

Article

LG-YOLOv8: A Lightweight Safety Helmet Detection Algorithm Combined with Feature Enhancement

Zhipeng Fan ^{1,2,*}, Yayun Wu ¹, Wei Liu ^{1,2}, Ming Chen ^{1,2} and Zeguo Qiu ^{1,2}

¹ School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China; wuyayun@s.hrbcu.edu.cn (Y.W.); weiwei@hrbcu.edu.cn (W.L.); chenming@hrbcu.edu.cn (M.C.); qiuze@hrbcu.edu.cn (Z.Q.)

² Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin 150028, China

* Correspondence: fanzhipeng@hrbcu.edu.cn

Abstract: In the realm of construction site monitoring, ensuring the proper use of safety helmets is crucial. Addressing the issues of high parameter values and sluggish detection speed in current safety helmet detection algorithms, a feature-enhanced lightweight algorithm, LG-YOLOv8, was introduced. Firstly, we introduce C2f-GhostDynamicConv as a powerful tool. This module enhances feature extraction to represent safety helmet wearing features, aiming to improve the efficiency of computing resource utilization. Secondly, the Bi-directional Feature Pyramid (BiFPN) was employed to further enrich the feature information, integrating feature maps from various levels to achieve more comprehensive semantic information. Finally, to enhance the training speed of the model and achieve a more lightweight outcome, we introduce a novel lightweight asymmetric detection head (LADH-Head) to optimize the original YOLOv8-n's detection head. Evaluations on the SWHD dataset confirm the effectiveness of the LG-YOLOv8 algorithm. Compared to the original YOLOv8-n algorithm, our approach achieves a mean Average Precision (mAP) of 94.1%, a 59.8% reduction in parameters, a 54.3% decrease in FLOPs, a 44.2% increase in FPS, and a 2.7 MB compression of the model size. Therefore, LG-YOLOv8 has high accuracy and fast detection speed for safety helmet detection, which realizes real-time accurate detection of safety helmets and an ideal lightweight effect.

Keywords: safety helmet testing; lightweight network; C2f-GhostDynamicConv; BiFPN; LADH-Head



Citation: Fan, Z.; Wu, Y.; Liu, W.; Chen, M.; Qiu, Z. LG-YOLOv8: A Lightweight Safety Helmet Detection Algorithm Combined with Feature Enhancement. *Appl. Sci.* **2024**, *14*, 10141. <https://doi.org/10.3390/app142210141>

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Pedro Moreno-Bernal

Received: 30 September 2024

Revised: 26 October 2024

Accepted: 27 October 2024

Published: 6 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As industrialization advances, the emphasis on safety management in the construction industry has grown. Safety helmets, which are essential personal protective equipment, play a vital role on construction sites, substantially reducing the potential for head injuries. However, the consistent and conscientious use of these helmets by workers is not always guaranteed, highlighting the need for real-time monitoring to ensure their utilization and, by extension, secure work environments. Unfortunately, such monitoring is typically resource-intensive and laborious, and it also poses challenges in terms of cost and practicality when it comes to deploying on edge devices for an efficient outcome.

Currently, certain accomplishments have been attained in the domain of object detection. Particularly, the approaches based on deep learning have made breakthrough advancements in the performance detection of numerous tasks. Presently, the mainstream methods are mainly categorized into two types: two-stage algorithms and single-stage algorithms. The former mainly relies on the two-stage object detection algorithm of candidate regions to achieve high-precision object detection. Representative algorithms in this category include R-CNN [1], Faster R-CNN [2], etc. The second category uses a single network to predict the location and category of the object directly from the input image, avoiding the complex process of generating candidate regions and achieving faster detection speed, and the main representative networks in this category include SSD [3] and YOLO [4–9].

These two types of methods have certain advantages for target detection tasks, but considering the specific task of safety helmet detection and in the real scene of a construction site for objects of different sizes and the influence of other uncontrollable factors, especially for some places with limited resources, it is a general trend to design high precision lightweight models. YOLOv8-n [10] demonstrates enhanced accuracy and speed over its predecessors, yet the detection accuracy for small objects like helmets remains suboptimal, and the substantial model size complicates deployment on edge devices. Based on this, on the basis of the YOLOv8-n algorithm, this paper optimizes the model to a certain extent and introduces the novel LG-YOLOv8 safety helmet detection algorithm, which is a high-precision lightweight model with fast detection speed and strong detection effect.

The main contributions of this paper are as follows:

(1) In the backbone of the YOLOv8-n model, a novel structure named C2f-Ghost-DynamicConv is introduced. This structure amalgamates the strengths of the Ghost module and dynamic convolution, substantially decreasing computational load and parameter count through a distinctive feature generation process. Concurrently, it employs dynamic convolution to adaptively modify the convolution kernel, thereby enhancing the model's feature representation capabilities. This innovation notably improves the detection performance in complex construction settings, such as those involving distant or partially obscured helmets, leading to heightened accuracy and efficiency.

(2) Optimize the traditional unidirectional feature fusion method into a bidirectional feature pyramid structure, enhancing feature fusion and multi-scale feature extraction capabilities through the bidirectional feature pyramid network. By leveraging contextual information and combining top-down and bottom-up feature transfer, it effectively extracts multi-scale features, addressing the issues of background interference and leakage in complex scenes and different sizes of helmets. This improves the model's recognition accuracy for helmets.

(3) In the construction of the detection head, the LADH-Head has been implemented as an upgrade. It utilizes local dense connections and an adaptive module to significantly improve the detailed features of small targets such as safety helmets, thus minimizing information loss and substantially reducing the occurrence of missed and false detections of safety helmets. Furthermore, the application of the asymmetric convolution operation serves to diminish the parameter count and computational demand. This not only lightens the hardware resource demands of the model but also ensures exceptional detection performance.

(4) The proposed lightweight detection algorithm network was evaluated on the SHWD dataset, and the mAP and FPS reached 94.1% and 153.3, respectively, and the GFLOPs also reached 3.7. Compared with other methods, the accuracy of the model is improved, the execution speed is also significantly improved, and the computational cost is significantly reduced.

2. Related Work

The application of deep learning in the field of helmet detection has gradually become a research hotspot, and some remarkable research results have been obtained. Lian et al. [11] introduced a novel network, HR-YOLO, featuring a residual network architecture and a PANMS method to enhance safety helmet detection precision. Han et al. [12] designed a lightweight end-to-end convolutional neural network, GhostNet, as the backbone, incorporating multi-scale segmentation and feature fusion during feature processing. Using this design, the accuracy reached 93.5%, and the FPS also achieved good results. Chen et al. [13] utilized the K-means++ algorithm to categorize helmet sizes in images and subsequently applied an enhanced Faster-RCNN algorithm for helmet usage detection. These studies predominantly concentrate on elevating detection accuracy and efficiency, yet this often escalates model complexity, computation demands, and parameter counts, overlooking the constrained computational capabilities of embedded systems at construction sites. The advent of edge computing has seen an increasing number of surveillance

systems implementing recognition algorithms on edge-embedded devices, catalyzing the emergence of lightweight target detection algorithms as a burgeoning research focus. Chen et al. [14] employed PP-LCNet as the backbone of YOLOv4 and incorporated depthwise separable convolution to decrease model parameters, thereby enhancing both accuracy and detection speed.

Based on the above research, we conclude that the current focus is on improving the accuracy and efficiency of object detection, but this leads to increasing model complexity, increasing the amount of calculation and parameters, so lightweight object detection algorithms have gradually become a new research hotspot, and many lightweight networks have emerged. Examples are ShuffleNetV2 [15], MobileNetV2 [16], GhostNet [17], etc. As a result, we extensively optimize the architecture of YOLOv8-n. Our goal is not only to improve the detection accuracy but also to reduce the computational overhead and improve the operating efficiency.

3. Materials and Methods

3.1. The Original YOLOv8 Algorithm

This paper builds upon YOLOv8-n version 8.1, introduced by the Ultralytics team in 2023. YOLOv8-n ensures smooth interoperability with prior YOLO iterations, enabling direct performance evaluations. The YOLOv8-n algorithm, incorporating fresh enhancements over YOLOv5, exhibits remarkable performance in object detection, achieving a previously unmatched equilibrium between speed and accuracy [18,19]. The algorithm's architecture is primarily segmented into input, backbone network, neck network, and output, as depicted in Figure 1.

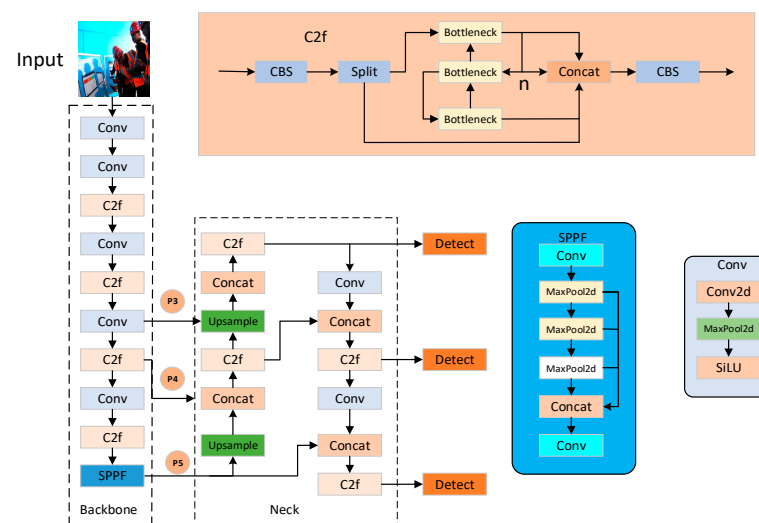


Figure 1. Network structure of YOLOv8.

The backbone network is primarily tasked with feature extraction. The YOLOv8-n algorithm replaces the earlier C3 structure [20] with the C2f structure, which enriches gradient flow and tailors the channel count to suit different algorithm scales. The neck network is comprised of two components: the Feature Pyramid Network (FPN) [21] and the Path Aggregation Network (PAN) [22], which together construct the feature pyramid. PANet merges bottom-up and top-down strategies to effectively capture multi-scale information, aiding in the fusion of information across various feature layers. This enhancement allows the detector to more adeptly accommodate objects of diverse sizes and shapes, thereby boosting detection accuracy. Furthermore, YOLOv8-n omits the 1×1 convolution prior to upsampling within the neck component, thereby diminishing computational load. In the head component, the conventional decoupled-head structure [23] is employed to segregate the classification and detection functionalities, aiming to mitigate the inherent tension between classification and localization objectives. YOLOv8-n applies the BCE loss for

classification tasks, while the regression loss integrates the focus loss, specifically the DFL loss, along with the CIOU loss.

3.2. LG-YOLOv8 Algorithm

Enhancing the efficiency of the YOLOv8-n algorithm for detecting safety helmets by optimizing its lightweight capabilities, this study introduces the LG-YOLOv8. The algorithm is characterized by optimizing the original C2f module by introducing the C2f-GhostDynamicConv module, an advanced dynamic convolution method. This module enhances feature representation by dynamically adjusting the convolution kernel, allowing the model to more accurately capture details and adapt to diverse detection contexts. Simultaneously, this innovation significantly reduces the model's parameter count and computational load, providing a robust foundation for the deployment of lightweight object detection. Subsequently, the unidirectional pyramid network in the neck section is upgraded to a bidirectional pyramid network (BiFPN), a configuration that enriches the representation of feature maps across different scales via a bidirectional fusion approach, thereby further elevating detection precision. Lastly, the detection head is innovatively replaced with LADH-Head, a lightweight asymmetric detection head, which ensures accurate detection of targets of varying sizes while preserving the model's operational efficiency through the asymmetric multi-stage compression technique and dual-head architecture. By integrating these modules, the YOLOv8-n model not only retains its computational efficiency advantage but also achieves a significant leap in target detection accuracy and adaptability, particularly in handling complex visual environments, rendering it more suitable for deployment on devices with constrained computational resources.

3.2.1. GhostDynamicConv

To facilitate the model's operation on edge devices for accurate helmet detection within smart construction site settings that have constrained computational resources, we have integrated the C2f-GhostDynamicconv module to refine the detection model. This architecture is designed to augment the C2f component of the backbone network by harnessing the combined advantages of the Ghost module and dynamic convolution, utilizing a novel feature extraction method. It selectively merges various convolution kernels in response to the incoming data and adapts the weights, thus promoting the identification and portrayal of features pertinent to helmets. This strategy adeptly tackles the difficulties associated with identifying small objects, dense crowds, and intricate backgrounds within helmet detection contexts. Moreover, this module optimizes the usage of computational resources and enhances the efficacy of operations performed with reduced precision floating-point arithmetic.

- GhostModule

The traditional convolution method has a large amount of calculation, which cannot eliminate redundant features well for safety helmet detection, and the smaller convolution kernel cannot reduce memory and FLOP to a certain extent. Han et al. [17] introduced the Ghost module, with the core idea of GhostNet being to produce additional feature maps through inexpensive operations and fewer parameters. The output of regular convolution may have redundancy, and this kind of redundant feature map can be obtained by simple linear operation based on a certain base feature. The Ghost module, as the base module of GhostNet, extracts rich feature information by regular convolution and generates the redundant feature information by adopting inexpensive linear transformation, which can significantly reduce the computational resources needed by the model. Figure 2 shows the schematic diagram of conventional convolution and Ghost convolution, both of which can be specifically expressed by the formula, where x denotes the cheap linear operation, and both of them will output the same number of feature maps.

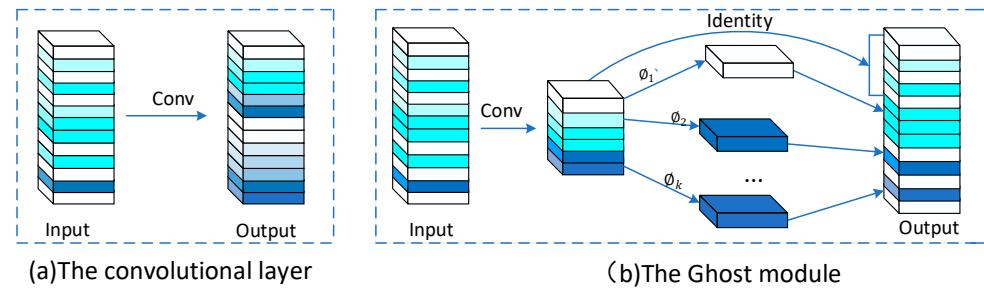


Figure 2. Conventional convolution and Ghost module. (a) The convolutional layer; (b) The Ghost module.

Standard convolution:

$$Y = \text{Conv}(X, W) \text{ where } Y \in \mathbb{R}^{H' \times W' \times C'} \quad (1)$$

Cheap transformations:

$$Y^{\text{ghost}} = \{\phi_1(Y), \phi_2(Y), \dots, \phi_k(Y)\} \text{ where } k = \frac{s \times C'}{C} = s \quad (2)$$

In this scenario, if the input feature map is of size $H \times W \times C$, the standard convolution produces a basic feature map of size $H' \times W' \times C'$, where C' is typically less than C . Subsequently, the “artifacts” are generated through cost-effective transformation s times. This results in s -fold “artifacts” produced by the inexpensive transformation, leading to a total of $s \times C'$ feature maps. In the design principle of GhostModule, depthwise separable convolutions are employed to carry out efficient linear transformations, thereby generating the so-called “Ghost feature maps”. These feature maps are produced by depthwise separable convolution, where s represents the standard convolution operation and ϕ indicates the linear transformation implemented through depthwise separable convolution. With this design, GhostModule can considerably reduce the number of parameters and computational cost while preserving the accuracy of the network.

Assuming that there are s redundant features for each base feature, then only n/s base features need to be output, followed by an inexpensive linear transformation of each base feature to generate the corresponding redundant features. This reduces the computation of the model, and the GFLOPS is $1/s$ of the standard convolution. Let the average size of the convolution kernel used in the linear operation part be $d \times d$, the theoretical speedup ratio r , as shown in Equation (3). In the context of the output feature layer, h denotes the length and w represents the width. The variable k signifies the dimension of the standard convolution kernel, while n indicates the total number of convolutions performed. The parameter c corresponds to the channel count of the input. If d squared is equivalent to k squared and s is no greater than c , the theoretical increase in speed is s times that of the initial configuration:

$$r = \frac{n \cdot h \cdot w \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h \cdot w \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h \cdot w \cdot d \cdot d} = \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \quad (3)$$

- C2f-GhostDynamicConv

Large-scale visual training tasks such as helmet detection often require a low FLOP model that can still provide good detection advantages; therefore, DynamicConv was merged with GhostBottleneck and incorporated into the initial C2f module, yielding the C2f-GhostDynamicConv module. Dynamic convolution, as discussed in the literature [24,25], improves feature representation over traditional static convolution. The main core idea is to dynamically generate convolution kernels based on input features or to fine-tune the convolution kernels. Dynamic convolution can adaptively adjust the weights of the convolution kernel according to the characteristics of the input image, which can better capture the detailed features of the helmet, in addition to dealing with environmental

changes, angular changes, and occlusion problems, and improve the robustness of the detection rate in a variety of site environments. DynamicConv is based on a dynamic perception whose basic principle is illustrated in the following equations:

$$y = g\left(\tilde{W}^T(x)x + \tilde{b}(x)\right) \quad (4)$$

$$\tilde{W}(x) = \sum_{k=1}^k \pi_k \tilde{W}_k, \tilde{b}(x) = \sum_{k=1}^K \pi_k(x) \tilde{b}_k \quad (5)$$

$$\text{s.t. } 0 \leq \pi_k(x) \leq 1, \sum_{k=1}^k \pi_k(x) = 1 \quad (6)$$

Within the given formula, x signifies the input data, while y represents the resulting output. It is evident that x is subjected to a pair of separate processes: initially, it serves as a parameter for the attention mechanism that formulates the convolution kernel, and subsequently, it is applied in the convolution process itself. The variables W , b , and g correspond to the weight matrix, bias vector, and activation function, respectively. The term $\pi_k(x)$ indicates the attention coefficient for the k -th linear transformation, highlighting that this coefficient is contingent upon the input x . The kernel is composed of a convolutional filter intertwined with the k -th linear transformation. The dynamic convolution is fundamentally made up of k convolutional kernels and adheres to the conventional architecture of convolutional neural networks (CNNs). It incorporates batch normalization (BatchNorm) and the Rectified Linear Unit (ReLU) as its activation function. Figure 3 provides a detailed depiction of the architecture of DynamicConv.

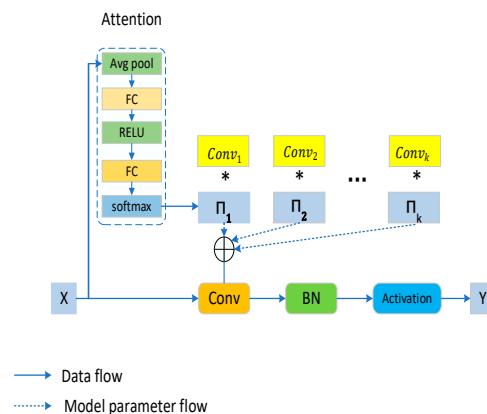


Figure 3. The structure of DynamicConv.

In contrast to DynamicConv, which is predominantly concerned with channel-wise attention, SENet, or Squeeze-and-Excitation Networks [26], treats the entire convolutional kernel as a focus of attention. By extending the concepts of DynamicConv and adopting the architectural innovations of the GhostBottleneck module, the initial C2f module was upgraded to form the C2f-GhostDynamicConv module. This innovative design markedly enhances precision with a minimal increase in the computational complexity, measured in FLOPs. Figure 4 contrasts this innovative design with the C2f module in the baseline model YOLOv8-n.

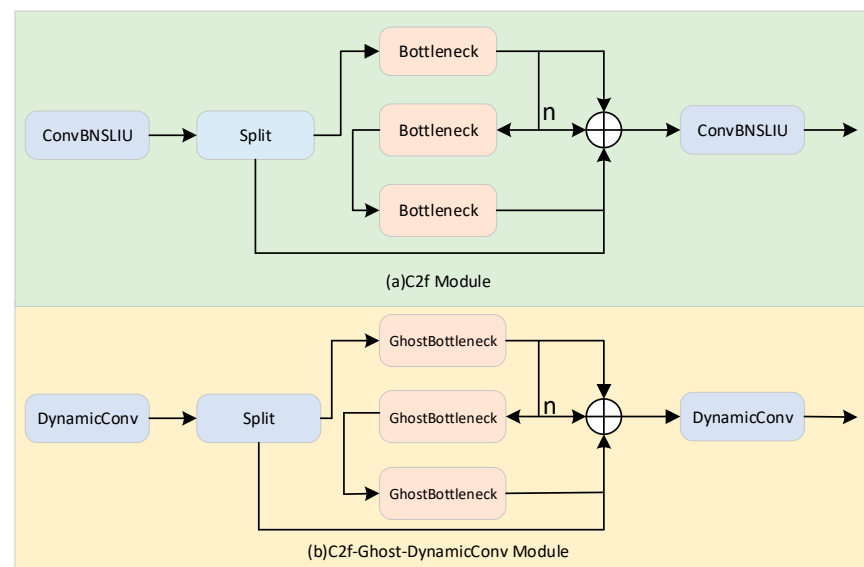


Figure 4. Comparison of C2f and C2f-GhostDynamicConv modules: (a) C2f module; (b) C2f-GhostDynamicConv module.

3.2.2. BiFPN

The neck design of YOLOv8 draws inspiration from PANet [27], enhancing feature fusion by combining the Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) architectures, as illustrated in Figure 5a,b. Unlike YOLOv5 [28] and YOLOv7, YOLOv8's FPN architecture employs top-down fusion, where high-level features are up-sampled and integrated into low-level features, enriching them with semantic details. Conversely, PAN adopts a bottom-up strategy, downsampling low-level features and incorporating them into high-level features, thereby augmenting spatial precision. The integration of these techniques forms the PANet structure, facilitating adaptive fusion of shallow and deep features, which substantially enhances the model's capability to discern fine details. Nonetheless, when merging features of disparate resolutions, the features are typically aggregated through straightforward summation, resulting in unequal contributions from various layers. As the network progresses deeper, the impact of high-level features intensifies, while low-level features diminish. To address this, TAN et al. [29] proposed BiFPN to produce feature maps with more enriched semantic content.

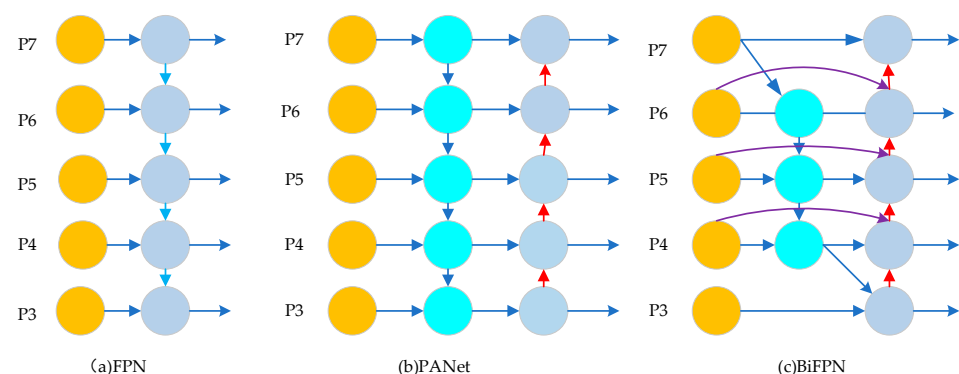


Figure 5. (a) FPN introduces a top-down path to fuse multiscale features from the third to the seventh level (P3–P7); (b) PANet enhances the FPN (Feature Pyramid Network) by incorporating an additional bottom-up pathway; and (c) BiFPN offers a superior balance between accuracy and efficiency.

BiFPN introduces bi-directional connectivity between neighboring levels of the feature pyramid, which means that information can flow from higher level features to lower level features (top-down path) or from lower level features to higher level features (bottom-up

path), which greatly enhances semantic representation and differentiation. BiFPN adopts separated convolution and lightweight design, which can improve efficiency and also reduce the number of parameters and computation. For the helmet detection task, the bidirectional feature pyramid can better solve the situation that small targets are not well recognized, which in turn leads to omission and misdetection [30].

As illustrated in Figure 5c, BiFPN is constructed upon bidirectional cross-scale connections and weighted multi-scale feature fusion. Distinct from PANet, BiFPN incorporates learnable weights to assess the significance of various input features during both top-down and bottom-up processes. The BiFPN network excludes nodes with a single input edge; such nodes are deemed insufficient for effective feature fusion due to their limited information content and reduced contribution to the feature network. Incorporating additional edges between input and output nodes within the same layer and obtaining higher-level fused features through iterative stacking enhances the feature fusion efficiency and accuracy of the BiFPN network, thereby boosting the model's capability to extract features pertinent to a specific scene. By configuring the learning parameters to ensure that the input layers are of uniform size and augmenting each feature layer with additional weights, the BiFPN network is capable of effectively merging input features of varying resolutions and evaluating their contribution to the output. The fast normalized fusion method is employed to weight the feature layer, specifically through the BiFPN weighted feature fusion technique:

$$O = \sum_i \frac{w_i \cdot I_i}{\sum_j w_j + \epsilon} \quad (7)$$

O represents the output from the BiFPN network; I_i denotes the feature from the i th layer; w_i signifies the weight associated with the i th layer feature; and w_j refers to the j th weight within the i th layer; $\epsilon = 0.001$ is used to avoid numerical instabilities, as indicated by the notation.

$$P_6^{td} = \text{Conv} \left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{\epsilon + w_1 + w_2} \right) \quad (8)$$

$$P_6^{out} = \text{Conv} \left(\frac{w'_1 \cdot P_6^{td} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right) \quad (9)$$

P_6^{td} denotes the feature transmitted to the sixth layer in a top-down manner; P_6^{out} represents the resultant feature of the sixth layer; Conv refers to the convolution process. P_6^{in} indicates the initial feature of the sixth layer; P_7^{in} signifies the entry feature for the seventh layer; P_5^{out} is the feature escalated within the fifth layer; Resize encompasses both upsampling and downsampling operations. The symbol ϵ denotes an infinitesimally small positive value; w_1 is the weighting factor for the transition from P_6^{in} to P_6^{td} ; w_2 is the weighting applied to P_6^{td} following the downsampling of P_7^{in} ; w'_1 is the weighting for the progression from P_6^{td} to P_6^{out} ; w'_2 is the weighting assigned to P_6^{out} ; and w'_3 is the weighting for P_6^{out} subsequent to the upsampling of P_5^{out} .

3.2.3. LAHD-Head

In YOLOv8-n, a symmetric multistage compression framework is implemented, with uniform compression ratios across all detection heads. The detection head of YOLOv8 features a decoupled architecture, consisting of dual parallel streams tasked with extracting features for object classification and localization independently. Each stream is refined by a 3×3 convolutional module, adeptly capturing spatial characteristics within the image and supplying comprehensive data for subsequent categorization and positioning tasks. Following this, the output from each convolutional module is subjected to further refinement by a Conv2d module, which sharpens the feature representation and bolsters the model's object identification prowess. Ultimately, the results from these parallel tracks are utilized to calculate the classification loss (Cls loss) and the bounding box regression loss (Box loss), respectively. This decoupled approach not only optimizes the model's

detection accuracy in intricate environments but also streamlines the training process by enabling targeted optimization for distinct objectives. However, despite the decoupled design's substantial improvement in detection capabilities, it does not optimally harness the sophisticated features derived from the backbone network. The schematic of the YOLOv8-n detection head is depicted in the accompanying Figure 6.

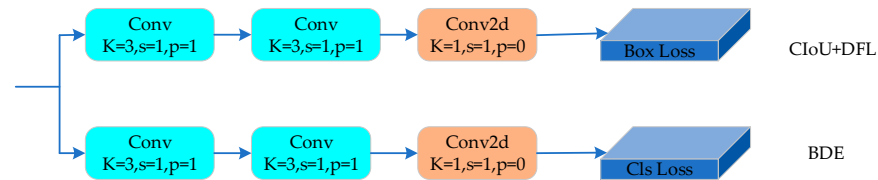


Figure 6. YOLOv8-n Detection Head.

To address the issue that conventional detection heads do not efficiently leverage the sophisticated features derived from the backbone network, this study presents an innovative, lightweight, asymmetric decoupled detection head (LADH-Head) [31]. Figure 7 illustrates the architecture of the detection head introduced in this study. A notable benefit of the LADH-Head is its implementation of 3×3 depthwise separable convolution (DWConv) in place of the standard 3×3 convolution, which significantly cuts down the parameter count and also manages to broaden the receptive field, enhance the task-specific parameters of the IOU branch, and mitigate both misdetections and false positives. The DWConv achieves a reduction in computational complexity by breaking down the standard convolution operation into depthwise convolution and pointwise convolution. The adoption of this asymmetric, lightweight detection head can substantially decrease the parameter count and enhance computational efficiency, making it ideal for environments with limited resources. Furthermore, the pointwise convolution integrates data across different channels, maintaining an adequate level of capability for feature extraction. By upgrading the detection head component, the model is now capable of higher detection accuracy while also reducing the number of model parameters and the computational resources required. This reduction in complexity and resource demands enhances computational efficiency.

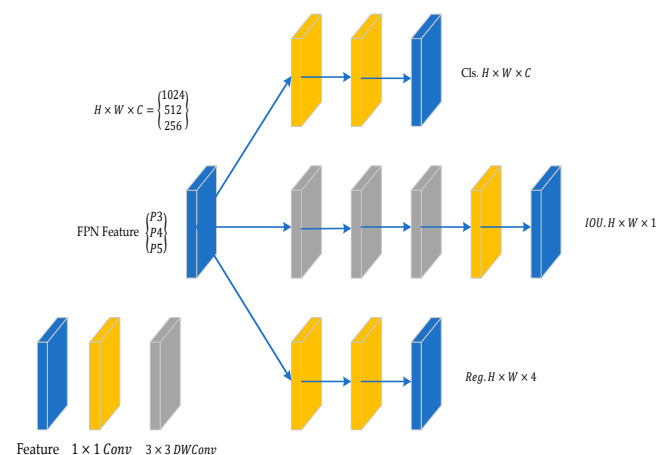


Figure 7. Network structure of the lightweight asymmetric detection head (LADH-Head).

4. Results

4.1. Experimental Environment and Dataset

The environment configuration for this experiment is shown in Table 1 below.

Table 1. Experimental environment configuration.

Name	Environmental Parameters
Operating system	Windows 11 Chinese version 64 bit
Graphics processing unit	NVIDIA GeForce RTX4050
Video memory	16
Python	3.8
Framework	PyTorch1.11

The key parameters for this network training are shown in Table 2.

Table 2. Configuration of training parameters.

Parameter	Value
Mosaic	1.0
Weight decay	0.0005
Batch size	8
Epochs	100
Momentum	0.937
Learning rate	0.01

The publicly available dataset SHWD (Safety-Helmet-Wearing Dataset) [32–34] is used for this experiment. The SHWD dataset is a high-quality dataset focusing on helmet-wearing detection, which consists of a series of well-labeled images and is designed to train and evaluate a computer vision model on the task of helmet-detecting performance. The SHWD dataset contains 7581 images, including head objects wearing helmets (positive sample) and head objects not wearing helmets (negative sample). In this paper, the SHWD dataset is divided into a training set, a test set, and a validation set according to the ratio of 7:1:2 by random partition. After partitioning, the training set contains 5457 images, the test set contains 607 images, and the validation set contains 1517 images. The SHWD dataset contains two categories, “hat” and “person”, respectively. Below is a selection of images from the SHWD dataset, as depicted in Figure 8.

**Figure 8.** Example of experimental dataset (Reprinted from [35]).

4.2. Evaluation of Indicators

To verify the model’s performance, the main evaluation metrics include precision, mean average precision, number of parameters, FLOPs, model size, and frames per second (FPS).

$$P = \frac{TP}{TP+FP} \quad (10)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

$$FPS = \frac{1}{Latency} \quad (12)$$

$$Parameters = (I + 1) \times O = I \times O + O \quad (13)$$

where TP (true positive) denotes a positive sample correctly categorized as positive, FP (false positive) denotes a negative sample incorrectly categorized as positive, and FN

(false negative) denotes a positive sample incorrectly categorized as negative. N denotes the sample category, P (precision) denotes the precision rate, and AP (average precision) denotes the average precision of a given analogy. I denotes the number of channels in the input feature map, O denotes the number of channels in the output feature map, and the +1 in the formula comes from the bias term, which has one for each output channel. The number of parameters and the amount of computation FLOPs measure the complexity of the model, while FPS measures the model's speed.

4.3. Ablation Experiments

To determine the impact of the suggested enhancements, a series of six experiments were crafted with the YOLOv8-n model serving as the foundational point for ablation analysis. The objective of these experiments was to measure the practicality and efficacy of each augmentation component, with the outcomes detailed in Table 3.

Table 3. Results of ablation experiments.

Model	G-DConv	BiFPN	LADH	Para/10 ⁶	Model Size/M	FLOPS/G	P/%	mAP _{0.5} /%	FPS
Model-1	×	×	×	3.0	6.0	8.1	92.41	92.2	106.3
Model-2	✓	×	×	2.2	4.6	5.7	93.4	93.7	158.6
Model-3	×	✓	×	2.0	4.2	7.1	93.5	94.5	204.4
Model-4	×	×	✓	2.4	4.8	5.7	91.9	94	247.1
Model-5	✓	✓	×	1.7	3.4	5.7	93.5	94	133.8
Model-6	✓	✓	✓	1.2	2.7	3.7	92.2	94.1	153.3

Model-1 refers to the original YOLOv8-n, while Model-2 is the version where the C2f module is replaced with C2f-GhostDynamicConv in the YOLOv8n backbone. Compared to the baseline, Model-2 improves accuracy by 1.5%, with a 26.7% reduction in parameters and a 29.2% drop in computation. This shows that the replacement model has a more significant impact on parameter reduction. Model-3 introduces a bidirectional feature pyramid in the feature fusion network, slightly increasing computation but boosting accuracy by 2.3%, reducing parameters by 33.3%, and decreasing model size by 30% compared to Model-2. This demonstrates that the bidirectional feature pyramid excels at feature extraction and model optimization through separable convolution and a lightweight design. Model-4 replaces the baseline detection head with the lighter LADH-Head, which does not improve accuracy or model size over Model-3 but enhances frames per second by 20.9%. Model-5 adds a bidirectional feature pyramid to Model-2, resulting in a slight accuracy increase, a 22.7% reduction in parameters, and a 26% decrease in model size. Model-6 builds on Model-5 by adding LADH-Head, showing a 1.9% accuracy improvement compared to Model-1, with the model size reduced to 55% of the original and computation cut by 54.3%. These experiments show that the optimized YOLOv8n algorithm not only maintains high accuracy in helmet detection but also significantly reduces parameters, computation, and model size.

4.4. Model Training Analysis Before and After Improvement

Figure 9 shows the changes in key metrics of the YOLOv8-n and LG-YOLOv8 models throughout the training process, further demonstrating the superiority of the improved models. Figure 9a–d depicts the fluctuations of precision, recall, mAP_{0.5}, and mAP_{0.5:0.95}, respectively, for the training cycle (epoch = 100). The comparison of the accuracies clearly shows that the accuracy of LG-YOLOv8 is always higher than that of YOLOv8-n throughout the training process, which indicates that the model has been improved. By observing the recall rate, the recall rate increases rapidly in the first 40 epochs. And in the whole training process, the recall rate of LG-YOLOv8 is always higher than that of YOLOv8-n, which indicates that the improved model has a lower miss rate and can better detect all positive samples. The mAP_{0.5} curve shows that LG-YOLOv8 is

higher than YOLOv8-n in the whole training process, which indicates that the improved model has stronger robustness for safety helmet detection and can detect and identify targets more effectively. It can be observed from mAP_0.5:0.95 that the enhanced model demonstrates more superior overall detection performance under diverse matching circumstances throughout the entire 100 rounds of training. Figure 10 presents the variations of different types of losses of the YOLOv8-n and LG-YOLOv8 models during training and validation. The loss functions in Figure 10a,b indicate that the enhanced model converges more rapidly and has better performance in inference. In summary, the superiority of LG-YOLOv8 in all indicators indicates that optimized modules can effectively improve the overall performance of the model, further highlighting the higher accuracy and reliability of LG-YOLOv8 in safety helmet detection.

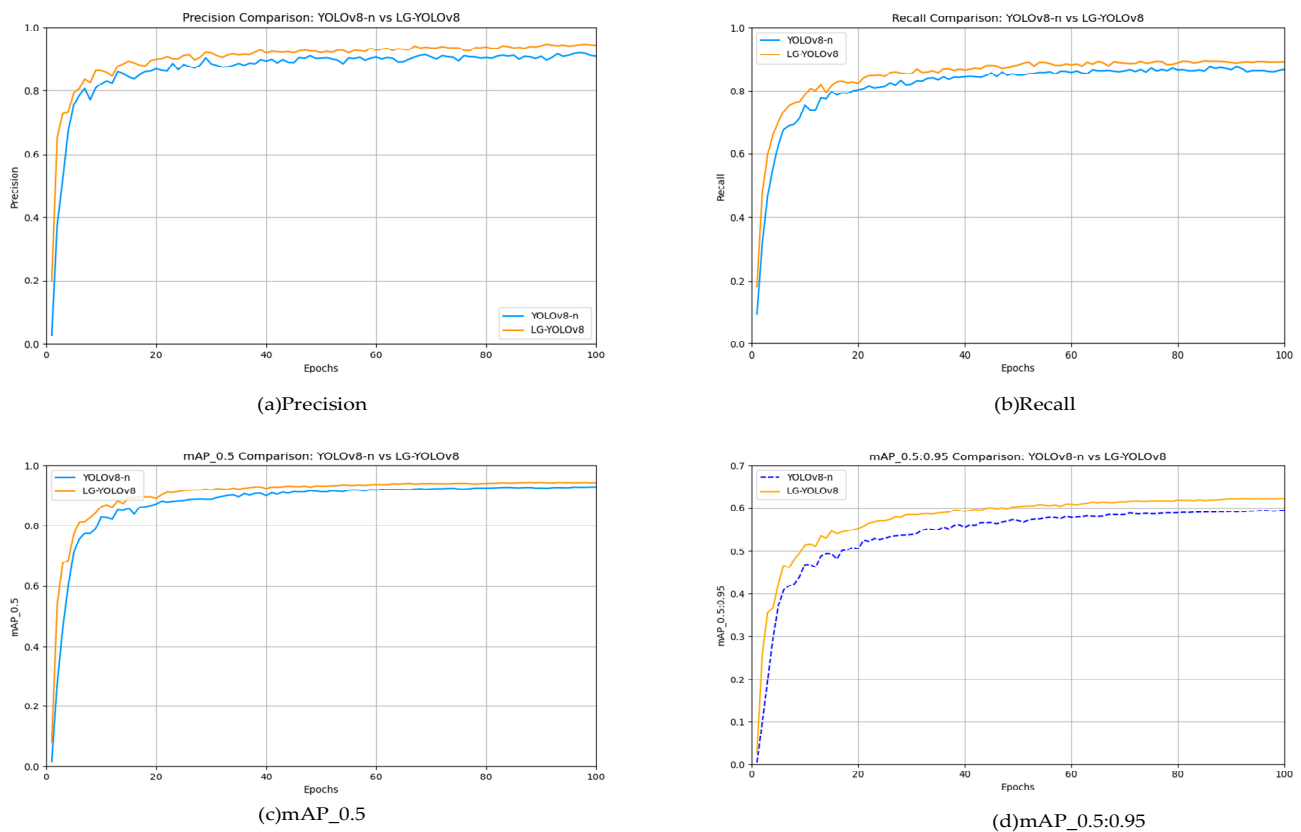


Figure 9. Changes in key metrics during YOLOv8-n and LG-YOLOv8 trainings.

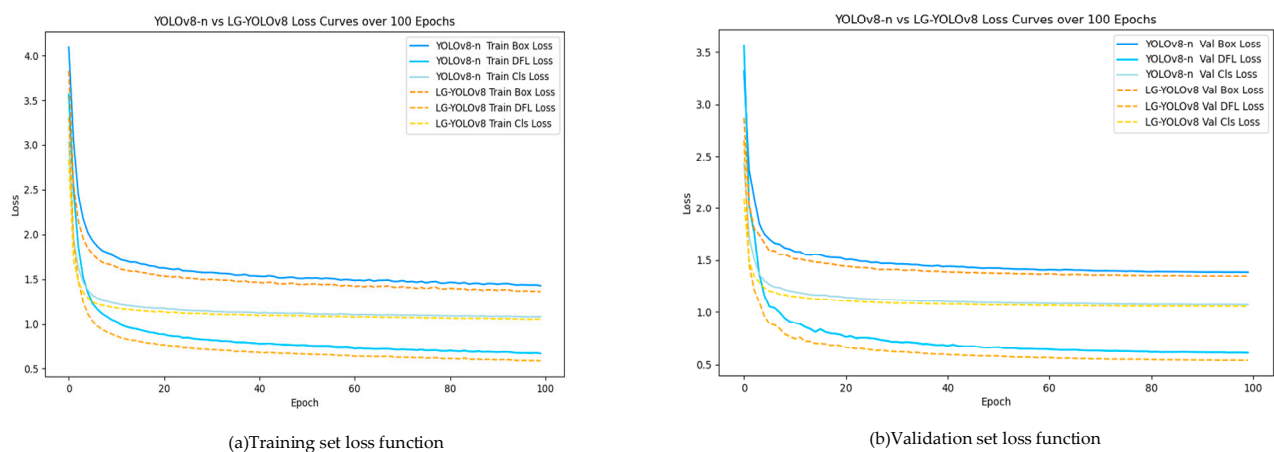


Figure 10. Changes in loss during YOLOv8-n and LG-YOLOv8 training.

4.5. Comparative Experiments

To underscore the effectiveness and practicality of the LG-YOLOv8 algorithm for compact studies, we compared its performance to leading algorithms in helmet detection. The comparative analysis and respective data are elaborated in Table 4.

Table 4. Comparison results of different models.

Model	Parameters/10 ⁶	FLOPS/G	P/%	mAP _{0.5} /%
SSD [3]	12.3	28.7	-	90.2
YOLOv3-Tiny	8.7	12.9	-	61.2
YOLOv3	61.5	154.9	-	89.1
YOLOv5s	7.1	15.8	93.6	90.9
YOLOv7	37.2	105.1	93.3	93.4
ML-YOLOv3 [36]	18.1	46.0	-	86.5
MG-YOLOv5s [37]	4.06	67	-	95.4
YOLOv8-SLIM-CA [38]	2.74	11.3	93.8	94
YOLOv8n-ASF-DH [39]	5.01	11.8	94	95.6
LG-YOLOv8	1.2	3.7	92.2	93.4

Table 4 reveals that LG-YOLOv8 outperforms SSD in average accuracy by a margin of 3% while substantially reducing parameter count and computational load. When juxtaposed with YOLOv5 and YOLOv7, our model demonstrates a commendable lightweight effect. In comparison to enhanced variations of YOLOv3 and YOLOv3-Tiny, the refined algorithm of this study shows marked enhancements in average accuracy, with a pronounced decrease in computational demand and parameter requirements. Further, against YOLOv5-MN, LG-YOLOv8 exhibits a marginal uptick in average accuracy with a notable 59.3% reduction in computation. Against three advanced YOLOv8 iterations—YOLOv8-Slim-CA, FEV-YOLOv8n, and YOLOv8-S-LE—the algorithm developed herein stands out, especially in parameter number and computation efficiency. The current model's strength is not just in maintaining high precision but also in optimizing resource usage, making it highly suitable for edge device deployment. The effectiveness of LG-YOLOv8 is visually corroborated by Figure 11, highlighting the algorithm's advantage with greater clarity.

Histogram comparison of the results of different algorithms

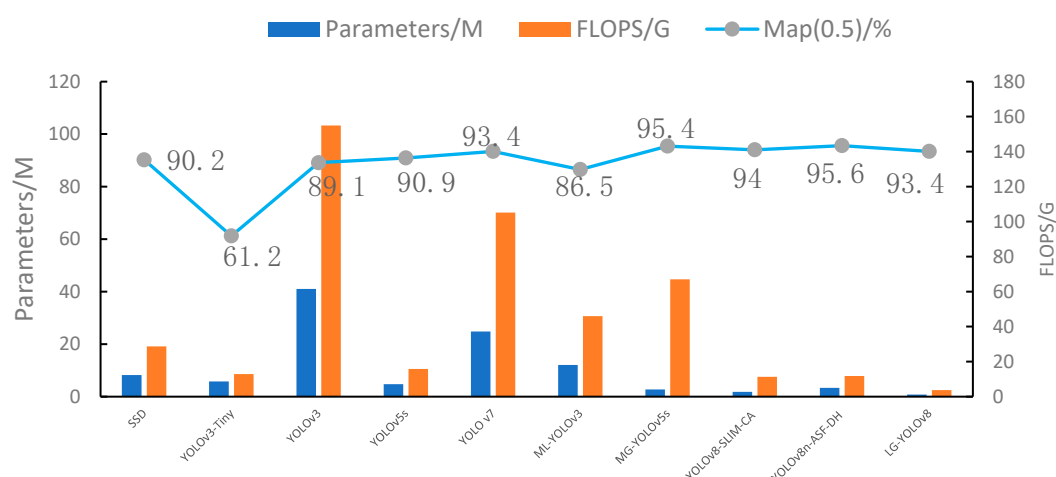


Figure 11. Histogram comparison of results of different algorithms.

4.6. Visualization of the Results

To offer a clearer evaluation of the enhanced YOLOv8-n model's detection capabilities, we chose safety helmet detection in diverse challenging environments from the SWHD dataset. The selected original images pose challenges in identifying small objects, handling

crowded scenes, and managing complex scenarios. As shown in Figure 12, our proposed method not only enhances the feature extraction ability but also effectively addresses missing cases in some complex scenes and small target cases.



Figure 12. Visualization results for different scenarios (adapted from ref. [35]). (a) The pictures in the original dataset and the helmet detection pictures in different scenarios; (b) Base model YOLOv8-n; (c) Improved model LG-YOLOv8.

5. Conclusions

To counteract the deficiencies of the standard YOLOv8-n algorithm and address external factors that can affect helmet detection performance, this manuscript presents an advanced model known as LG-YOLOv8. Constructed upon the YOLOv8-n foundation, this model integrates a triad of architectural enhancements. These improvements notably decrease the computational load without sacrificing accuracy, aligning the model more closely with real-world usage scenarios. The introduction of the C2f-GhostDynamicConv module serves to refine the original C2f module, leading to more efficient use of computational resources and better performance in low-precision floating-point operations. Moreover, the paper presents a revamped feature extraction segment and detection head, complete with a bidirectional feature pyramid that amplifies the semantic data extracted from helmet images, consequently bolstering detection capabilities. The novel lightweight asymmetric detection head, termed LADH-Head, minimizes model parameters and accelerates detection without compromising the integrity of the classification and regression processes initially challenged by the traditional coupled head design. Empirical evidence supports that LG-YOLOv8 is adept at fulfilling helmet detection requirements with robust practicality, achieving a commendable equilibrium between detection proficiency and computational expenditure. It outperforms its contemporaries in specific performance metrics and is better suited for deployment on resource-constrained edge devices. While the algorithm has demonstrated impressive outcomes, there remains ample opportunity for refinement. In subsequent research, we aim to further unearth the untapped potential of the YOLO model for helmet detection. Additionally, we plan to delve into more potent

algorithms for the detection and tracking of smaller targets to augment their recognition precision. We intend to investigate the adaptability of our method to other object detection algorithms to harvest superior performance. Moreover, we will seek to amass a more extensive and diverse dataset to fortify the models' generalization capabilities, rendering them more potent and adaptable in intricate industrial settings.

Author Contributions: Conceptualization, Y.W. and Z.F.; methodology, Z.F.; formal analysis, Y.W. and Z.F.; investigation, Y.W., M.C. and W.L.; resources, Y.W. and Z.F.; data curation, Y.W.; writing—original draft preparation, Z.F.; writing—review and editing, Y.W. and Z.F.; visualization, Z.F.; supervision, Y.W. and Z.F.; project administration, Z.F. and Z.Q.; funding acquisition, Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Heilongjiang Postdoctoral Fund to pursue scientific research (grant number LBH-Z23025), Heilongjiang Province Colleges and Universities Basic Scientific Research Business Expenses Project (grant number 2023-KYYWF-1052), Harbin University of Commerce Industrialization Project (grant number 22CZ04), Collaborative Innovation Achievement Program of Double First-class Disciplines in Heilongjiang Province (grant number LJGXCG2022-085).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article. The SHWD dataset can be download in <https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset> (accessed on 22 July 2024). Additional original contributions presented in the study are included in the article, and further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
4. Xu, X.; Jiang, Y.; Chen, W.; Huang, Y.; Zhang, Y.; Sun, X. DAMO-YOLO: A Report on Real-Time Object Detection Design. *arXiv* **2022**, arXiv:2211.15444.
5. Adarsh, P.; Rath, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694.
6. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
7. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
8. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
9. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You Only Look One-level Feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; p. 13034.
10. Glenn, J.; Ayush, C.; Jing, Q. Ultralytics YOLOv8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 2 July 2024).
11. Lian, Y.; Li, J.; Dong, S.; Li, X. HR-YOLO: A Multi-Branch Network Model for Helmet Detection Combined with High-Resolution Network and YOLOv5. *Electronics* **2024**, *13*, 2271. [\[CrossRef\]](#)
12. Liang, H.; Seo, S. Automatic detection of construction workers' helmet wear based on lightweight deep learning. *Appl. Sci.* **2022**, *12*, 10369. [\[CrossRef\]](#)
13. Chen, S.; Tang, W.; Ji, T.; Zhu, H.; Ouyang, Y.; Wang, W. Detection of safety helmet wearing based on improved faster R-CNN. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–7.
14. Chen, J.; Deng, S.; Wang, P.; Huang, X.; Liu, Y. Lightweight helmet detection algorithm using an improved YOLOv4. *Sensors* **2023**, *23*, 1256. [\[CrossRef\]](#) [\[PubMed\]](#)

15. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 116–131.
16. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
17. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
18. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [\[CrossRef\]](#)
19. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [\[CrossRef\]](#)
20. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H. Cspnet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE 2020/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
21. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
22. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
23. Ma, N.; Su, Y.; Yang, L.; Li, Z.; Yan, H. Wheat Seed Detection and Counting Method Based on Improved YOLOv8 Model. *Sensors* **2024**, *24*, 1654. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Yang, B.; Bender, G.; Le, Q.V.; Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. *Adv. Neural Inf. Process. Syst.* **2019**, *32*. Available online: <https://neurips.cc/virtual/2019/poster/13351> (accessed on 26 October 2024).
25. Han, K.; Wang, Y.; Guo, J.; Wu, E. ParameterNet: Parameters Are All You Need for Large-scale Visual Pretraining of Mobile Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 15751–15761.
26. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
27. Kim, J.H.; Kim, N.; Park, Y.W.; Won, C.S. Object detection and classification based on YOLO-V5 with improved maritime dataset. *J. Mar. Sci. Eng.* **2022**, *10*, 377. [\[CrossRef\]](#)
28. Qu, Z.; Gao, L.Y.; Wang, S.Y.; Yin, H.N.; Yi, T.M. An improved YOLOv5 method for large objects detection with multi-scale feature cross-layer fusion network. *Image Vis. Comput.* **2022**, *125*, 104518. [\[CrossRef\]](#)
29. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
30. Guo, C.; Fan, B.; Zhang, Q.; Xiang, S.; Pan, C. Augfpn: Improving multi-scale feature learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 12595–12604.
31. Zhang, J.; Chen, Z.; Yan, G.; Wang, Y.; Hu, B. Faster and Lightweight: An Improved YOLOv5 Object Detector for Remote Sensing Images. *Remote Sens.* **2023**, *15*, 4974. [\[CrossRef\]](#)
32. Wang, J.; Zhu, G.; Wu, S.; Luo, C. Worker’s helmet recognition and identity recognition based on deep learning. *Open J. Model. Simul.* **2021**, *9*, 135–145. [\[CrossRef\]](#)
33. Yue, S.; Zhang, Q.; Shao, D.; Fan, Y.; Bai, J. Safety helmet wearing status detection based on improved boosted random ferns. *Multimed. Tools Appl.* **2022**, *81*, 16783–16796. [\[CrossRef\]](#)
34. Sun, H.; Gong, P. November. A Safety-Helmet Detection Algorithm Based on Attention Mechanism. In Proceedings of the 2021 7th IEEE International Conference on Network Intelligence and Digital Content, Beijing, China, 17–19 November 2021; pp. 11–15.
35. njvisionpower. Safety-Helmet-Wearing-Dataset. 2019. Available online: <https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset> (accessed on 22 July 2024).
36. Deng, L.; Li, H.; Liu, H.; Gu, J. A lightweight YOLOv3 algorithm used for safety helmet detection. *Sci. Rep.* **2022**, *12*, 10981. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Xiao, Z.; Liu, W.; Ye, Z.; Yuan, J.; Liu, S. MG-YOLOv5s: A Faster and Stronger Helmet Detection Algorithm. *Comput. Syst. Sci. Eng.* **2024**, *48*, 1009–1029. [\[CrossRef\]](#)
38. Lin, B. Safety helmet detection based on improved YOLOv8. *IEEE Access* **2024**, *12*, 28260–28272. [\[CrossRef\]](#)
39. Lin, B. YOLOv8n-ASF-DH: An Enhanced Safety Helmet Detection Method. *IEEE Access* **2024**, *12*, 126313–126328. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.