposed method adopted a k-means clustering algorithm and cross-scale connections with weighted feature fusion under different scales to increase the recognition rate and improve real-time performance. The mAP of the model improved by 2%, reaching 87.4%.

Xu et al. [16] proposed a helmet-wearing detection algorithm based on MobileNet-SSD in 2021. The algorithm addresses the challenges of detecting small objects, complex backgrounds, and interferences. The algorithm utilized the lightweight MobileNet architecture, resulting in improved detection speed. Additionally, the authors employed a transfer learning strategy to overcome difficulties in model training. The proposed algorithm provided a detection speed 10.2 times higher than that of the SSD algorithm, albeit with a minor loss in accuracy.

Wu et al. [17] proposed an improved algorithm for detecting correct usage of work clothes and helmets in 2021, which utilized a transformer-based self-attentive coding feature fusion network. A quality Focus loss function was introduced to address the problem of inconsistent inference during the training and testing phases. The detection method achieved a mAP of 44.6% and an average precision (AP) of 79.5%, with a processing speed of 117 frames per second.

In 2021, Zhu et al. [18] proposed an algorithm for detecting safety helmet-wearing based on YOLOv5 by improving methods, such as the candidate box, convolution layer, input, and quantization. The improved YOLOv5 algorithm outperformed the original YOLOv5 in detection accuracy, Intersection over Union (IoU), and detection time.

Ge et al. [19] proposed a method for detecting safety helmet-wearing that improved the accuracy of detecting small objects and reduced accuracy reduction in complex backgrounds in 2022. The proposed method combined high and low-level features to capture more detailed information based on YOLOv4. To lessen the aliasing effect after feature map fusion and ensure feature stability, a 3×3 convolution operation is used on the fused feature maps. The improved model achieved a 4.27% increase in mAP compared to YOLOv4.

In 2022, Qu et al. [20] proposed a safety helmet-wearing detection method for power grid operators based on YOLOv3. The detection accuracy of the YOLOv3 model could reach 92.59%. In addition, the model could detect 15 images per second, which can achieve effective detection in complex operation scenarios.

In 2022, Wang et al. [21] proposed an improved helmet-wearing detection method based on YOLOv5 to address issues such as false detection and missed detection in complex environments for small and dense objects. They integrated a coordinate attention mechanism into the backbone of YOLOv5, resulting in an average accuracy of 95.9%, which increased by 5.1% compared to YOLOv5.

As helmet objects on construction sites are small, CenterNet struggles with small object recognition. In 2022, Zhao et al. [22] proposed the FPN-CenterNet framework, which used an Asymmetric Convolution Network (ACNet) to improve the feature extraction of the backbone. They also employed the Distance IoU (DIoU) loss function to optimize the accuracy of frame prediction. The improved algorithm achieved a mAP increase of 4.99% compared to CenterNet and the FPS reached 25.81.

In 2022, Zhao et al. [23] proposed a real-time object detection method based on YOLOv3 to address the issue of low resolution and intensity contrast in video images. The image was pre-processed using Gamma correction, and the detection speed was improved by deriving the most suitable prior box size based on the K-means++ algorithm. The proposed method achieved an improvement of over 2%.

In 2022, Hayat et al. [24] used the YOLOv5x architecture to train a safety helmet detection model on a benchmark dataset, effectively detecting small and low-light objects. The

YOLOv5x achieved the highest mAP of 92.44% compared to other YOLO architectures.

Although the methods mentioned above improved the algorithm for detecting hardhats, the models had numerous parameters and computations, making them unsuitable for deployment on edge computing terminals. Furthermore, some researchers have utilized open-source datasets, such as the Safety Helmet Detection Dataset [25]. The dataset comprised only three classes, not fully presenting various objects in images. Additionally, the model trained on the dataset exhibited poor detecting performance to occluded and crowded objects. In particular, interfering objects were incorrectly predicted by the model. Consequently, the lightweighting of a model represents a superior solution, while a well-annotated dataset can also improve the robustness of the model.

The main contributions of this paper are:

- Based on the Safety Helmet Detection Dataset, a random background augmentation method is proposed to obtain more background images, which reduces the number of predicted false positive instances and improves detection precision.
- The backbone of the YOLOv5s is replaced with the MobileNetv3 backbone, significantly reducing the number of parameters and computations.
- CBAM is integrated into the network to compensate for the reduction in detection precision. Ablation experiments are conducted to explore the most effective method of integrating CBAM.
- A hardhat-wearing detection architecture covering numerous substation areas is proposed to meet the practical application by exploring PIoT and edge computing technologies.

III. YOLO and MobileNet Algorithms

A. Introduction of YOLO

The YOLO series of one-stage object detection algorithms are known for their high detection speed and precision. In June 2020, YOLOv5 was released as an open-source algorithm on the Internet. YOLOv5s, a small model in the series, has a model file size that is approximately 90% smaller than that of YOLOv4 while maintaining a similar level of accuracy. The YOLOv5 series includes five models, namely YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5n, which are classified based on their model size. The YOLOv5s model consists of three components: backbone, neck, and head. When the input image has a shape of 640 × 640, the backbone extracts feature maps of five different sizes: 320 × 320, 160 × 160, 80 × 80, 40 × 40, and 20 × 20. The neck further extracts features and fuses feature maps from the backbone. The head predicts small, medium, and large objects using three small-size feature maps.

Table I compares the performance of YOLO series models on the different datasets. The YOLOv5s model, which employed a Conv2D (6 × 6) and Cross Stage Partial (CSP) Darknet53 structure, achieved a high accuracy with a mAP50 of 56.8% and the fastest speed with an FPS of 155 on the Common Objects in Context (COCO) dataset. Therefore, this paper used the YOLOv5s algorithm to improve hardhat-wearing detection.

TABLE I. Performance Comparison of the YOLO Series Models

Model	Network	FPS	VOC 2007 (mAP/%)	VOC 2012 (mAP/%)	COCO (mAP50/%)	GPU
YOLOv1 [26]	GoogleNet (modified)	45	66.4	57.9	-	Titan X
YOLOv2 [27]	Darknet19	40	78.6	73.4	44.0	Titan X
YOLOv3 [28]	Darknet53	20	-	-	57.9	Titan X
YOLOv4 [29]	CSPDarknet53	62	-	-	**65.7**	Tesla V100
YOLOv5s	Conv2D (6 × 6) + CSPDarknet53	**155**	-	-	56.8	Tesla V100

B. Introduction of MobileNet

Traditional deep learning-based algorithms require large amounts of graphics memory and many floating-point calculations, making them unsuitable for deployment and operation on devices with limited computing resources. However, the MobileNet has proposed Depth-wise Separable Convolution (DSC) composed of depth-wise and point-wise convolution to replace ordinary convolution, which reduces parameters and improves operation speed [30].

The MobileNetv2 introduced an inverted residual and linear bottleneck structure [31]. It utilized the advantages of depth-wise separable convolution to effectively reduce computations of intermediate convolution operations, ensuring the algorithm's performance and avoiding information loss by removing the Rectified Linear Units (ReLU) activation function. The MobileNetv2 had a parameter size of approximately 6.9 MB and achieved a TOP-1 classification accuracy of 74.7% on the ImageNet dataset. This model was smaller and more accurate than the MobileNetv1.

The MobileNetv3 utilized the Neural Architecture Search (NAS) method to obtain its network structure [32], achieving improved accuracy and efficiency compared to the MobileNetv2. The Hard-Swish activation function replaced the swish activation function. Additionally, a Squeeze-And-Excite module was added to improve accuracy, distinguishing it from v1 and v2.

Fig. 1 displays the structure of MobileNetv3. The input image has a shape of 224 × 224 × 3. It first undergoes a 3 × 3 convolutional layer with a stride of two, followed by a Batch Normalization (BN) layer and the Hard-Swish activation function. Next, the output feature maps from the previous layer pass through 11 or 15 bottleneck structures for feature

extraction. Next, the extracted feature maps are passed through an average pooling layer to reduce their size. After that, the output feature maps are passed through a 1 × 1 convolutional layer, a BN layer, and the Hard-Swish activation function in sequence. The final classification output is obtained through the Fully Connected layer.

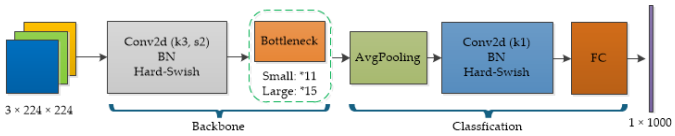


Fig. 1. MobileNetv3 structure. where small and large denote the MobileNetv3-small network and MobileNetv3-large network, respectively

Overall, the MobileNetv3 was chosen to replace the backbone of YOLOv5s in this paper due to its advantages in lightweight.

IV. Methodology

A. Detection Architecture

The cloud center is a cluster of servers with powerful computing capabilities, connecting through fast communication links. Load-balancing technology distributes user requests to multiple active nodes, ensuring redundancy, reducing network congestion and overload, and improving workload distribution. Managers periodically collect images taken by edge computing terminals at power work sites to enrich the original hardhat dataset. These images are then annotated to gradually form a diverse and sufficient dataset. This process allows the trained model to gradually achieve better accuracy. Since the MobileNetv3 backbone replaces the YOLOv5s backbone, the detection model file size becomes smaller, and transmitting the smaller model file to edge computing terminals reduces the transmission consumption of PIoT significantly.

Fig. 2 shows the detection architecture of hardhat-wearing in substation sites. First, the cloud center utilizes powerful servers to train the hardhat-wearing detection model on the hardhat dataset. In addition, the model is transmitted via PIoT to edge computing terminals to perform hardhat-wearing detection tasks. Furthermore, edge computing terminals give workers without hardhats a warning. Finally, edge computing terminals upload the detection results to the cloud center through PIoT.

B. Hardhat-YOLO Structure

The proposed method, hardhat-YOLO, is based on the YOLOv5s and MobileNetv3-small networks. The network structure is shown in Fig. 3. The hardhat-YOLO network consists of three components, similar to YOLOv5s: the backbone for feature extraction, the neck for enhanced feature extraction and feature fusion, and the head for prediction. The improved backbone uses the MobileNetv3-small backbone and CBAMs to extract feature maps. The improved neck comprises the YOLOv5s neck and CBAMs. The head of this model is identical to that of YOLOv5s. Additionally, data augmentation methods, including image distortion, spatial translation, rotation, and copy-and-paste, are employed to enhance the accuracy of the trained model.

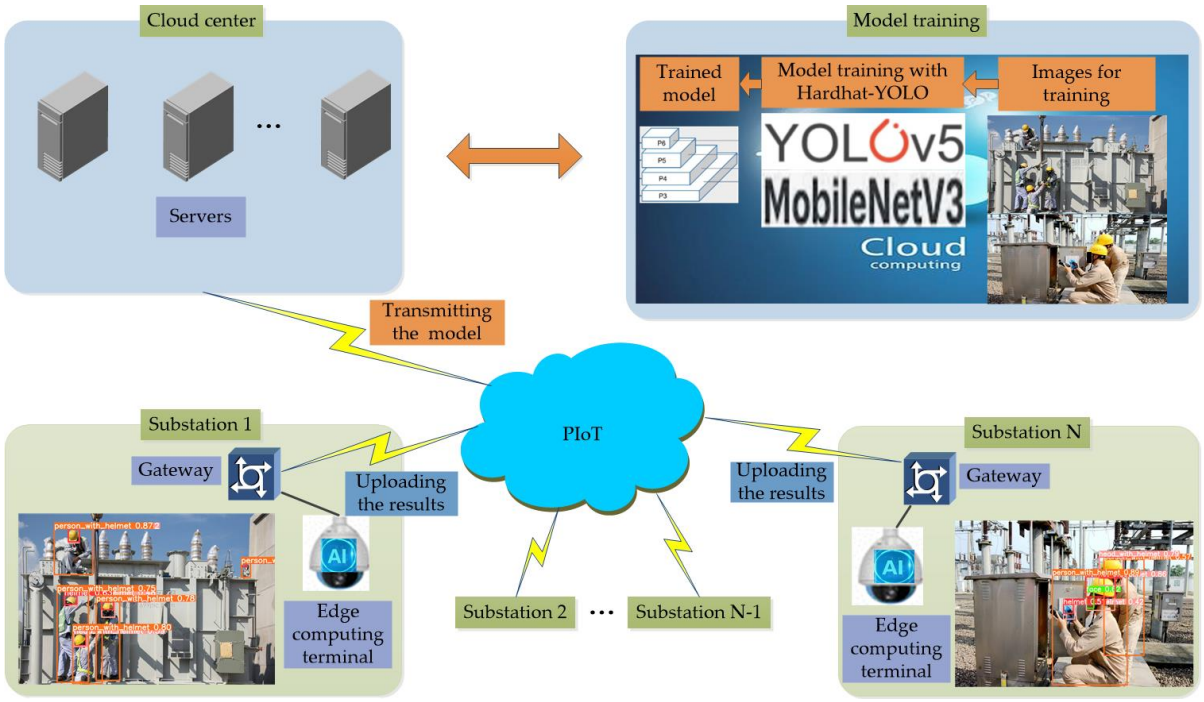


Fig. 2. Hardhat-wearing detection architecture in substation sites

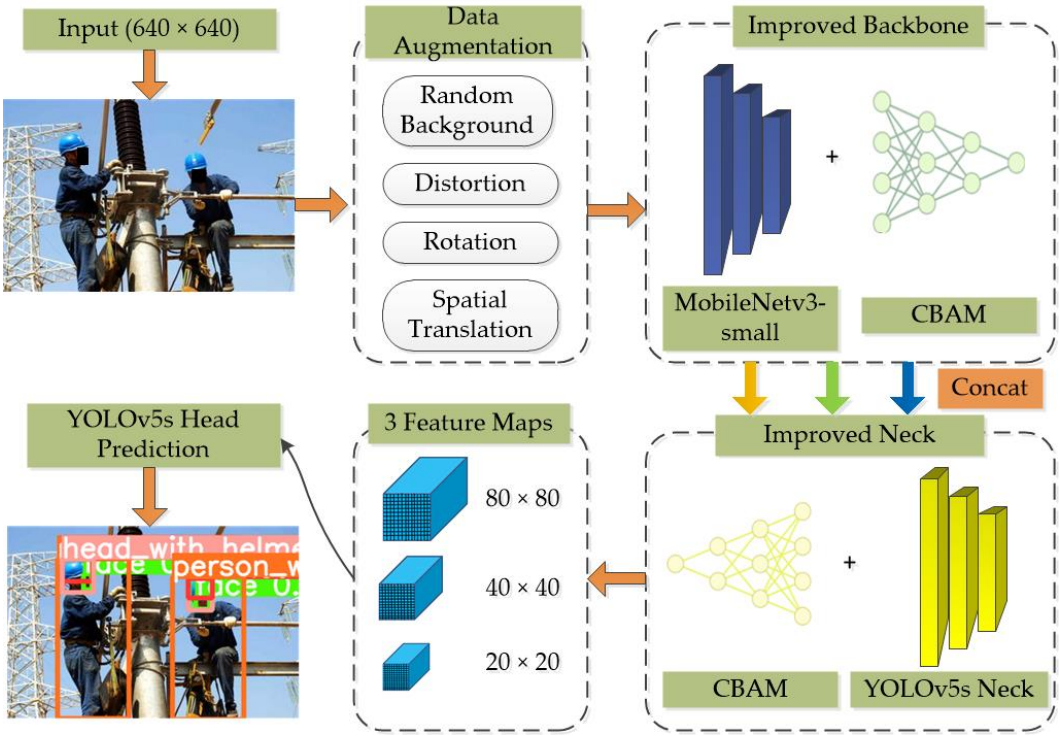


Fig. 3. Hardhat-YOLO structure

C. Data Augmentation

The public hardhat dataset comprises 5000 images classified into three categories: person, head, and helmet. However, the dataset has some issues. First, the dataset was annotated in the PASCAL VOC format without being fully annotated. Therefore, it was reannotated in the YOLO format, which includes six categories: helmet, head_with_helmet, person_with_helmet, head, person_no_helmet, and face, fully presenting various objects in images. Furthermore, the model trained on the dataset struggled to accurately detect interfering and occluded objects, as demonstrated in Fig. 4. Workers with baseball and bamboo hats were incorrectly predicted as "person_with_helmet" objects. Meanwhile, workers behind protective netting and steel bars were missed detection.

To enhance the robustness of the trained model, some images containing interfering, occluded, and long-distance objects were added to the dataset. Additionally, a random background augmentation method is proposed to obtain more images, compensating for the lack of background images. Background images taken from construction and substation sites contained no objects. Therefore, image distortion, spatial translation, rotation, and copy-and-paste methods were randomly utilized to create new background images with original background images. The random background augmentation method is shown in Fig. 5.

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

Fig. 4. Detection results of sample images contained interfering or occluded objects. (**a**) Detection results of a worker with a baseball hat; (**b**) Detection results of a worker with a bamboo hat; (**c**) Detection results of workers behind steel bars; (**d**) Detection results of workers behind protective netting

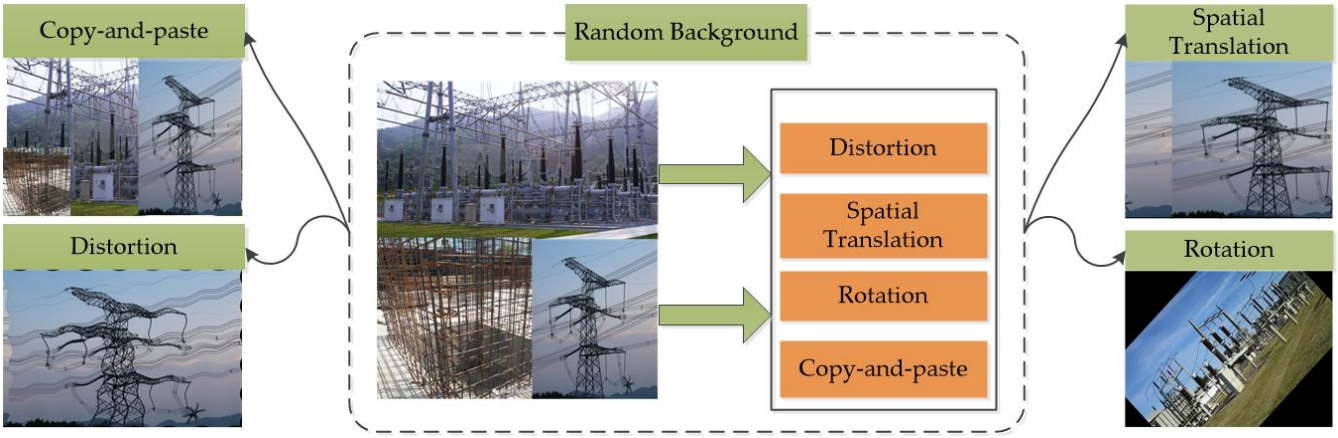


Fig. 5. Random background augmentation method

Finally, the image number of the customized dataset was increased to 6000. Fig. 6 compares the number of labels in three datasets. The public dataset contained 25,501 labels in total. After reannotated public datasets, the number of labels increases significantly. After data augmentation, the number of labels further increased. The customized dataset had 80,149 labels in total. The label numbers for each class were 19,803, 16,756, 16,387, 7015, 6205, and 13,983, respectively.

D. Replacing the YOLOv5s Backbone

The YOLOv5s backbone extracts features from the input image to create three initial feature maps. The feature maps have sizes of 80 × 80, 40 × 40, and 20 × 20, respectively. Therefore, replacing the backbone should ensure that the new backbone can also output three initial feature maps. Fig. 7 illustrates the replacement of the YOLOv5s backbone with the MobileNetv3-small backbone. Specifically, layers 0 to 4, 5 to 6, and 7 to 8 of the YOLOv5s backbone were replaced by layers 0 to 3, 4 to 8, and 9 to 12 of MobileNetv3, respectively. The feature map's shape is presented as height × width × channel. The ConvBNHSwish structure contains a convolutional layer, batch normalization layer, and hard swish activation function. The ConvBNSiLU structure contains a convolutional layer, batch normalization layer, and Sigmoid Linear Unit (SiLU) activation function. The C3 is the feature extraction structure in YOLOv5.

E. Integrating CBAM

Replacing the YOLOv5s backbone resulted in a decrease in model precision. Using an attention mechanism can effectively compensate for a reduction in accuracy. The CBAM structure is lightweight and does not significantly increase the parameters and computations required. The CBAM combined channel and spatial attention modules, which can be readily incorporated into any convolutional neural network (CNN) architecture. Figure 8 illustrates the architecture of CBAM, which consists of channel and spatial attention modules that are applied sequentially to the input feature map. The input feature map is sequentially multiplied by the two attention feature maps to obtain the final output.

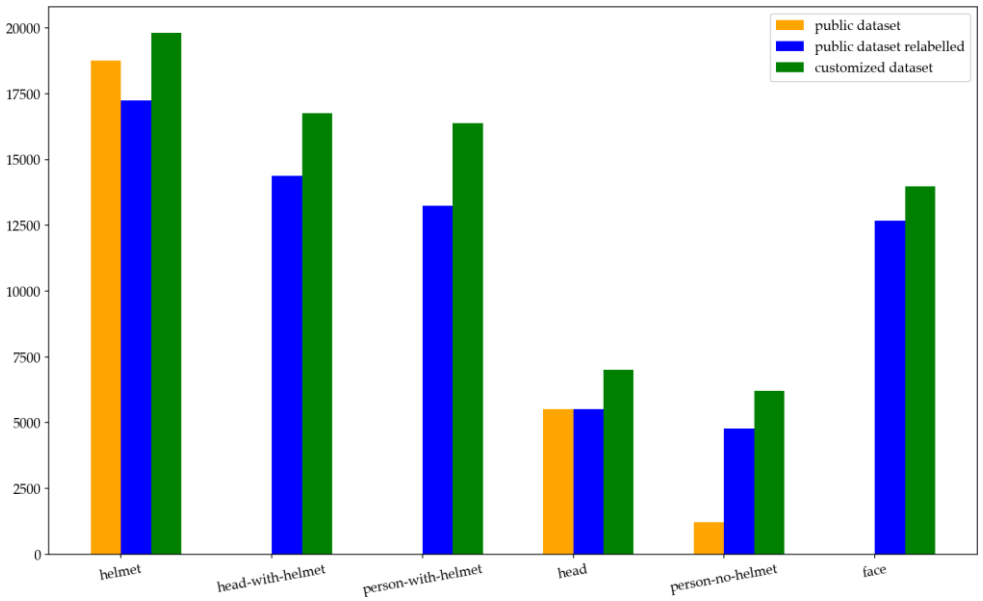


Fig. 6. Comparison of the number of labels in three datasets

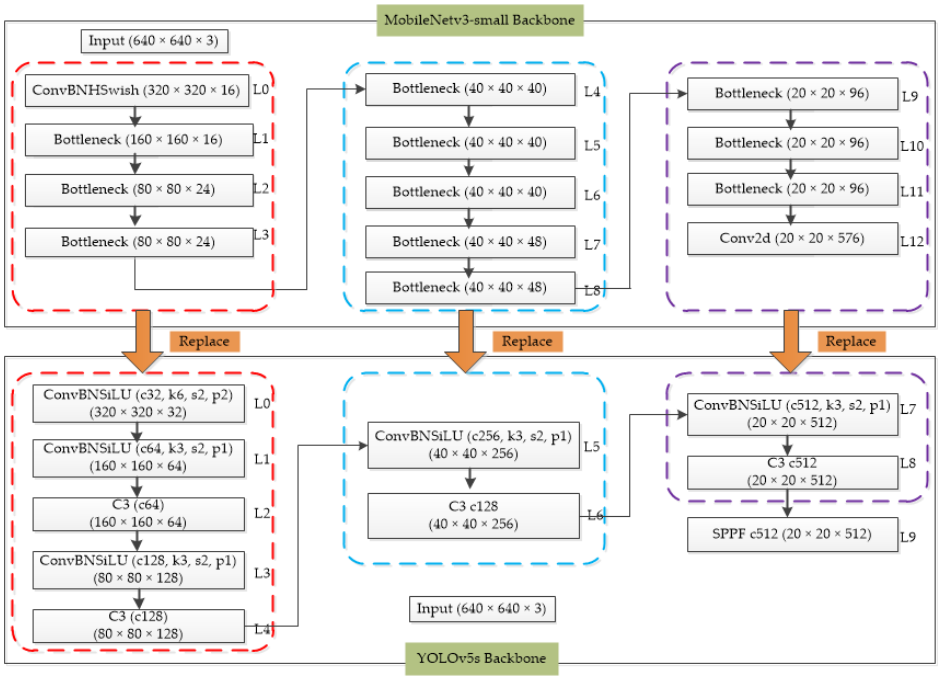


Fig. 7. Replacing the YOLOv5s backbone. Where c denotes channel, k denotes kernel, s denotes stride, and p denotes padding

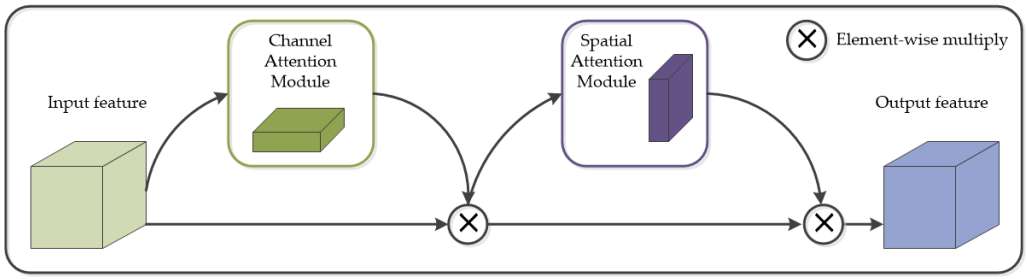


Fig. 8. CBAM architecture

Table II displays the results of four ablation experiments conducted to verify the effective integration of CBAM into the original network. The first method only integrated CBAM into the neck, inserting CBAMs behind the concatenation (40 × 40 × 304) and concatenation (80 × 80 × 152) layers, respectively. The second method only integrated CBAM into the backbone, inserting CBAMs behind three output feature maps of which shapes were 80 × 80 × 24, 40 × 40 × 48, and 20 × 20 × 576, respectively. The third method combined the first and second methods, integrating CBAMs into the backbone and neck. Based on the third method, the fourth method inserted CBAMs additionally behind two output feature maps of which shapes are 320 × 320 × 16 and 160 × 160 × 16, respectively. Where yes denotes that CBAM is integrated behind the feature map, and no is the opposite.

Fig. 9 displays the third method of integrating CBAM, which is the most effective method with the highest mAP50. Five CBAMs are integrated into the original network, in which three CBAMs integrate into the backbone, while two CBAMs integrate into the neck.

TABLE II. METHODS OF INTEGRATING CBAM

Method	Backbone (feature map: 320 × 320 × 16)	Backbone (feature map: 160 × 160 × 16)	Backbone (feature map: 80 × 80 × 24)	Backbone (feature map: 40 × 40 × 48)	Backbone (feature map: 20 × 20 × 576)	Neck (concat: 40 × 40 × 304)	Neck (concat: 80 × 80 × 152)
1	No	No	No	No	No	Yes	Yes
2	No	No	Yes	Yes	Yes	No	No
3	No	No	Yes	Yes	Yes	Yes	Yes
4	Yes	Yes	Yes	Yes	Yes	Yes	Yes

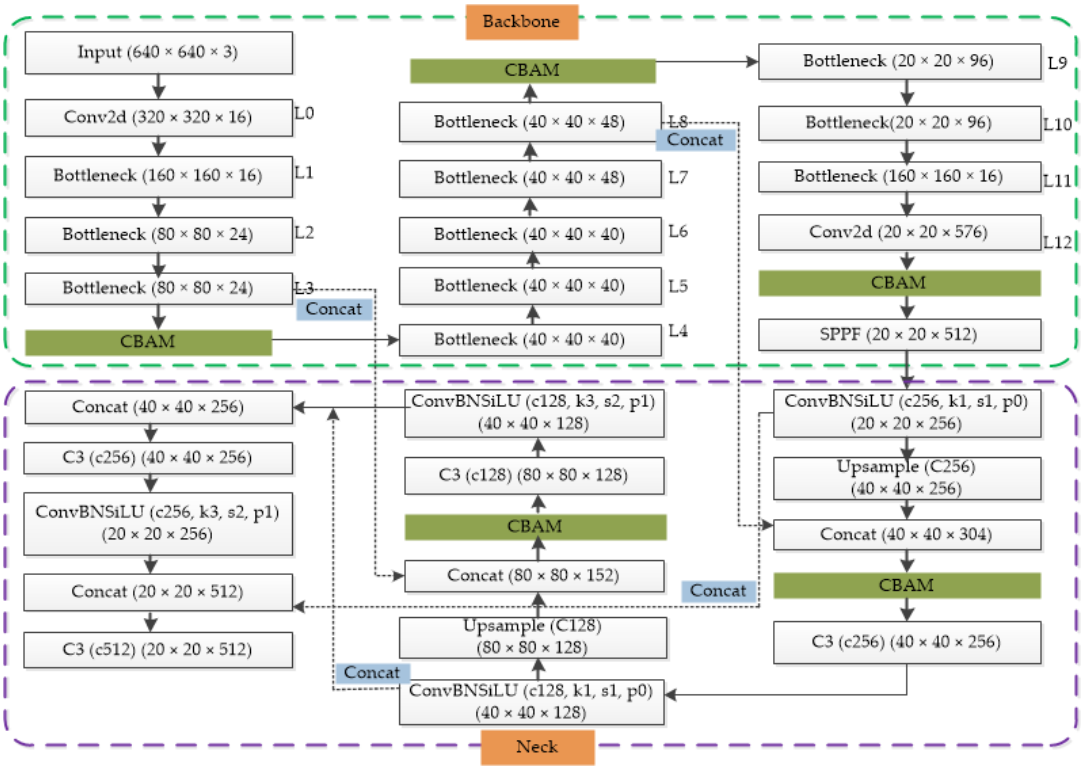


Fig. 9. A sample method of integrating CBAM

V. Experiments and Discussion

A. Experimental Environment

Table III displays the experimental hardware and software. The customized hardhat dataset was divided into a training dataset of 5500 images and a validation dataset of 500 images. The parameters 'img-size', 'batch', and 'epoch' parameters were set to 640, 16, and 300, respectively. The pre-trained weight file of YOLOv5s.pt was used.

TABLE III. Experimental Hardware and Software

Name	Model/Specification	Version
Graphics Processing Unit (GPU)	NVIDIA GeForce RTX 3060 12GB	-
Central Processing Unit (CPU)	Intel Core i7-13700KF 3.4 GHz	-
Random Access Memory (RAM)	32GB	-
Compute Unified Device Architecture (CUDA)	-	11.8
Pytorch	-	2.0.1
Python	-	3.8.17
YOLOv5	-	v7.0-186-g0acc5cf
MobileNet	small	v3

B. Training Results

The validation dataset comprises 7730 instances, of which 'helmet', 'head_with_helmet', 'person_with_helmet', 'head', 'person_no_helmet', and 'face' instances are 2006, 1668, 1524, 595, 473, and 1464 respectively. Precision indicates the detection accuracy for each class. Recall indicates the detection completeness for each class. The mAP50 is the mean average precision calculated at a threshold of 0.50 IoU, which is a key metric for evaluating model detection accuracy. The mAP50-95 means the mean average precision across IoU thresholds ranging from 0.5 to 0.95.

Table IV presents the training results of the YOLOv5s model trained by the YOLOv5s algorithm, including Precision, Recall, mAP50, and mAP50-95 for all classes. The mAP50 for each class was 0.883, 0.911, 0.915, 0.886, 0.889, and 0.783, respectively.

TABLE IV. Training Results of the YOLOv5s Model

Class	Instances	Precision	Recall	mAP50	mAP50-95
all	7,730	0.888	0.825	**0.878**	0.545
helmet	2,006	0.945	0.817	0.883	0.532
head with helmet	1,668	0.928	0.834	0.911	0.587
person with helmet	1,524	0.884	0.884	0.915	0.654
head	595	0.898	0.848	0.886	0.54
person no helmet	473	0.826	0.841	0.889	0.625
face	1,464	0.848	0.725	0.783	0.33

Table V displays the training results of the YOLOv5s-M3s model trained by the YOLOv5s network with the MobileNetv3 backbone. The above metrics for all classes were 0.878, 0.746, 0.828, and 0.476. The mAP50 for each class was 0.836, 0.887, 0.877, 0.835, 0.824, and 0.711, respectively.

TABLE V. TRAINING RESULTS OF THE YOLOV5-M3S MODEL

Class	Instances	Precision	Recall	mAP50	mAP50-95
all	7,730	0.878	0.746	**0.828**	0.476
helmet	2,006	0.926	0.743	0.836	0.477
head with helmet	1,668	0.944	0.768	0.887	0.549
person with helmet	1,524	0.873	0.814	0.877	0.555
head	595	0.896	0.769	0.835	0.481
person no helmet	473	0.791	0.758	0.824	0.514
face	1,464	0.839	0.621	0.711	0.278

The models obtained from the four ablation experiments paid distinct attention to different classes of objects due to the different locations of the integrated CBAMs. Table VI compares training results for the four ablation experiments integrating CBAM. The mAP50 for all classes of the four models were 0.829, 0.826, 0.833, and 0.830, respectively. Method 3 integrated three CBAMs into the backbone and two CBAMs into the neck, resulting in the highest mAP50 of 0.833. Consequently, the hardhat-YOLO model was trained using this method.

TABLE VI. COMPARISON OF TRAINING RESULTS FOR FOUR ABLATION EXPERIMENTS INTEGRATING CBAM

Class	mAP50 method 1	mAP50 method 2	mAP50 method 3: hardhat-YOLO	mAP50 method 4
all	0.829	0.826	**0.833**	0.83
helmet	0.84	0.841	0.849	0.844
head with helmet	0.884	0.886	0.887	0.892
person with helmet	0.875	0.876	0.873	0.877
head	0.839	0.833	0.841	0.839
person no helmet	0.827	0.824	0.831	0.834
face	0.711	0.698	0.714	0.696

Fig. 10 shows the training Precision-Recall curves for four ablation experiments, further demonstrating that the third method achieved the highest mAP50 than the other methods.

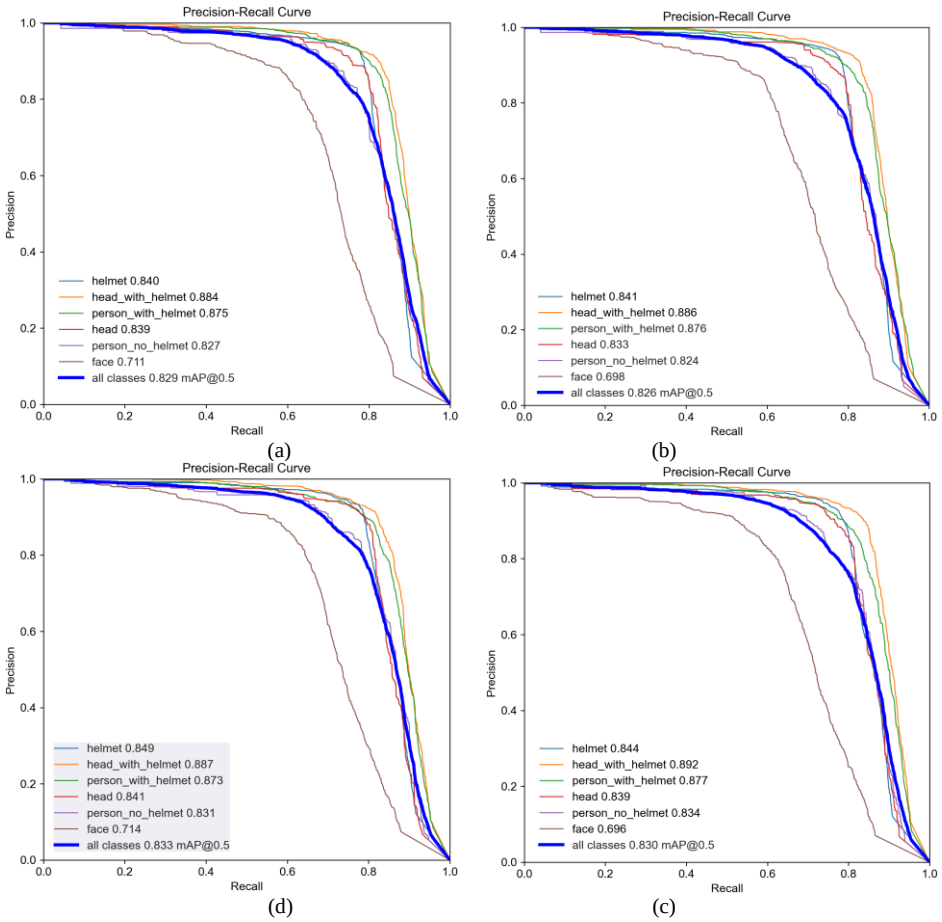


Fig. 10. Precision-Recall curves of four ablation experiments. (a) Precision-Recall curve of method 1 with a mAP50 of 82.9%; (b) Precision-Recall curve of method 2 with a mAP50 of 82.6%; (c) Precision-Recall curve of method 3, hardhat-YOLO, with a mAP50 of 83.3%; (d) Precision-Recall curve of method 4 with a mAP50 of 83.0%

Fig. 11 compares the Precision, Recall, mAP50, and mAP50-95 metrics of three models for all classes. The YOLOv5s model achieved the best performance on all metrics, with a mAP50 of 0.878. After replacing the backbone, the mAP50 of the YOLOv5s-m3s model decreased by 5% compared to the YOLOv5s model, reaching 0.828. After integrating CBAM, the mAP50 of the hardhat-YOLO model increased to 0.833, which is 0.5 percentage points higher than the YOLOv5s-m3s model. The hardhat-YOLO model with CBAMs improved the mAP50 of other classes by reducing the mAP50 of the 'person_with_helmet' class. Specifically, the mAP50 of the 'helmet', 'head', and 'person_no_helmet' classes increased by 1.3%, 0.6%, and 0.7%, respectively, compared to the YOLOv5s-M3s model.

C. Validation Results

Images and videos from substation sites were used to validate the detection effectiveness and speed of the hardhat-YOLO model. The model predicted the results by inputting images and videos, with each object having a bounding box with a confidence value.

The model can detect various media types, including images, videos, cameras, and video streams. The image formats supported include Portable Network Graphics (PNG) and Joint Photographic Experts Group (JPEG), while the video formats are Moving Picture Experts Group-4 (MP4).

a) Effectiveness Validation: Fig. 12 displays the detection results of four sample images. The colors of bounding boxes with confidence values are orange if workers are wearing hardhats and yellow if not. The hardhat-YOLO model accurately predicted all 'person_with_helmet', 'helmet', 'head_with_helmet', and 'face' objects in Fig. 12(a) and 12(b). Furthermore, the model correctly identified a worker wearing a baseball hat in Fig. 12(c) as a 'person_no_helmet' object. Fig. 12(d) shows a correctly predicted 'person_with_helmet' object behind protective netting.

Fig. 13 displays the real-time prediction results of the video captured by a camera. The hardhat-YOLO model accurately predicted all objects when the worker wore and removed a hardhat.

Fig. 14 shows the prediction results of a sample video. The model can correctly detect whether or not the two workers in the video are wearing hardhats.

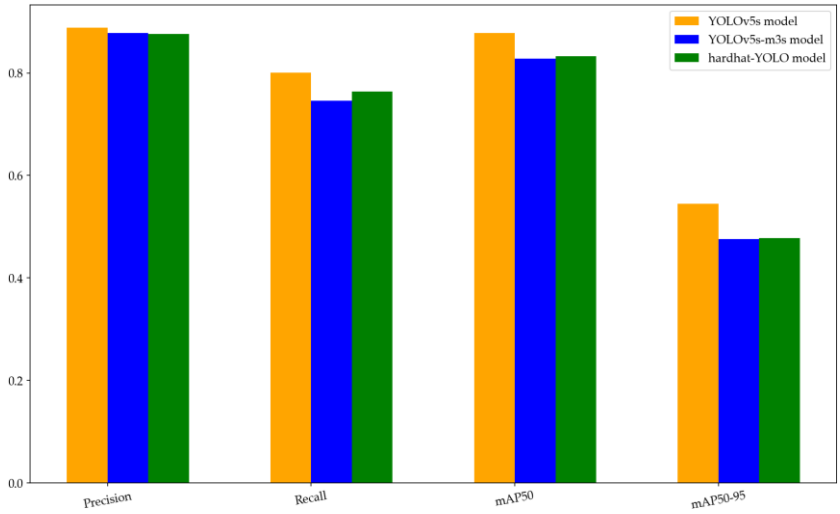
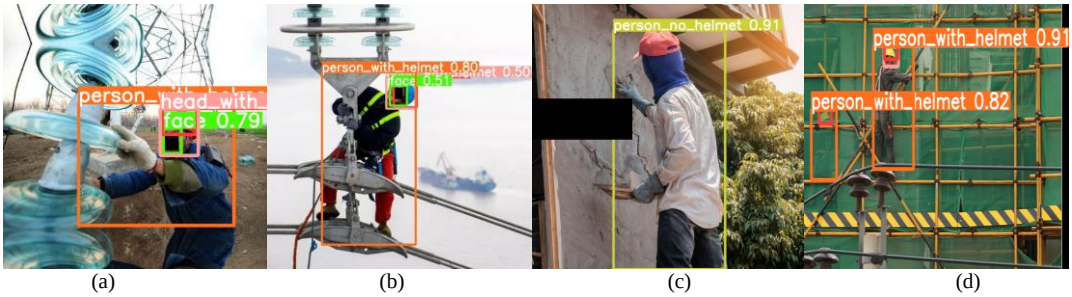
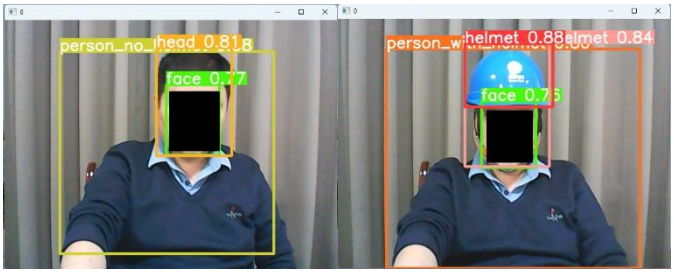


Fig. 11. Four metrics comparisons of the three models



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

Fig. 12. Prediction results of sample images. (**a**) Prediction results of image 1; (**b**) Prediction results of image 2; (**c**) Prediction results of image 3; (**d**) Prediction results of image 4



(a) (b)

Fig. 13. Prediction results from a camera. (**a**) Prediction results of a worker with a hardhat; (**b**) Prediction results of a worker without a hardhat

(a) (b)

Fig. 14. Prediction results of a sample video. (**a**) Prediction results of two workers not wearing hardhats; (**b**) Prediction results of two workers wearing hardhats

b) Speed Validation: The Parameters metric refers to the amount of graphic memory the model requires. The GFLOPs metric refers to the number of computations the model inference requires. The Parameters, GFLOPs, and mAP50 metrics of the three models are shown in Table VII. The hardhat-YOLO model had 4,533,682 parameters, decreasing by approximately 35.5% compared to the YOLOv5s model with 7,026,307 parameters. After integrating CBAM, the hardhat-YOLO model parameters increased by only about 1.2% compared to the YOLOv5s-M3s model. However, the mAP50 of all classes increased by 0.5%. The GFLOPs of the hardhat-YOLO model decreased by about 54.4% compared to the YOLOv5s model with 15.8 GFLOPs. After integrating CBAM, the GFLOPs of the hardhat-YOLO model only slightly increased by about 0.1 compared to the YOLOv5-m3s model. Although the mAP50 of hardhat-YOLO decreased by 4.5% compared to the YOLOv5s model, the number of parameters and computations were significantly reduced.

TABLE VII. THE PARAMETERS, GFLOPS, AND MAP50 METRICS COMPARISON OF THE THREE MODELS

Model	Parameters	GFLOPs	mAP50 (all classes)
YOLOv5s	7,026,307	15.8	0.878
YOLOv5s-M3s	4,477,091 (-36.3%)	7.1 (-55%)	0.828 (-5%)
hardhat-YOLO	4,533,682 (**-35.5%**)	7.2 (**-54.4%**)	0.833 (-4.5%)

The detection speed of the three models was evaluated using the images from the validation dataset. Latency is the forward propagation time of a model, which refers to the time it takes for a model to predict an image or video. It includes the time spent in pre-processing, inference, and Non-Maximum Suppression (NMS) processes. FPS is the reciprocal of latency, which measures the average detection speed per image with higher values indicating faster detection.

Table VIII compares the pre-process, inference, NMS, latency, and FPS metrics of the three models. The hardhat-YOLO model achieved an FPS of 172.4, which increased by 17.3% compared to the YOLOv5s model with an FPS of 147. After integrating CBAM with fewer parameters and computations, the FPS of the hardhat-YOLO model decreased slightly compared to the YOLOv5s-M3s model with an FPS of 178.6. Where ms denotes millisecond.

TABLE VIII. DETECTION SPEED COMPARISON OF THE THREE MODELS

Model	Pre-process (ms)	Inference (ms)	NMS (ms)	Latency (ms)	FPS
YOLOv5s	0.3	4.4	2.1	6.8	147
YOLOv5s-M3s	0.3	3.2	2.1	5.6	178.6
hardhat-YOLO	0.3	3.8	1.7	5.8	172.4 (**+17.3%**)

D. Discussion

Comparing the effectiveness and speed of the three models, the hardhat-YOLO model achieved a good balance between accuracy and speed. As a result, the model is easily deployable on substation terminals for hardhat-wearing detection. This paper employs experimental data to assess the usability of the model. However, it does not deploy the model to edge computing terminals to verify its usability. This is a limitation of the paper.

VI. Conclusions

This paper proposes a lightweight model, hardhat-YOLO, customized for hardhat-wearing detection. To improve the accuracy and robustness of the model, a random background augmentation method is introduced to obtain more background images and images from complex work sites are added to the original dataset. The MobileNetv3-small backbone replaces the YOLOv5s backbone, reducing the parameters and computations. The CBAM has been effectively integrated into the network to enhance detection precision with a slight increase in parameters and computations. The proposed model has fewer parameters, fewer GFLOPs, fast speed, and a small file size, resulting in suitable precision. The smaller model is transmitted to the edge computing terminals through PIoT, significantly reducing bandwidth consumption. The validation results demonstrate that the proposed model achieves appropriate precision and fast detection speed. Compared to the original YOLOv5s model, the proposed model has slightly lower accuracy but significantly improved lightweight level and detection speed. As a result, the lightweight hardhat-YOLO model is suitable for practical hardhat-wearing detection in substation sites. Future works should consider the deployment of the deep learning-based model. Utilizing model branch reduction and knowledge distillation further reduces the parameters and computations of the model.

References

[1] J. Wang, H. Zhou, H. Sun, Z. Su, and X. Li, "A Violation Behaviors Detection Method for Substation Operators Based on YOLOv5 and Pose Estimation," In *Proceedings of the 2022 IEEE 3rd China International Youth Conference on Electrical Engineering (CIYCEE)*, Wuhan, China, 3-5 November 2022, pp. 1–5. Available: https://doi.org/10.1109/CIYCEE55749.2022.9958961

[2] J. Li, H. Liu, T. Wang, M. Jiang, S. Wang, K. Li, and X. Zhao, "Safety Helmet Wearing Detection based on Image Processing and Machine Learning," In *Proceedings of the 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, Doha, Qatar, 4-6 February 2017, pp. 201–205. Available: https://doi.org/10.1109/ICACI.2017.7974509

[3] J. Cui, D. Wang, H. Li, W. Zhang, J. Zhang, and G. Zhang, "Lightweight of Intelligent Real-Time Detection Model Based on YOLO-v4," In *Proceedings of the 2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)*, Wuhan, China, 19-21 August 2022, pp. 244–247. Available: https://doi.org/10.1109/ICFEICT57213.2022.00052

[4] W. Jia, S. Xu, Z. Liang, Y. Zhao, H. Min, S. Li, and Y. Yu, "Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector," *IET Image Processing*, vol. 15, pp. 3623-3637, 2021. Available: https://doi.org/10.1049/ipr2.12295

[5] S. W. Wang, "Design and Implementation of Abnormal Behavior Detection System for Power Operation," Master, Wuhan Textile University, Wuhan, 2021. Available: https://doi.org/10.27698/d.cnki.gwhxj.2021.000120

[6] H. F. Zheng, "Research on Video Recognition of Safety Protection Measures for Electric Power Construction Personnel", Master, Guangdong University of Technology, Guangdong, 2021. Available: https://doi.org/10.27029/d.cnki.ggdgu.2021.000180

[7] Q. Wang and Y. G. Wang, "Research on Power Internet of Things Architecture for Smart Grid Demand," In *Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Beijing, China, 20-22 October 2018, pp. 1–9. Available: https://doi.org/10.1109/EI2.2018.8582132

[8] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K. C. Wang, "Review of Internet of Things (IoT) in electric power and energy systems," *IEEE Internet of Things Journal*, vol. 5, pp. 847-870, 2018. Available: https://doi.org/10.1109/JIOT.2018.2802704

[9] L. de MBA Dib, V. Fernandes, M. D. L. Filomeno, and M. V. Ribeiro, "Hybrid PLC/Wireless communication for smart grids and Internet of Things applications," *IEEE Internet of Things Journal*, vol. 5, pp. 655-667, 2018. Available: https://doi.org/10.1109/JIOT.2017.2764747

[10] K. W. Choi, A. A. Aziz, D. Setiawan, N. M. Tran, L. Ginting, and D. I. Kim, "Distributed wireless power transfer system for Internet-of-Things devices," *IEEE Internet of Things Journal*, vol. 5, pp. 847-870, 2018. Available: https://doi.org/10.1109/JIOT.2018.2790578

[11] N. Nezamoddini and Y. Wang, "Risk management and participation planning of electric vehicles in smart grids for demand response," *Energy*, vol. 116, pp. 836-850, 2016. Available: https://doi.org/10.1016/j.energy.2016.10.002

[12] W. S. Shi, H. Sun, J. Cao, Q. Zhang, and W. Liu, "Edge computing - an emerging computing model for the Internet of Everything era," *Journal of Computer Research and Development*, vol. 54, pp. 907–924, 2017. Available: https://doi.org/10.7544/issn1000-1239.2017.20160941

[13] W. S. Shi, X. Z. Zhang, Y. F. Wang, and Q. Y. Zhang, "Edge computing: state-of-the-art and future directions," *Journal of Computer Research and Development*, vol. 56, pp. 69–89, 2019. Available: https://doi.org/10.7544/issn1000-1239.2019.20180760

[14] T. G. Xiao, L.Q. Cai, K. Y. Tang, X. Gao, and C. Y. Zhang, "Improved SSD's helmet wearing detection method," *Journal of Sichuan University of Science & Engineering (Natural Science Edition)*, vol. 33, pp. 68-76, 2020. Available: https://doi.org/10.11863/j.suse.2020.04.10

[15] Z. T. Chen, K. M. Yin, Y. Zhang, R. Z. Jin, W. Y. Zhi, and C. F. Shen, "The research of safety helmet-wearing detection based on EfficientDet," *Information Technology & Standardization*, vol. Z1, pp. 19-23+29, 2021.

[16] X. F. Xu, W. F. Zhao, H. Q. Zhou, L. Zhang, and Z. Y. Pan, "Detection algorithm of safety helmet wear based on MobileNet-SSD," *Computer Engineering*, vol. 47, pp. 298-305+313, 2021. Available: https://doi.org/10.19678/j.issn.1000-3428.0058733

[17] H. Y. Wu, J. S. Lei, L. F. Chen, and S. Y. Yang, "Improved detection algorithm and its application in safety control in substation scenario," *Computer Engineering and Applications*, vol. 58, pp. 313-320, 2022. Available: https://doi.org/10.3778/j.issn.1002-8331.2107-0005

[18] X. C. Zhu and Z. T. Chen, "Safety helmet wearing detection based on improved YOLOv5," *Journal of Nanjing Institute of Technology (Natural Science Edition)*, vol. 19, pp. 7-11, 2021. Available: https://doi.org/10.13960/j.issn.1672-2558.2021.04.002

[19] Q. Q. Ge, Z. J. Zhang, L. Yuan, X. M. Li, and J. M. Sun, "Safety helmet wearing detection method of fusing environmental features and improved YOLOv4," *Journal of Image and Graphics*, vol. 26, pp. 2904-2917, 2021. Available: https://doi.org/10.11834/jig.200606

[20] W. Q. Qu, Z. B. Qiu, C. B. Liao, and X. Zhu, "Detection on safety helmet wearing of power grid operators based on YOLOv3," *Journal of Safety Science and Technology*, vol. 18, pp. 214-219, 2022.

[21] L. M. Wang, J. Duan, and L. W. Xin, "YOLOv5 helmet wear detection method with introduction of attention mechanism," *Computer Engineering and Applications*, vol. 58, pp. 303-312, 2022. Available: https://doi.org/10.3778/j.issn.1002-8331.2112-0242

[22] J. H. Zhao, H. R. Wang, and L. Wu, "FPN-Centernet helmet wearing detection algorithm," *Computer Engineering and Applications*, vol. 58, pp. 114-120, 2022. Available: https://doi.org/10.3778/j.issn.1002-8331.2202-0181

[23] L. J. Zhao, R. X. Zhuang, H. Wang, H. W. Yao, and N. Liu, "Intelligent detection of safety protection equipment of power substation based on

improved YOLOv3 algorithm," *Electric Power Science and Engineering*, vol. 38, pp. 1-8, 2022. Available: https://doi.org/10.3969/j.ISSN.1672-0792.2022.05.001

[24] A. Hayat and F. Morgado-Dias, "Deep learning-based automatic safety helmet detection system for construction safety," *Applied Sciences*, vol. 12, pp. 8268-8282, 2022. Available: https://doi.org/10.3390/app12168268

[25] Safety Helmet Detection, Available online: https://www.kaggle.com/andrewmvd/hard-hat-detection (accessed on 4 January 2024).

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016, pp. 779–788. Available: https://doi.org/10.48550/arXiv.1506.02640

[27] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21-26 Jul 2017, pp. 6517–6525. Available: https://doi.org/10.1109/CVPR.2017.690

[28] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv*, 2018. Available: https://doi.org/10.48550/arXiv.1804.02767

[29] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020. Available: https://doi.org/10.48550/arXiv.2004.10934

[30] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, W. J. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv*, 2017. Available: https://doi.org/10.48550/arXiv.1704.04861

[31] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, 18-22 June 2018, pp. 4510–4520. Available: https://doi.org/10.48550/arXiv.1801.04381

[32] A. G. Howard, M. Sandler, and G. Chu, "Searching for MobileNetV3," In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea, 27 October-3 November 2019, pp. 1314–1324. Available: https://doi.org/10.48550/arXiv.1905.02244