Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.040475

ARTICLE

MG-YOLOv5s: A Faster and Stronger Helmet Detection Algorithm

Zerui Xiao, Wei Liu, Zhiwei Ye*, Jiatang Yuan and Shishi Liu

College of Computer Science, Hubei University of Technology, Wuhan, 430000, China

*Corresponding Author: Zhiwei Ye. Email: weizhiye121@126.com

Received: 20 March 2023 Accepted: 23 May 2023 Published: 17 July 2024

ABSTRACT

Nowadays, construction site safety accidents are frequent, and wearing safety helmets is essential to prevent head injuries caused by object collisions and falls. However, existing helmet detection algorithms have several drawbacks, including a complex structure with many parameters, high calculation volume, and poor detection of small helmets, making deployment on embedded or mobile devices difficult. To address these challenges, this paper proposes a YOLOv5-based multi-head detection safety helmet detection algorithm that is faster and more robust for detecting helmets on construction sites. By replacing the traditional DarkNet backbone network of YOLOv5s with a new backbone network composed of RepStemBlock and MG, introducing the C3STR multi-head self-attention mechanism and adding a small detection head P6 to replace the original feature pyramid network, the improved model reduces parameters by 44.1%, responds faster to inference, significantly outperforms the original model in accuracy and mAP (2.3% and 2.7% improvement). Two models are presented in this paper: a large model (YOLOv5s-hat) for optimal performance, mAP@0.5 is 0.957, and a small model (MG-YOLOv5s) for real-time detection on embedded or mobile devices. Compared to large models the lightweight model has 44% fewer parameters, mAP@0.5 of 0.954 and an Fps of 67. Overall, the proposed algorithm shows promise for improving helmet detection on construction sites, with the potential for deployment on a range of devices.

KEYWORDS

Helmet-wearing detection; YOLOv5; lightweight model; object detection

1 Introduction

In recent years, China has faced economic challenges in the form of shrinking exports and sluggish investment demand, which have hindered its development. Despite the significance of investment, export, and domestic demand in boosting the economy, expanding domestic demand has become the primary driver of economic growth. In response to this situation, the government has proposed accelerating urbanization to stimulate rural consumption and expand the rural consumption market.

However, the construction industry, which is indispensable to urbanization, is fraught with significant hazards. With the rise of more high-rise buildings, the safety vigilance of construction workers has not kept pace, resulting in an increasing number of casualties. Construction work is inherently risky, and workers are particularly vulnerable to injuries, with the most prevalent casualties caused by the failure to wear helmets. Therefore, it is crucial to strictly supervise construction workers to ensure that they wear helmets and other protective gear.

According to Eurostat and IBS statistics for 2020, while head injuries account for only about 7% of non-fatal accidents, they are responsible for over 30% of fatal accidents. It is therefore imperative that safety measures for construction workers, including wearing safety helmets, are subject to more stringent supervision and regulation. Attention to the safety of construction workers should not only be limited to the construction code, but there should be a greater emphasis on the implementation of safety measures such as wearing safety helmets and other protective gear.

Wearing a hard hat in the construction industry is crucial to protecting the personal safety of the staff working in this high-risk industry. Hard hats can effectively and promptly protect the construction personnel's head and prevent or reduce head injuries caused by falling objects or site collapses. However, traditional measures for regulating workers' wearing of hard hats, such as human patrols, are becoming less effective due to the increasing number of construction workers, larger buildings, and more complex floors. This traditional method can easily lead to misjudgments, visually fatigued personnel, and consuming a lot of financial and material resources in construction operations.

Therefore, with the development of computer vision technology, there is an urgent need to explore more efficient and effective methods of supervision. Helmet-wearing detection based on computer vision is an essential research task in the field of intelligent site safety. The goal is to detect whether construction workers on site are wearing helmets. In recent years, with the rapid development of the economy and the increase of construction projects, wearing helmets has become a necessary protective measure for each worker. However, some workers lack awareness of the importance of personal protective equipment and safety, making it necessary to strengthen their protection.

To address these issues, this paper proposes a deep-learning model based on YOLOv5 for helmet-wearing detection. The deep learning model can replace human eye discrimination and has a high recognition rate, which makes it an efficient and effective way to monitor worker safety. With the application of target detection in the field of industrial security, the construction industry can benefit from intelligent development and ensure the safe and sustainable development of the industry.

Traditional helmet detection algorithms often face challenges in working environments that are complex and variable, which limits their generalization and robustness, making them unsuitable for deployment on platforms with low computing power. To overcome these challenges, this paper proposes a helmet detection algorithm based on YOLOv5, which addresses the shortcomings of existing methods. The proposed algorithm features the following research contributions:

- A RepStemBlock module is designed to replace the original Focus module, strengthening the connection between the network and the global feature information and improving the network's generalization capability.
- The downsampling part of the model is replaced by the MG module, which significantly reduces the model size and the number of parameters without affecting the connection between the network and the global feature information.
- The original C3 module in the last layer is replaced by the C3STR module, enhancing the overall feature extraction capability of the network.
- A P6 module is designed to improve the detection of small helmets and enhance the model's performance.

The remaining sections of this paper are organized as follows: Section 2 discusses related work on helmet detection. Section 3 describes the proposed model's framework and implementation details.

Section 4 presents experimental results that demonstrate the effectiveness of the proposed algorithm. Finally, Section 5 concludes the paper.

2 Overall Network Framework

The conventional approach to target detection involves three main stages: region selection, feature extraction, and classification using a machine learning-based classifier. Initially, an exhaustive strategy is adopted to traverse the image using sliding windows and to set different sizes and aspect ratios for region selection. Feature extraction and feature vector matching are performed on a massive feature database, followed by inputting the matched features into a traditional classifier to obtain the outcome. Nonetheless, this method has several limitations. Firstly, the time complexity of region selection technology is high. Secondly, traditional feature extraction techniques such as SIFI feature extraction have low real-time performance and require constant subsampling and interpolation operations, which may not accurately extract features of objects with smooth edges. Therefore, their application scope is restricted, and they are not universal. Finally, conventional classifiers such as random forests cannot produce continuous output, and when executing regression, they are unable to make predictions beyond the range of training set data, leading to overfitting when modeling data with specific noise. As a result, conventional target detection approaches cannot effectively solve the helmet-wearing detection problem.

The main objective of target detection algorithms is to extract valuable semantic information from images and develop efficient computational models and techniques that can be widely applied in practical scenarios, such as embedded devices, autonomous driving, aerial object detection, text detection, surveillance, rescue operations, robotics, face detection, pedestrian detection, visual search engines, computation of objects of interest, brand detection, and many more [1,2]. Many researchers, both domestically and internationally, have used machine learning-based helmet detection methods, including skin color detection [3] to locate the face region and obtain the image of the region above the face, using Hu moments as the feature vector of the image, and then using support vector machines (SVM) to detect whether workers are wearing helmets. Other methods include using background subtraction [4] and HOG integral maps to locate the head region, enhancing the image while maintaining the hue and saturation information to a certain extent, and calculating the color features of the helmet to determine whether it is worn. Li et al. [5] proposed a method for locating the head region and calculating the color features of the helmet to detect whether the worker is wearing a helmet. There is also a combination of using image feature information and directional gradient histogram (HOG) for detecting construction workers [6] and color features and circular Hough transform (CHT) to detect whether workers are wearing helmets. In addition, some studies have used support vector machines (SVM) to train the head region image data of motorcycle riders to detect whether they are wearing helmets [7].

All of the above-mentioned traditional helmet detection methods are based on manually selected features for classification and detection, and the features are computationally slow to check and not very accurate. The detection environment is relatively simple, and they cannot adapt to complex and changing construction scenarios. Deep learning (DL) was introduced in the early 2000 s after support vector machines (SVM), multilayer perceptron (MLP), artificial neural networks (ANN), and other shallower neural networks became popular [8]. DL has become a popular and effective method for helmet detection because it can automatically learn features and representations from large amounts of data, without the need for manual feature selection.

Deep learning is a subfield of machine learning that utilizes multi-layer neural networks to learn representations of data and make predictions. Deep learning-based target detection methods have been shown to produce better detection results than traditional methods. As the technology continues to advance, deep learning-based target detection methods are becoming the mainstream algorithm in the field of industrial security, and researchers worldwide are increasingly applying deep learning technology to the detection of safety helmets.

There are two main types of deep learning-based target detection algorithms: two-stage (also known as candidate box-based) and one-stage (based on regression). The two-stage algorithms, such as the R-CNN [9] series (e.g., fast R-CNN [10] and faster R-CNN [11]), generate a series of sparse candidate frames through a convolutional neural network (CNN) and then classify and regress the candidate frames. This approach involves finding the foreground to obtain candidate frames and then performing detection within those frames. While this method can produce highly accurate results, it is not suitable for real-time applications.

In contrast, one-stage detection algorithms based on regression do not require foreground and background separation and only need one forward pass of the network to obtain the target location and classification results. This approach is much faster than the two-stage detection algorithm and can produce fast and accurate detection results. Some of the representative algorithms in this category are YOLO series (including YOLOv1 [12], YOLOv2 [13], YOLOv3 [14], YOLOv4 [15], and YOLOv5 [16]), Single Shot Detector (SSD) [17], Deconvolution Single Shot Detector (DSSD) [18], M2Det [19], RefineDet++ [20], and RetinaNet [21].

YOLOv1 and YOLOX are anchor free methods, YOLOv2, YOLOv3, all the way up to YOLOv5 are anchor base methods. YOLO v1 divides images of $416 * 416$ into grids of $7 * 7$. By default, each grid returns two objects, that is, a vector of length $S * (B * 5 + C)$ is predicted finally, where $s = 7$, $b = 2$, c is the number of categories of class. The reason why YOLOX, also the anchor free method, has better performance than YOLOv1 is that YOLOX is similar to centernet, which makes the prediction of model more accurate by predicting the center point. For YOLOX, there is a featuremap that shows the probability that a location is a center point, which is a better indicator of an object's features than its length and width. YOLOv2, because of the emergence of faster R-CNN at this time, the last layer of featuremap in faster R-CNN can return 9 anchors at each location. As a prior knowledge, anchor plays a huge role, so it is introduced into YOLOv2. Because of the existence of anchor, To some extent, it solves the problem that YOLOv1 has poor detection performance against small targets compared with R-CNN series. YOLOv2 was followed by YOLOv3. At that time, the birth of FPN further solved the problem of small targets. For small targets, the feature may disappear or be very small after multi-layer winder. It has both surface and semantic features, because the addition of fpn to YOLOv3 has achieved better performance. Besides the improvement of backbone, yolov4 and YOLOv5 also make innovations in data enhancement, such as the introduction of mosaic, mixup and other methods. Mosaic refers to combining four pictures into one for target detection. The advantage of this method is that rich background information helps detection, while mixup is combining two pictures together. In the neck part, the author also used panet's method, not only through two up-sampling concat, but also through two down-sampling on this basis. This allows features to be fused together more effectively. YOLOv6 is an object detection framework developed by Meituan's Visual Intelligence Department. Worth mentioning is that YOLOv6 also adopts the anchor-free approach and abandons the previous anchor-based method. YOLOv7 is developed by the original team of YOLOv4 and aims to make the YOLO algorithm faster and better.

In many working environments, the complexity and variability of the environment can pose a challenge for traditional helmet detection algorithms. These algorithms may lack the necessary generalization and robustness to be deployed on platforms with low computing power. The YOLO series network architecture is the most classic one-stage algorithm and the most widely used target detection network in the industrial field. YOLOv5 extends upon the advantages of YOLOv4 by optimizing its backbone and neck to yield better detection accuracy and faster inference speed [22]. YOLOv6 has better accuracy for small target detection, YOLOv7 is suitable for detecting close range targets, YOLOv5 has the smallest number of parameters and the fastest detection speed, and YOLOv5 has good generalization ability, so YOLOv5 is more suitable for embedded equipment site helmet detection. There are four versions in the official code of YOLOv5, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Among them, YOLOv5s has the smallest volume [23]. To address the shortcomings of existing helmet detection methods, this paper proposes a novel helmet detection algorithm based on YOLOv5s. This algorithm outperforms traditional detection methods in terms of accuracy, speed, and real-time performance, but it still has limitations in detecting small and dense targets. Therefore, this paper aims to improve the YOLOv5s detection algorithm to achieve better results in detecting small and dense targets. By enhancing the algorithm's capabilities, it can better adapt to complex and variable working environments, and provide more reliable helmet detection for safety purposes.

3 Approach

As the project of helmet detection is a long-term, large-scale, and critical initiative, the model needs to satisfy the daily monitoring frequency requirements during training, while also being deployable on embedded or mobile devices commonly used in construction sites, such as smart helmets, cell phones, and monitors. Hence, designing a model that can achieve both daily monitoring needs and real-time detection on embedded or mobile devices is a challenging task for helmet detection. To address this challenge, we present the modifications made in YOLOv5s to create the MG-YOLOv5s, a real-time helmet detector that can be used on embedded or mobile devices. The network architecture of our helmet detector is illustrated in Fig. 1 and comprises a backbone network, a neck, and a head. To aggregate features, we incorporate an SPP [24] and a PAN [25] in the neck.

3.1 Repstemblock

In this section, we draw inspiration from recent articles such as RepVGG [26], RepLKNet [27], and ElAN-NET, to design an efficient Stem module which we name RepStemBlock (as shown in Fig. 2). We found that multi-convolutional fusion with convolutional kernels of different sizes results in higher accuracy and speed for helmet-wearing detection compared to the original Focus module. Therefore, we implemented a convolution kernel with kernelsize = 13, 9, and 5 to downsample the input image during the first spatial downsampling, added fusion to the specific feature maps obtained, and then fused them with a shortcut of kernelsize = 3 after passing the activation function and increasing the number of channels. This ensures strong representation ability while reducing the model size and the number of parameters. The RepStem module essentially fuses the results of multiple convolution kernels of different sizes. At the shallow level, the results of the larger convolution kernel's convolution are fused, and the resulting features are concatenated. This module can focus on different dimensions of information, improve the receptive field of each point, and reduce the loss of original information.

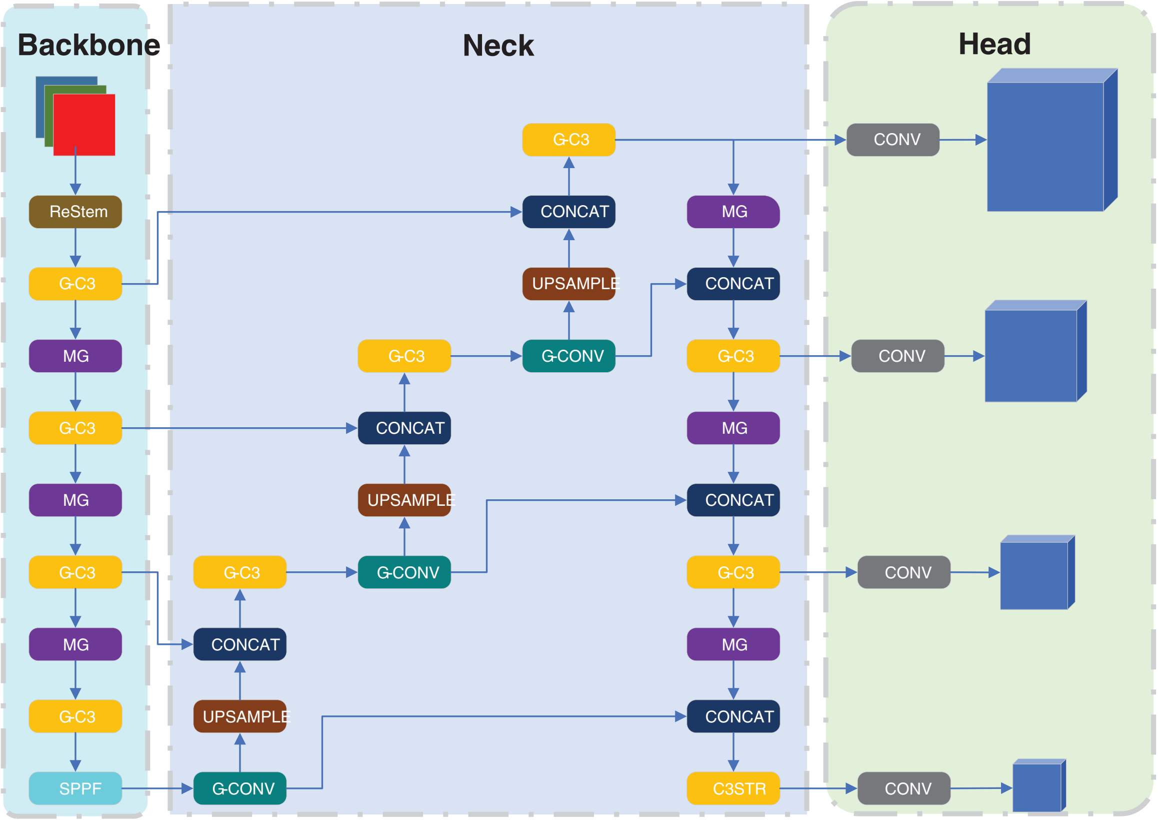


Figure 1: The architecture of the proposed MG-YOLOv5s

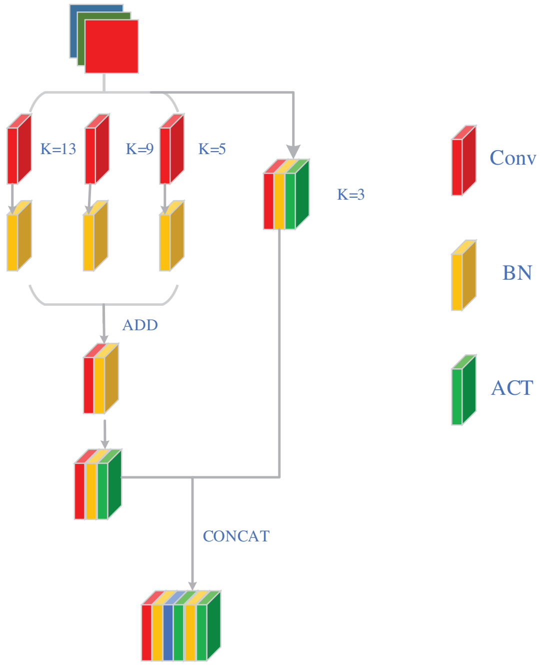


Figure 2: The structures of RepStem

3.2 MG

Convolutional neural networks have made significant strides in computer vision in recent years, but the cost of these advances has often increased computational complexity and the number of parameters required. To address these issues, we have introduced the Ghost module, which addresses the problems caused by traditional convolutions that can result in huge parameter sizes. After convolution, there are often many similar feature maps and redundancies that can lead to inefficiencies in neural networks. The Ghost module is designed to reduce these redundancies and improve the efficiency of the network.

In GhostNet [28], a Ghost module was introduced to address the problem of high computational complexity and parameter numbers in convolutional neural networks. The Ghost operation, as used in GhostNet, involves dividing the convolutional layer into two parts: a primary path and a cheap expansion path. The primary path performs regular convolution, while the expansion path employs a combination of pointwise convolution and linear operations to generate more feature maps, resulting in more features with fewer parameters.

The Ghost operation, as used in GhostNet, involves dividing the convolutional layer into two parts: a primary path and a cheap expansion path. The primary path performs regular convolution, while the expansion path employs a combination of pointwise convolution and linear operations to generate more feature maps. Given the input data x, with c channels and dimensions $h * w$, the operation to generate n feature maps using the Ghost module can be expressed as follows:

$$B = W * (A||E(A)) \tag{1}$$

$$C = BN(B) \tag{2}$$

$$D = \mathrm{ReLU}(C) \tag{3}$$

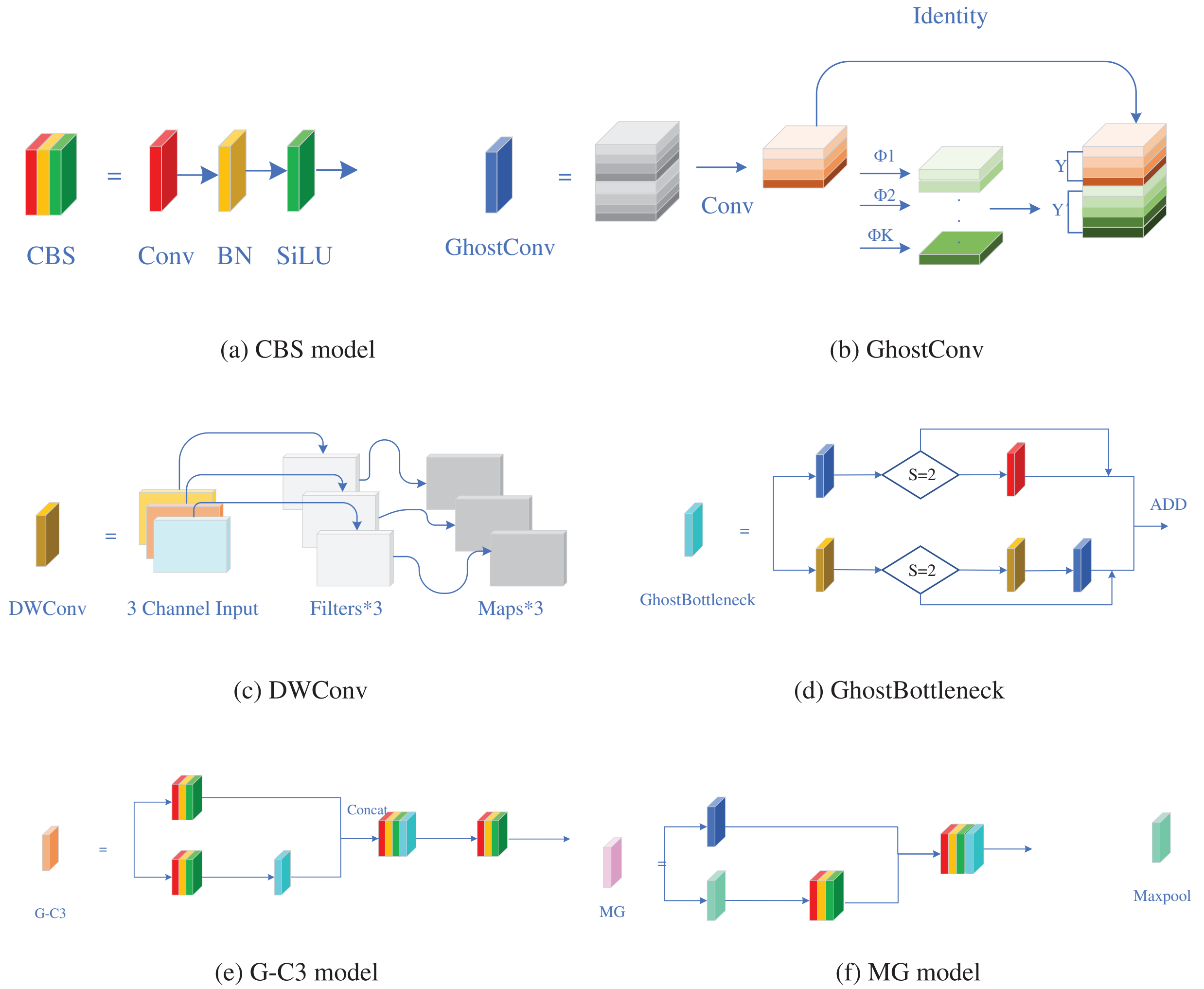
Here, W is the learnable convolutional filter, "||" represents channel concatenation, E is the cheap expansion operation, BN is the batch normalization operation, and ReLU is the activation function. The resulting feature maps have dimensions $C * D$, where C is the number of output channels and D is the spatial dimension of the output feature map. The kernel size of the convolutional filter is $k * k$.

Building on the Ghost module, we have designed an MG module that effectively prevents overfitting while enhancing the model's ability to capture global information, leading to improved performance. The MG module consists of a GhostConv layer with a Maxpool for Concatenation. While GhostConv effectively reduces the number of parameters and obtains more semantic information during downsampling, a significant amount of detailed feature information is lost. These details, however, may contain valuable features for small objects that could be overlooked during downsampling. To address this issue, we introduce a Maxpool residual linking approach, which not only prevents overfitting but also reduces dimensionality, removes redundant information, compresses features, simplifies network complexity, and reduces computation and memory consumption. Additionally, it alleviates the convolutional layer's over-sensitivity to location, leading to feature invariance. The G-C3 module is mainly composed of a GhostBottleneck layer and a residual structure. Fig. 3 Shows a range of structures that we have designed.

3.3 C3STR

In the final stage of the convolution process, the computational cost can be accurately calculated based on the number of convolution kernels (n) and channels (c). Due to the large value of n and c, this step requires a significant amount of computational power.

While the Vision Transformer [29] can only generate feature maps of a single resolution, the Swin Transformer [30] (as shown in Fig. 4) is designed to merge deeper image patches and perform self-attentive computation within a local window of each layer, resulting in a linear computational complexity concerning the input image size.



(a) CBS model

(b) GhostConv

(c) DWConv

(d) GhostBottleneck

(e) G-C3 model

(f) MG model

Figure 3: The structures of the MG model

The backbone network of YOLOv5 target detection is comprised of multiple convolutional layers. As the number of layers increases, the features extracted from the input image become more abstract, and the spatial resolution decreases. This can lead to the loss of important spatial and feature information. To enhance the detection of large hard hats, we added a C3STR (as shown in Fig. 5) module before the P5 module. While most helmet detectors focus on detecting small helmets, we aimed to improve the detection of larger helmets. We replaced the original C3 module with the C3STR module, which uses a Swin Transformer Block and a residual structure. The Swin Transformer Block consists of a moving window-based MSA module and a two-tier MLP with GELU nonlinearity. Each MSA module and MLP is preceded by a LayerNorm layer, and a residual connection is applied after each module. This approach increases the retention of useful information and suppresses irrelevant information, leading to better detection of large helmets by YOLOv5s. In mathematical formulas, this can be expressed as

$$\widehat{z}^{l} = W - MSA\left(LN\left(z^{l-1}\right)\right) + z^{l-1} \tag{4}$$

$$\widehat{z}^{l} = MLP\left(LN\left(\hat{z}^{l}\right)\right) + \hat{z}^{l} \tag{5}$$

$$\widehat{z}^{l+1} = SW - MSA\left(LN\left(z^{l}\right)\right) + z^{l} \tag{6}$$

$$z^{l+1} = MLP\left(LN\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1} \tag{7}$$

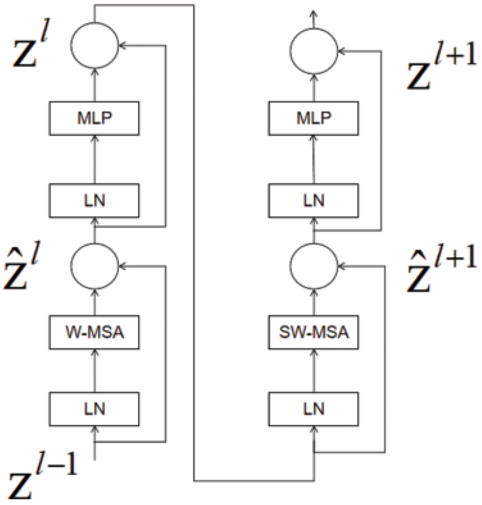


Figure 4: The structures of STR

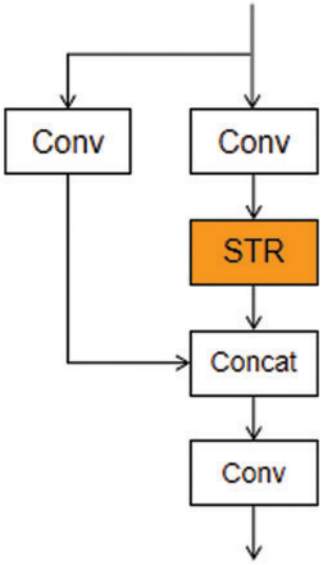


Figure 5: The structures of C3STR

z^1 and $\hat{\mathrm{z}}^l$ denote the output characteristics of the (S) WMSA module and the MLP module of block, respectively.

3.4 P6

To enhance the performance of the network and ensure that it can learn the global characteristics of hard hats in fine detail, a small target detector named P6 is added to the original three-channel detection. By integrating semantic information from different dimensions, the network can perform more refined processing of small targets, improving the accuracy of identification and localization. This approach reduces the likelihood of missed or false detections of small hard hats.

The expanded four-channel receptive field structure, as illustrated in Fig. 6, enables the network to gather information from a larger spatial range, enhancing its ability to capture small targets. By considering features from multiple dimensions, including spatial, channel, and semantic information, the network gains a more comprehensive understanding of the object. As a result, the network can better distinguish small hard hats from their surroundings and accurately locate them in the image. This finer-grained processing allows for improved performance in real-world scenarios, where detecting small hard hats can be crucial for ensuring worker safety.

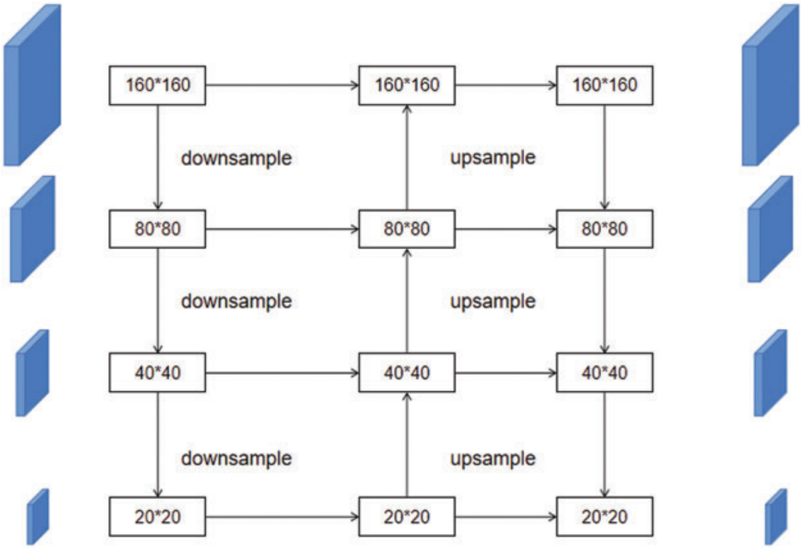


Figure 6: The structures of four-channel

3.5 Enhancement

During this experiment, it was discovered that certain data enhancement techniques were not effective in improving helmet detection. Specifically, top-down flipping and mosaic were found to be unsuitable. The up-and-down flipping technique was removed from the data enhancement pipeline, resulting in improved performance.

Additionally, it was observed that mosaic enhancement was not effective when working with small images, as it led to a decrease in performance. However, it was found to perform well when applied to larger images.

On the other hand, random cropping was found to be a useful data enhancement technique for improving helmet detection. This technique involves randomly selecting and cropping sections of the image, which provides the model with a wider range of training samples and helps improve its ability to generalize to new, unseen data.

3.6 K-Means++

The COCO dataset, which is a generic target detection dataset, was used to obtain the initial a priori anchor parameters of YOLOv5 by applying k-means clustering. However, as the COCO dataset contains 80 categories of target types, and this paper focuses only on detecting two categories of helmets in construction site scenarios, the size of the prior bounding boxes needs to be redesigned to better suit this specific task. To achieve this, we utilized the *k*-means++ algorithm to multidimensionally cluster the labeled target boxes within the helmet-wearing dataset (as shown in Table 1), resulting in the identification of different numbers and sizes of prior bounding boxes. This approach of redesigning the anchor boxes specifically for the task of helmet detection led to improved accuracy by enabling the model to more accurately match the a priori anchor frames with the actual targets (as shown in Table 2 and Fig. 7). By achieving a better match between the prior bounding boxes and the actual targets, the model can better localize and detect the helmets in the given images. This approach is important as it allows for the customization of the anchor boxes for a specific task, which can lead to improved performance over using generic anchor frames based solely on a priori knowledge.

Table 1: Peior anchor box sacles

Feature map scale	Anchor1	Anchor2	Anchor3
P3	(8, 10)	(10, 12)	(13, 16)
P4	(18, 21)	(24, 28)	(34, 39)
P5	(45, 54)	(64, 76)	(91, 108)
P6	(138, 162)	(199, 228)	(353, 407)

Table 2: Effect evaluation of different anchors on the test set

Method	P	R	mAP/%
Baseline	0.921	0.900	0.942
K-means++	0.927	0.904	0.948

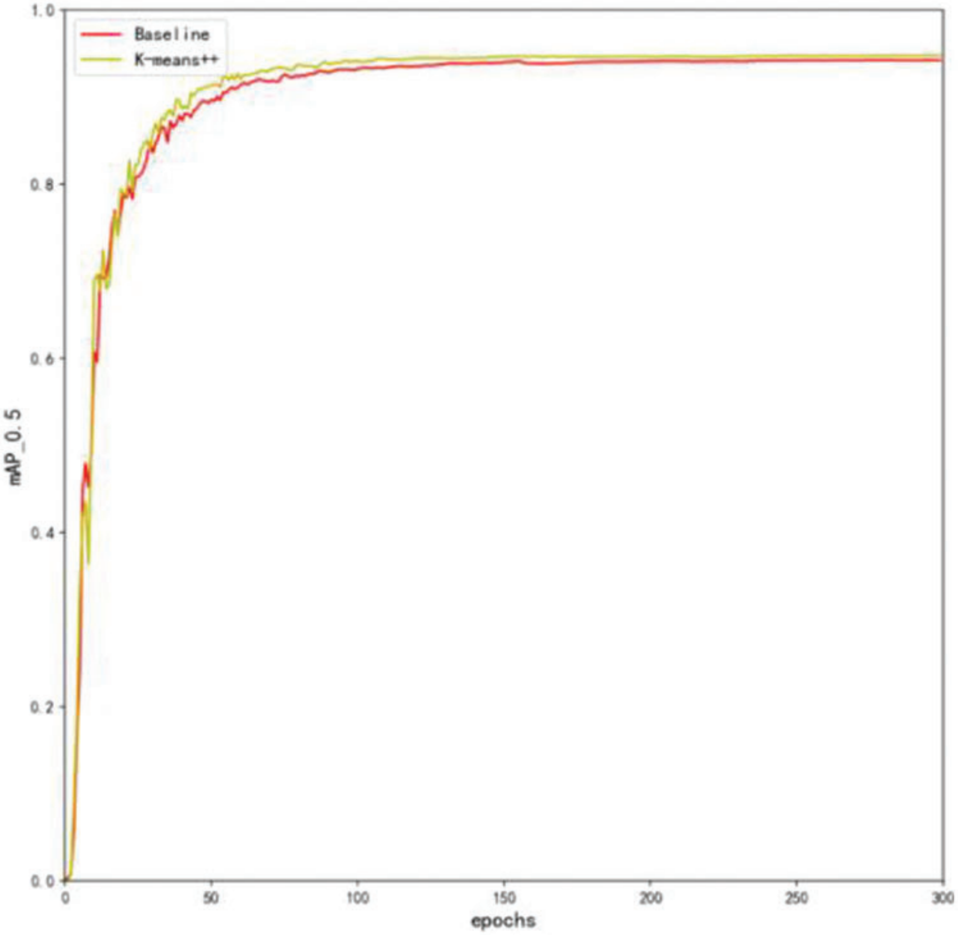


Figure 7: Result of contrast

4 Experiment

4.1 Experimental Environment and Data Set

In this experiment, the PyTorch framework was utilized to construct and train the network. The training and testing of the results were performed within this framework. A tower GPU professional workstation was used for neural network training, equipped with an Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz processor and 16 GB of RAM. The configuration of the machine used for testing was an Intel Xeon(R) Silver 4210R @ 2.40 GHz processor, 128 GB of RAM, and an NVIDIA Tesla M40 24 GB graphics card (as shown in Table 3).

The dataset employed in this experiment was the SafetyHelmetWearing-Dataset (SHWD) [31], which provides data sets for safety helmet-wearing and head detection. It comprises 7,581 images, of which 9,044 images depict objects wearing hard hats (positive), and 111,514 images depict everyday head objects (not wearing helmets or negative). The dataset includes various construction scenarios, which enable a more comprehensive reflection of actual construction scenarios. Finally, the dataset was partitioned into a training set and a validation set in a ratio of 9:1 (as shown in Table 4).

Table 3: Experiment operating environment (List of equipment and versions required for the experiment)

Category	Entry	Version
Hardware configuration	GPU	NVIDIA Tesla M40 24 GB
	CPU	Intel(R) Core (TM) i7-10700
Software configuration	PyTorch	12.1
	CUDA	11.3

Table 4: Dataset category allocation

Target category	Training number	Test number	Total
Human safety helmet-wearing objects (positive)	8140	904	9044
Normal head objects (not wearing or negative)	100362	11152	111514

4.2 Evaluation Indicators

In this experiment, several evaluation metrics were used to measure the performance of the helmet detection system. These metrics include Precision, Recall, mAP, speed index FPS, and time. TP refers to the test image and predicted image as hat images; FP refers to the test as person images; FN refers to the test as hat images, and prediction is person image:

- Mean Average Precision (mAP0.5): The mAP is a widely used metric in object detection that calculates the average precision of an algorithm over a range of IoU (Intersection over Union) thresholds. In this experiment, mAP at an IoU threshold of 0.5 was used to evaluate the performance of the helmet detection system:

$$AP = \int_0^1 (Precision) d(Recall) \tag{8}$$

$$mAP = \frac{1}{n} \sum\nolimits_{i=1}^{n} AP_i \tag{9}$$

- Recall (R): Recall is the proportion of actual positive samples that are correctly identified by the system. In the context of helmet detection, recall measures the ability of the system to detect all instances of helmets in the image:

$$Recall = \frac{TP}{TP + FN'} \tag{10}$$

- Precision (P): Precision is the proportion of positive predictions made by the system that is correct. In the context of helmet detection, precision measures the accuracy of the system in detecting helmets:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

- Parameters (Para/m): Parameters refer to the number of parameters in the model and the time it takes to process each image. In this experiment, the number of parameters in the model has reported as well as the processing speed in terms of parameters per second.

- FPS: The FPS (frames per second) represents the number of images that can be processed by an object detection method per second, and it tests the real-time performance of the detection method. The higher the FPS value, the faster the detection speed, f_n indicates the number of detected images, and T represents the detection time used.

$$FPS = \frac{f_n}{T} \tag{12}$$

4.3 Comparison Test

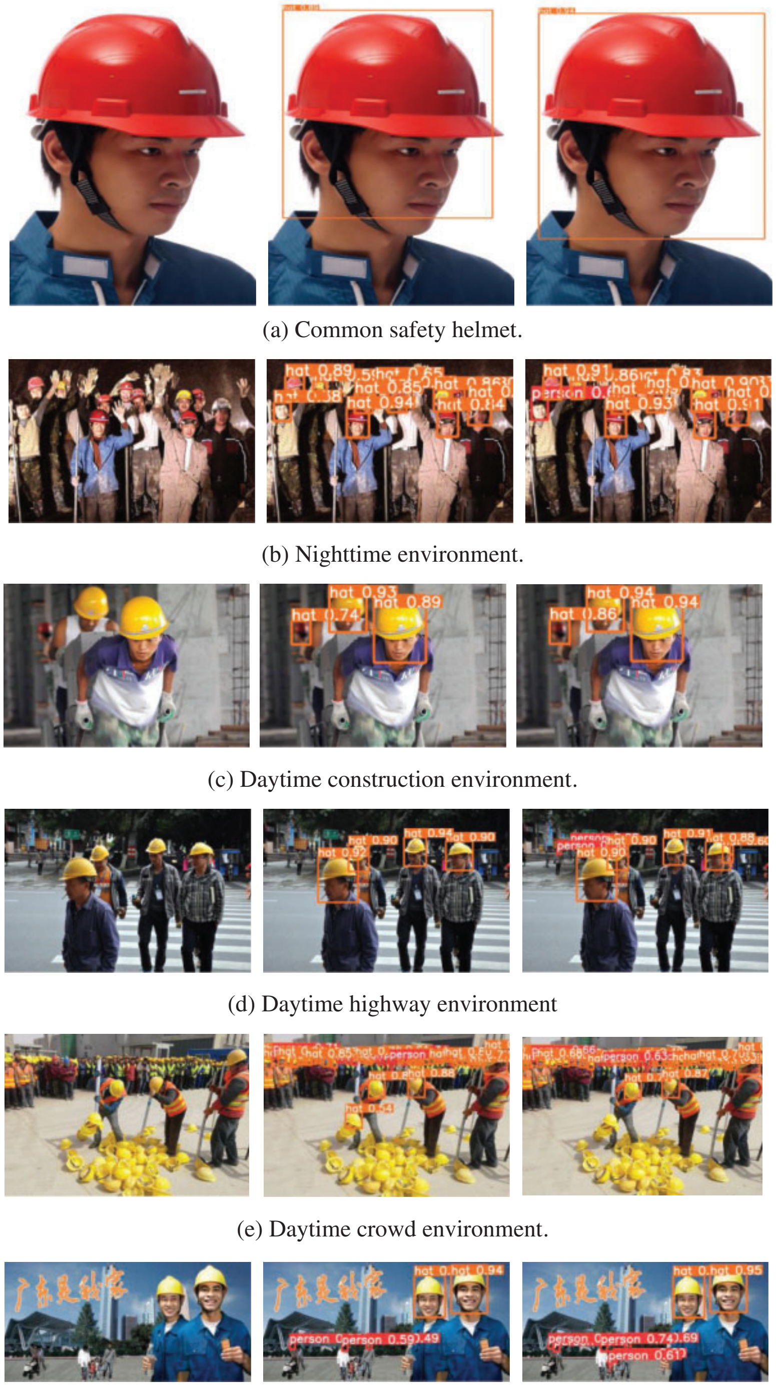
To evaluate the effectiveness and reliability of the proposed algorithm, we conducted a comparative analysis with several popular target detection algorithms, such as Faster R-CNN, SSD, YOLOv3, YOLOv3 + SPP, YOLOv5s, and YOLOv5s + Ghost algorithm, and the results are summarized in Table 5. Additionally, we provide some detection results of the MG-YOLOv5s helmet detection algorithm for the test sample set, as shown in Fig. 8.

Table 5: Comparison results of SHWD evaluation

Detection model	Para/m	AP50/% (Hat)	AP50/% (Person)	mAP/%	FPS (GPU)	Time/ms
Faster R-CNN	186	80.8	42.2	61.5	0.7	1428.6
SSD	23.75	78.8	68.2	73.5	38	26.3
YOLOv3	61.9	89.12	80.7	84.9	8	125
YOLOv3 + SPP	63.0	90.5	86.3	88.41	14	71.4
YOLOv5s [32]	7.26	93.3	91.7	92.7	79	12.7
YOLOv5s + Ghost	4.17	90.7	89.5	90.1	65	15.4
YOLOv5s-hat	7.31	95.9	95.6	95.7	83	12.0
MG-YOLOv5s	4.06	95.6	95.1	95.4	67	14.9

Among all models (as shown in Table 5), the mAP@0.5 score of Faster R-CNN were the lowest, with a large parameters and computational effort, resulting in only 0.7 FPS, making it unsuitable for helmet real-time wear detection. The one-stage detection model SSD had an mAP@0.5 value of 73.5% and a parameters of 23.75, which did not meet the detection requirements in terms of model accuracy and complexity. In the YOLO model series, The YOLOv3 achieved 84.9% mAP@0.5, but the parameters were 61.9, and there were only 8 pictures per second. YOLOv5s-Ghost had an mAP@0.5 of only 90.1%, and the accuracy was too low. Neither of them was not conducive to mobile deployment. The proposed MG-YOLOv5s model had the highest mAP@0.5 and the lightest structure among all models, exceeding Faster R-CNN, SSD, YOLOv3, YOLOv3-SPP, YOLOv5s and YOLOv5s-Ghost by 33.9%, 21.9%, 10.5%, 6.99%, 2.7% and 5.3%, respectively. The parameters were less than the baseline model, and FPS reached 67 frames per second to meet real-time detection of apple leaf diseases in real scenarios.

The yellow rectangular box in Fig. 8 represents the helmet-wearing situation detected by the model, while the red box represents the head without a helmet. The designed model can accurately detect whether a worker is wearing a helmet in different environments and crowds, as shown in Fig. 8. The left side of Fig. 8 shows the original image, the middle shows the detection effect of YOLOv5s, and the right side shows the detection effect of the MG-YOLOv5s designed in this paper.



(a) Common safety helmet.

(b) Nighttime environment.

(c) Daytime construction environment.

(d) Daytime highway environment

(e) Daytime crowd environment.

(f) Complex population.

Figure 8: Result of the test

In Figs. 8a–8f, we present several scenarios to illustrate the effectiveness of the proposed algorithm. In a dense crowd construction scene in a general environment, as shown in Fig. 8a, the original

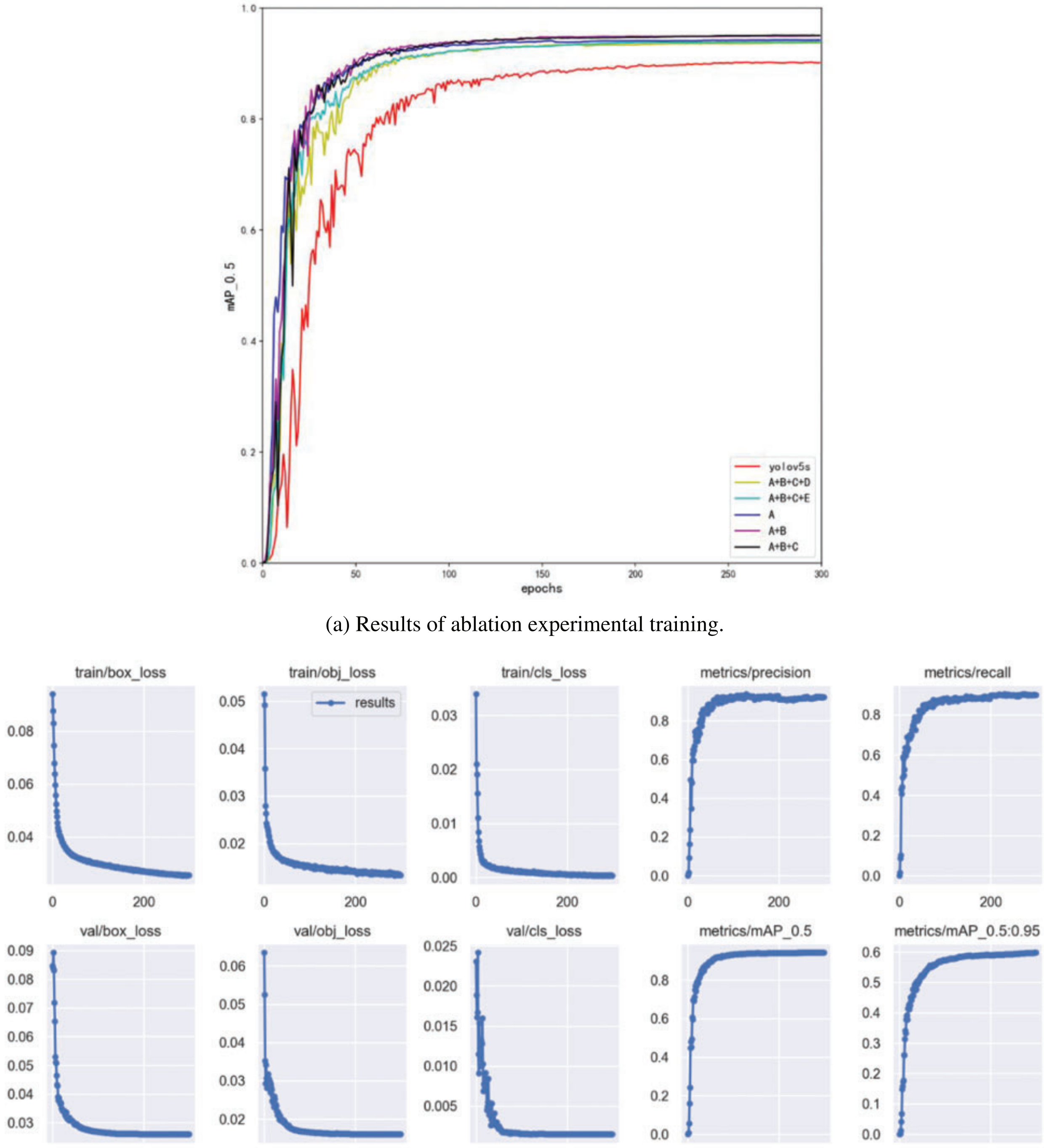
YOLOv5s mAP@0.5 of 0.89, 0.5 lower than MG-YOLOv5s. In Fig. 8b, in the nighttime environment, the original YOLOv5s wrongly detected the worker on the far left without a helmet as wearing a helmet, but the proposed algorithm correctly detected all workers wearing helmets. Fig. 8c depicts a construction site under a strong light environment, where YOLOv5s failed to detect the construction workers who did not wear helmets in the middle and far away, while the proposed algorithm correctly detected them. Similarly, in Fig. 8d, we present a complex environment in the daytime highway, where YOLOv5s had more missed detection and some false detection, whereas the proposed algorithm detected all workers wearing helmets correctly, and correctly detected pedestrians in the distance who were not wearing hard hats. Fig. 8e present the daytime dense crowd environment, where YOLOv5s had error detection, while the proposed algorithm successfully detected all workers wearing helmets. Finally, in Fig. 8f, we present a complex population environment, where we observed that YOLOv5s missed to detect pedestrian on the left half, while the proposed algorithm correctly detected all workers wearing helmets and pedestrian.

4.4 Ablation Experiment

To evaluate the effectiveness of different components of the proposed MG-YOLOv5s model, ablative experiments were conducted and the results are presented in Fig. 9a and Table 6. The following sections summarize the findings of each experiment:

- P6: To improve the detection of small targets, a four-channel detection method was introduced, and a new parameter P6 was added. The results show that the inclusion of P6 with a size of 0.14 m make a 1.4% increase in mAP, at the same time, the FPS of the model reaches 78, and the inference speed was 12.8 ms, indicating that it is an effective component.
- C3STR: The C3STR module was added to enhance the detection of larger helmets. The module introduces 0.13 m parameters and results in a 0.6% increase in mAP. The experiment shows that the C3STR module is an effective component for detecting larger targets.
- Repstem: A new module called Repstem was designed to replace the original Focus module. This replacement led to a reduction in parameters while also improving the performance of the network. The experiment results indicate that Repstem is a successful module in terms of both reducing parameters and enhancing performance.
- GhostNet: GhostConv was introduced in this study as an alternative to regular convolution to extract features and reduce the complexity of the model. The experiment results show that after incorporating GhostConv, the number of parameters was reduced by 63%, with a 1.8% decrease in mAP compared to the original YOLOv5s, the FPS of the model reached 62, and the inference speed was 16.1 ms.
- MG: To minimize the impact of using GhostConv on model performance, the MG module was designed. This module further reduces the number of parameters while maintaining performance. The experiments show that after using the MG module, the number of parameters is only 56% of the original, and the mAP is only 0.3% lower and 1.5% higher than that without using the module. Meanwhile, the FPS of the model reaches 67, i.e., the model can process 67 frames per second, and it takes only 14.9 ms to detect each helmet image, which can realize the real-time monitoring of helmet. The training results are shown in Fig. 9b.

In summary, the ablative experiments demonstrate the effectiveness of each component in improving the performance of the proposed MG-YOLOv5s model. The results suggest that the model is highly optimized, lightweight, and effective in detecting helmets in different environments.



(a) Results of ablation experimental training.

(b) A+B+C+E Training results.

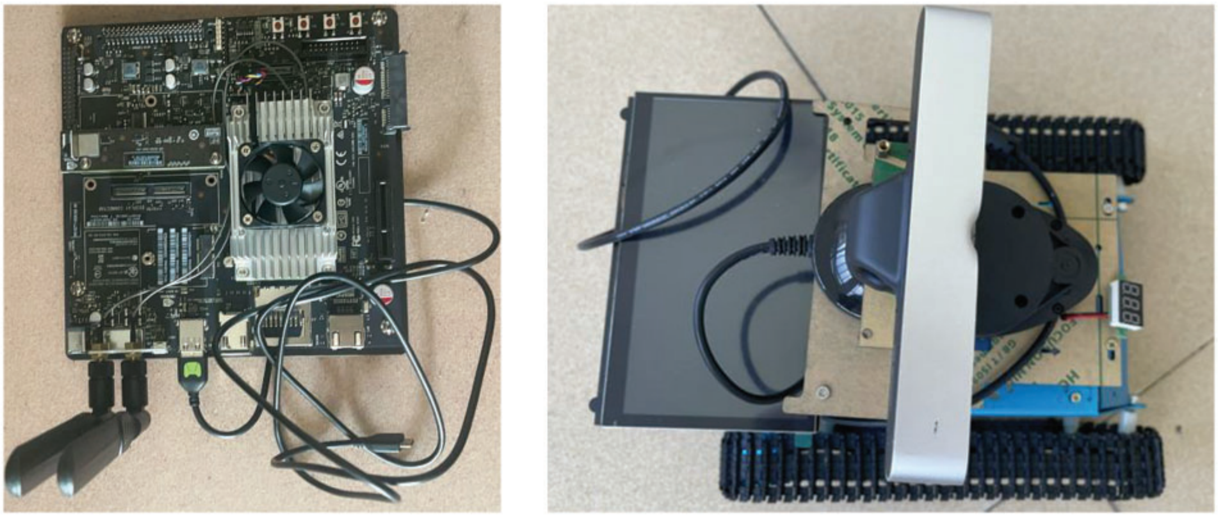
Figure 9: Result of MG

Table 6: Dataset category allocation

Basic model	A	B	C	D	E	Hat	Person	mAP	P	R	Pa/m	FPS	Time/ms
√						93.3	91.7	92.7	92.4	87.5	7.03	79	12.7
√	√					94.4	93.7	94.1	93.6	87.7	7.17	78	12.8
√	√	√				95.1	94.3	94.7	94.1	89.7	7.30	80	12.5
√	√	√	√			95.9	95.6	95.7	93.1	91.4	7.19	83	12.0
√	√	√	√	√		93.7	94.1	93.9	93.5	88.6	4.54	62	16.1
√	√	√	√		√	95.6	95.1	95.4	93.5	90.6	4.06	67	14.9

4.5 Mobile Equipment

To prove that the model can be easily deployed on embedded devices, we used the Jetson TX2 (the default mode for a is CPU device) developer component, which has many external interfaces and resources to make the hardware features of the core module fully useful, as shown in Fig. 10a. It mainly includes: (1) Jetson TX2 core module (2) WIFI/Bluetooth support (3) USB OTG interface (4) Gigabit Ethernet support. Connected to the cart with monitor (as shown in Fig. 10b) via the USB interface, it can be used to realize real-time helmet detection. The hardware platform Jetson TX2 used in this system is equipped with Ubuntu 16.04 system, which is an operating system based on Debian GNU/Linux, supporting x86/amd64/arm64 and other architectures.



(a) TX2

(b) Cart with monitor

Figure 10: Equipment

TensorRT is a GPU Inference Engine (GIE) introduced by Nvidia. Unlike common deep learning frameworks, TensorRT provides only forward propagation, or reasoning, capabilities, not training capabilities. TensorRT can decompose our trained models and then integrate them. The integrated models have a high degree of aggregation. For example, after the convolution layer and the activation layer are fused, the computational speed can be improved.

First, we configured the environment on Jetson TX2 and imported the MG-YOLOv5s model and the best training weights. In the second step, we modify the code so that we can successfully call the camera of the car connected via Jetson TX2. The third step is to use TensorRt acceleration, we use tensorrtx [33], compared to ONNX2TensorRT, tensorrtx is to manually construct the model structure, and then manually move the weight information over, very flexible, and manually construct the model structure is the most reliable, the highest degree of model reduction, and the lowest loss of accuracy. After converting our best training weight (best.pt) model into best.wts model through tensorrtx, the generated best. In the last step, we connected the USB interface of Jetson TX2 to the car and called the camera of the car (i.e., called the USB camera) to carry out the helmet wearing detection on the site. The results of this experiment on embedded devices are shown in Table 7, and the FPS of our MG on TX2 is 26, conforming to the general embedded devices in the industrial application of FPS on request.

Table 7: Dataset category allocation

Detection model	Para/m	mAP/%	FPS(GPU)	FPS (TX2)
YOLOv5s	7.26	92.7	79	13
YOLOv5s-hat	7.31	95.7	83	11
MG-YOLOv5s	4.06	95.4	67	26

5 Conclusion

The conventional method of detecting helmet-wearing is resource-intensive, time-consuming, and requires additional personnel to confirm the results. This study aims to propose a lightweight, high-precision, and mobile-terminal-compatible helmet-wearing detection model using deep learning-based target detection methods. While there are several methods available, this paper focuses on building a lightweight model based on the YOLO architecture due to time cost, computational complexity, and model robustness limitations.

The proposed model, MG-YOLOv5s, is an improvement on the original YOLOv5 network structure, with several modifications made to achieve a lightweight model that can accurately detect helmet-wearing on construction sites. The improved model reduces the number of parameters and improves the inference speed while surpassing the accuracy and mAP performance of the original YOLOv5s model. Each improved part of the model has a positive impact on the results, and the research objective is effectively achieved.

Although MG-YOLOv5s's mAP@0.5 is 0.3% lower than YOLOv5s-hat 's, the significant reduction in parameters and model size enables it to be used in real-time detection under construction sites, contributing to the safety of workers. This study establishes the groundwork for a smart site, where advanced technologies like deep learning-based helmet-wearing detection models can be used to improve worker safety and enhance productivity.

Acknowledgement: The authors would like to thank the authors of SHWD. They provided the open source dataset.

Funding Statement: This work was supported by the National Natural Science Foundation of China under Grant No. 41901296, the Fujian Provincial Key Laboratory of Data Intensive Computing and Key Laboratory of Intelligent Computing and Information Processing under Grant No. BD201801, and the Wuhan Science and Technology Bureau Knowledge Innovation Dawning Plan Project: Detection and Optimization Method of GNSS Hybrid Attacks for Connected and Autonomous Vehicles under Grant No. 2022010801020270.

Author Contributions: Conceptualization, Z.X.; methodology, Z.X. and W.L.; investigation, T.Y. and S.L.; validation, W.L. and Z.Y. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: In this article, the data set was created based on SafetyHelmet Wearing-Dataset (SHWD).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Agarwal, J. Terrail, and F. Jurie, "Recent advances in object detection in the age of deep convolutional neural networks," 2018. doi: 10.48550/arXiv.1809.03193.

[2] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning," *J. Latex Class Files*, vol. 14, no. 8, pp. 1–21, 2017.

[3] X. H. Liu and X. Ye, "The application of skin color detection and Hu moment in helmet recognition," *J. East China Univ. Sci. Technol. Nat. Sci. Ed.*, vol. 40, no. 3, pp. 365–370, 2014.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR'05)*, San Diego, CA, USA, 2005, vol. 1, pp. 886–893.

[5] Q. R. Li, "Based on the human body to identify the safety helmet video detection system research and implementation," *J. Univ. Electron. Sci. Technol. China*, vol. 2017, no. 2, pp. 1–99, 2017.

[6] A. H. M. Rubaiyat *et al.*, "Automatic detection of helmet uses for construction safety," in *Proc. 2016 IEEE/WIC/ACM Int. Conf. Web Intell. Workshops (WIW)*, Nebraska, NE, USA, 2016, pp. 135–142.

[7] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," *J. IET Intell. Transp. Syst.*, vol. 6, no. 3, pp. 259–269, 2012. doi: 10.1049/iet-its.2011.0138.

[8] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Proc. Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023. doi: 10.1007/s11042-022-13644-y.

[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, and U. C. Berkeley, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 580–587.

[10] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 2015, pp. 1440–1448.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks." in *Advances in Neural Information Processing Systems*. Montreal, Canada, 2015. vol. 28.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 779–788.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 7263–7271.

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[15] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[16] G. Jocher, "YOLOv5," Accessed: Apr. 22, 2023, 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[17] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Comput. Vis.–ECCV 2016: 14th Eur. Conf.*, Amsterdam, The Netherlands, 2016, pp. 21–37.

[18] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," arXiv preprint arXiv:1701.06659, 2017.

[19] Q. Zhao *et al.*, "M2Det: A single-shot object detector based on multilevel feature pyramid network," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, 2019, vol. 33. no. 1. 9259–9266.

[20] S. Zhang, L. Wen, Z. Lei, and S. Z. Li, "RefineDet++: Single-shot refinement neural network for object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 674–687, 2020. doi: 10.1109/TCSVT.2020.2986402.

[21] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Venice: IEEE*, Venice, Italy, 2017, pp. 2980–2988.

[22] Q. B. Wu, Z. Wang, H. F. Fang, J. J. Chen, and X. F. Wan, "A lightweight electronic water pump shell defect detection method based on improved YOLOv5s," *Comput. Syst. Sci. Eng.*, vol. 46, no. 1, pp. 961–979, 2023. doi: 10.32604/csse.2023.036239.

[23] Y. M. Li, J. Zhang, Y. Hu, Y. N. Zhao, and Y. Cao, "Real-time safety helmet-wearing detection based on improved YOLOv5," *Comput. Syst. Sci. Eng.*, vol. 43, no. 3, pp. 1219–1230, 2022.

[24] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, 2018. doi: 10.1016/j.neunet.2017.12.012.

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 8759–8768.

[26] K. Han *et al.*, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 1580–1589.

[27] A. Dosovitskiy *et al.*, "An image is worth 16 × 16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2010.

[28] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, Canada, 2021, pp. 10012–10022.

[29] X. Chu, L. Li, and B. Zhang, "Make RepVGG greater again: A quantization-aware approach," arXiv preprint arXiv:2212.01593, 2022.

[30] X. Ding, X. Zhang, J. Han, and J. Ding, "Scaling up your kernels to 31 × 31: Revisiting large kernel design in cnns," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 11963–11975.

[31] Njvisionpower, "Safetyhelmetwearing-dataset (SHWD)," Accessed: Apr. 22, 2023, 2019. [Online]. Available: https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset/

[32] J. Zhang *et al.*, "DWCA-YOLOv5: An improve single shot detector for safety helmet detection," *Proc. J. Sens.*, vol. 2021, pp. 1–12, 2021. doi: 10.1155/2021/9985747.

[33] X. Y. Wang, "tensorrtx," Accessed: Apr. 22, 2023, 2021. [Online]. Available: https://github.com/wang-xinyu/tensorrtx